International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Child Violence Detection in Surveillance Video Using Deep Transfer Learning and Ensemble Decision Fusion Learning

Elly Matul Imah[1]* Karisma[1]

[1]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Negeri Surabaya*
* Corresponding author's Email: ellymatul@unesa.ac.id

Abstract: Violence against children is a severe problem. Violence causes physical and mental trauma and can even threaten the lives of victims, especially children. Therefore, violent cases need special attention and require detection in their handling. Violence detection research is still a challenge for researchers and a considerable effort. A training process on video datasets is extensive empirical studies. Finding the optimal feature set and classifier is needed to achieve good recognition results. This paper presents violence detection using the visual geometry group network-16 (VGGNet-16)-based deep transfer learning feature extraction, combined with ensemble decision fusion learning. Ensemble decision fusion learning is a kind of ensemble learning method. It combines classifiers from multiple models and datasets. The majority of voting connects the classifier's output is used to decision fusion in this study. The majority voting counts the votes of the base learners and predicts the final class as an output; it is less biased toward the outcome. The deep learning classification methods used as an ensemble are LSTM, BiLSTM, GRU, and SVM. The experimental results show that a combination of VGGNet-16 and ensemble decision fusion learning can increase children's violence detection accuracy on surveillance videos. The obtained accuracy is 92.4% better 1.5% to 22.7% among other methods.

Keywords: Deep transfer learning, Child violence detection, Ensemble learning, Decision fusion learning, Deep learning.

1. Introduction

Violence against children is a global problem in societies with various backgrounds. Globally, every year it is estimated that one in two children aged 2-17 years experiences multiple acts of violence [1]. Approximately three in four children between the ages of two and four are subject to violent punishment by their caregivers [2, 3]. UNICEF reports that 1.8 billion children live in 104 countries where violence prevention and response services have been disrupted due to COVID-19 [4]. In some countries, the helplines call for domestic violence has increased 10-50% [5]. Developing an early detection system for violence is necessary with the high number of cases and fatalities impacted by violence against children.

Several studies on early violence detection have provided publicly accessible annotated video data. The authors applied the Histogram of Oriented Gradient (HOG) as feature extraction in [6]. Another study uses the Spatio Temporal Auto-Correlation Gradient (STACOG) feature to represent violent or non-violent activities that occur in surveillance videos [7]. Some utilize human skeleton parts (human skeleton) to detect violent mass activities [8]. Bruno Peixoto and his team used visual and audio features on violence detection [9].

Seymanur Akti and his team researched fight detection through surveillance cameras. They use the VGG16 and Xception network combined with Long Short Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) plus a self-attention layer as a classification method. Besides that, the authors also compiled a new dataset collected from surveillance cameras [10]. Aayush Jain and Dinesh Kumar Vishwakarma used motion features from dynamic images, using a deep neural network to study motion features to detect violence [11]. They classified several datasets, namely the Hockey Fight dataset and the Movies dataset. Mauricio Perez and his team did the same thing; they carried out video detection from surveillance cameras and collected a new dataset containing 1000 videos [12].

Research on the early detection of violence is still challenging. The development can be in system improvements in performance, accuracy, and detection time. Other studies apply it to some instances, such as violence against children. Violence against children is a severe problem. The consequences of this violence continue to be felt by children, both short and long-term. Moreover, children exposed to violence are more likely to become victims or perpetrators of violence in the future, which affects the new generation [13]. Lee Jia Thun and his team examine methods of detecting cyberbullying by collecting sample text/content testing comments from tweets. Tweets containing profane words are assumed to be more likely to become hate speech, leading to cyberbullying. These texts were selected and sent to parents via an app [14]. Apart from texts, recognizing emotions in voice signals can also detect child abuse [15]. Mahrukh Khan and his friends carried out the detection of violence in cartoons. With the selection of children’s spectacle, they hoped to prevent violent behavior both at home and at school [16].

Writers in [2] detect physical abuse in children using machine learning-based methods. They used skeletal data obtained by depth sensors. The problem is that not many schools or public places have these sensors. What is widely available in these places are surveillance cameras. In reality, the surveillance camera has not been used optimally to prevent an incident and is mainly used as evidence because actually, to carry out surveillance for 24 hours requires a lot of energy and materials. Therefore, we propose early detection of violence against children through video surveillance cameras. The use of deep learning for early violence detection has increased a lot to date and has shown good performance. Research conducted by Mostafa Mohamed Moaaz and Ensaf Hussein Mohamed managed to get an accuracy of up to 94.5% by applying deep learning to the Hockey dataset [17].

There are much research has been conducted about violence detection. However, there are still no specific ones for violence against children. This study focuses on violent cases against children using a deep learning approach. Detecting violent events against children is challenging because they are smaller than the perpetrators, and it is difficult to distinguish between their movements playing with abuse or tantrums. In addition to developing an early detection system, this study also compiled a dataset containing videos of violence against children because it is still difficult to find datasets devoted to violence against children. Before the detection process, feature extraction was performed using the Visual Geometry Group (VGG16). This study uses the wavelet method as the classical feature extraction method to compare with VGG-16 because wavelets show better performance than PCA[18]. The classification algorithm used is Long Short Term Memory (LSTM), Bidirectional LSTM, Gated Recurrent Unit (GRU), and Support Vector Machine (SVM). This study also used the classical machine learning algorithm, namely the Support Vector Machine (SVM), because SVM has a different approach and performs well when combined with deep transfer learning to detect violence in video data [19].

This study implements several combinations of the mentioned feature extraction method and classification algorithm on the Hockey, Movie, and Crowd dataset to perform better than the current study in terms of accuracy. The authors also collect a new dataset of violence against children. Furthermore, this study uses ensemble learning in the classification process, especially decision fusion, to detect violence against children based on video surveillance data. Decision fusion learning ensemble is an excellent method to improve classification and detection performance [20]. The ensemble learning method in the study by Saloni Kumari to detect diabetes mellitus can increase accuracy [21]. Deepak Gupta and Rinkle Rani’s research also shows that ensemble learning effectively improves model performance when applied to malware detection [22]. The ensemble learning in this study utilizes the best models of the combination algorithm in each dataset and improves the accuracy of child violence detection.

The arrangement of this paper includes the materials and methods used in this study described in Section 2. In Section 3, the results and explanations of the experiments that have been carried out will be explained. Then Section 4 discusses the conclusions.

2. Methods

In addition to detecting violence in children with deep transfer learning feature extraction, this study also analyzes the comparison of system performance using a deep learning approach with a conventional machine learning approach. The classical machine learning algorithm used as a comparison is the Support Vector Machine. The comparative feature extraction method used is wavelet. An explanation of

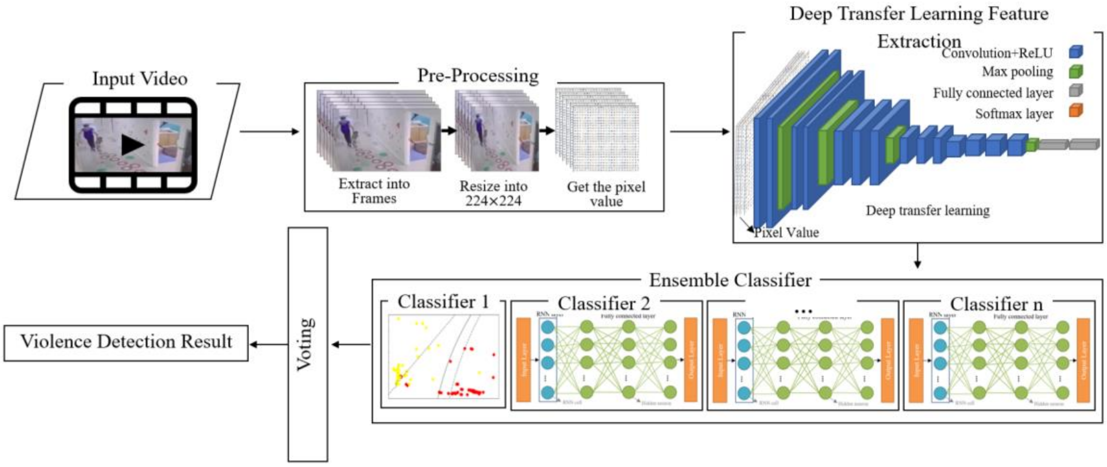


Figure. 1 Block diagram of child abuse classification

violence detection and the method used is described in this chapter in detail as follows.

2.1 Child violence detection

Many surveillance cameras are available in many places, especially in various public spaces such as schools, parks, supermarkets, and others. Surveillance cameras record almost all activities. This is a new problem because the records produced are not matched by adequate human resources to monitor. One solution that can use is to carry out monitoring automatically. This automatic supervision will undoubtedly help reduce the burden of monitoring because the monitoring process can be carried out non-stop, even with limited human resources.

Much automatic surveillance has been carried out, one of which is detecting acts of violence because it is not uncommon for acts of violence to be recorded by surveillance cameras. Detection of abnormalities in the video is a challenging task because the definition of anomalies can be ambiguous and vaguely defined. They vary widely based on the circumstances and situations in which they occur [23]. Various methods have carried out early detection of acts of violence. One widely used method and proven to produce good performance is the deep learning method. Several researchers have carried out early violence detection using deep learning methods such as [24, 25].

2.2 Data preprocessing

The process carried out in this study is depicted in Fig. 1. The first process carried out is preprocessing, then the data is partitioned into two parts, namely train data and data test. After that, each section is performed feature extraction and the results of the feature extraction will be classified. The next step is to evaluate the classification results. Evaluation is carried out by considering several parameters, namely accuracy, sensitivity, specificity, and Gmean. In addition to the parameters that indicate the model's performance, we also evaluate the time during training and testing. After knowing the evaluation of the model, then the best-performing model is collected from each dataset and carried out a deep learning ensemble by combining the classification results from each model in each dataset and then combining them. A voting process is carried out to determine the class of each video.

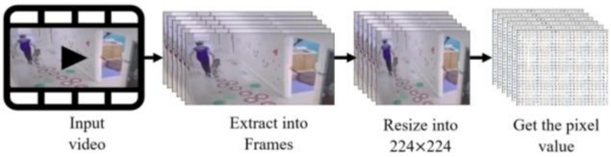


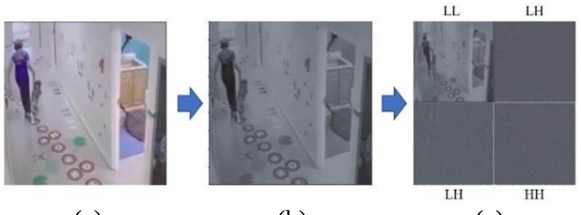
Figure. 2 Preprocessing illustration

The preprocessing process is extracting the video into a collection of images. The video data extract in some frames. Each image is resized to 224×224 to match the input size at the input layer when performing feature extraction using deep transfer learning. The following process is to get the pixel value of each image that will be used in the feature extraction process. In this process, a data matrix with dimensions $n \times 10 \times 224 \times 224 \times 3$ is generated, where n is the number of videos. An illustration of the preprocessing process for feature extraction usin deep transfer learning is shown in Fig. 2.

2.3 Feature extraction

2.3.1. Wavelet daubechies feature extraction

Discrete Wavelet Transform (DWT) is a linear transformation on a vector where each dimension is the square of an integer. Vectors are transformed to another space with the exact dimensions [26]. The



(a) (b) (c)

Figure. 3 Illustration of feature extraction using DWT: (a) before extracted features, (b) grayscale image, and (c) feature extraction result

principle of DWT outlines the input signal into two sub-signals: detail and approximation. The approximation corresponds to the low-frequency input signal with the most energy, and the detailed sub-band corresponds to the high-frequency input signal. This technique can be repeated at several levels by taking the approximation as the input signal [27]. Feature extraction using DWT is shown in Fig. 3.

This study use DWT with mother wavelet Daubechies 8. Fig. 3(a) shows one of the image frames from the video footage that has been converted to a size of 224×224. Prior to the feature extraction process using DWT, the frame is converted to grayscale, as shown in Fig. 3(b). The grayscale frame will be extracted and will produce four sub-band values as in Fig. 3(c). The resulting sub-bands are low-low (LL) coefficients containing approximation coefficients, low-high (LH) coefficients containing horizontal coefficients, high-low (HL) coefficients containing vertical coefficients, and high-high (HH) coefficients or detail coefficients. The sub-band used for the classification process is LL or approximation coefficient. In this process, a data matrix with dimensions is generated $n \times 10 \times 119 \times 119$ with n the number of videos.

2.3.2. Visual geometry group-16

Deep transfer learning is the process of transferring neural network parameters trained from one dataset for a particular task to another problem with different datasets and tasks [28]. This study uses the Visual Geometry Group-16 model as feature extraction. VGG16 has about 138 million parameters which causes this model to require a high computational evaluation and use a lot of memory and parameters. VGG16 consists of 5 convolutional layer blocks and three Full Connected layers and requires $224 \times 224 \times 3$ input. Each convolutional layer consists of several sub convolutional layers and a pooling layer. The feature extraction process using VGG 16 is illustrated in Fig. 4, and further explanation of the architecture in Fig. 4 can be seen in Table 1.

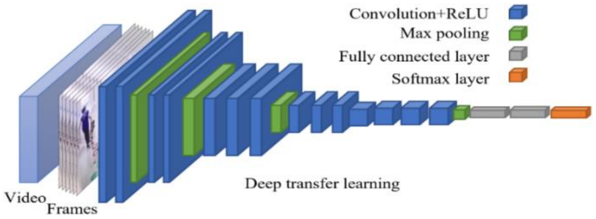


Figure. 4 VGG-16 architecture

Table 1. VGG-16 Architectural Components

Decription	Layers	Output shape	Parameter
Input	Input	(None, 224, 224, 3)	0
Block 1	2D Convolution	(None, 224, 224, 64)	1792
	2D Convolution	(None, 224, 224, 64)	36928
	2D Max Pooling	(None, 112, 112, 64)	0
Block 2	2D Convolution	(None, 112, 112, 128)	73856
	2D Convolution	(None, 112, 112, 128)	147584
	2D Max Pooling	(None, 56, 56, 128)	0
Block 3	2D Convolution	(None, 56, 56, 256)	295168
	2D Convolution	(None, 56, 56, 256)	590080
	2D Convolution	(None, 56, 56, 256)	590080
	2D Max Pooling	(None, 28, 28, 256)	0
Block 4	2D Convolution	(None, 28, 28, 512)	1180160
	2D Convolution	(None, 28, 28, 512)	2359808
	2D Convolution	(None, 28, 28, 512)	2359808
	2D Max Pooling	(None, 14, 14, 512)	0
Block 5	2D Convolution	(None, 14, 14, 512)	2359808
	2D Convolution	(None, 14, 14, 512)	2359808
	2D Convolution	(None, 14, 14, 512)	2359808
	2D Max Pooling	(None, 7, 7, 512)	0
	Flatten	(None, 25088)	0
	Fully Connected	(None, 4096)	102764544
	Fully Connected	(None, 4096)	16781312
Predictions	Softmax	(None, 1000)	4097000
Total Parameters			138357544

Fig. 4. shows the process of a video in feature extraction. The images resulting from preprocessing are processed through the VGG16 neural network, as shown in Table 1. The data will go through the first block on the VGG16 architecture. This first block contains two convolutional layers and one max-pooling layer. When going through the convolutional layer, the convolution process is carried out using a kernel 3 × 3 with stride 1. Stride is a neural network filter parameter that modifies the amount of movement on the image during the convolution process, as for the max-pooling layer, kernel 2 × 2 with stride 2. After going through the first block, a matrix of size $n \times 112 \times 112 \times 64$ was obtained, where n is the number of videos.

As in the first block, the second and third blocks also contain two convolutional neural networks and one max pooling layer. For the fourth and fifth blocks, three convolutional layers are used. After going through the five blocks, a matrix of sizes $n \times 7 \times 7 \times$

Figure. 5 Image of the 1st feature to 200th feature of transfer learning results

512 was obtained. The results of the fifth block will go through a flattened layer. In this layer, the resulting matrix is converted into a row matrix for each data so that a matrix of size is $n \times 25088$ obtained. In the last process, the data goes through two fully connected layers, which produce a matrix with dimensions of 20 × 4096.

Suiting the needs of this research, we do not use the prediction layer/softmax layer, which is a fully connected layer and is used to classify 1000 classes. Instead, we use another classification method. Therefore, the transfer learning results obtained have dimensions of 10 × 4096. The number 10 represents the number of images taken in one video. From Table 1 we know that the VGG16 architecture used as a parameter of 138357544. Because it does not use a softmax layer, the number of parameters used is reduced to 134260544. The results of transfer learning are depicted in Fig. 5.

2.4 Classification

2.4.1. Support vector machine

Support Vector Machine (SVM) is a classifier that works on the Structural Risk Minimization principle, which Vapnik and Chervonenkis first introduced in 1992. The learning process on the Support Vector Machine is looking for a support vector to obtain the best hyperplane. The advantage of SVM is that this algorithm provides a unique solution because it solves the convex optimization problem [29].

Most problems in real life are non-linear problems. Non-linear SVM can be an alternative problem solver. We have to find a non-linear transformation so that the data can be mapped to a high-dimensional feature space where the classification becomes linear. The transformation's selection is made so that the dot product leads to a kernel function $K(x, x_i)$ as in Eq. (1), which allows us to write the decision function as Eq. (2), with Lagrange multipliers $\beta_i \geq 0$ and b as bias. We can directly use kernel functions for non-linear problems called kernel tricks [30]. In this research, three kernel functions are used: linear kernel, Radial Basis Function (RBF) kernel, and polynomial kernel.

$$K(x, x_i) = \phi(x) \cdot \phi(x_i) \tag{1}$$

$$f(x) = sgn\left(\sum_{i=1}^{l} \beta_i y_i K(x, x_i) + b\right) \tag{2}$$

2.4.2. Long short term memory

A recurrent Neural Network (RNN) is one of the neural network algorithms that can store previous inputs in memory. Like RNN, Long Short Term Memory also has recurrent connections so that the previous neuron status is used as consideration for formulating output. However, LSTM has a unique formulation, namely memory cells that can store information for a long time. LSTM can also deal with vanishing gradient and exploding gradient problems that RNN cannot avoid during backpropagation optimization [31].

The main key of memory cells is the gate, which is a weighted function that regulates the flow of information in the cell. There are three types of gates, i.e., forget gate, which decides what information to remove from the cell, input gate determines what information from input data is used for updating memory status, and output gates that make decisions for output based on input data and memory cells [32]. LSTM implementation steps, namely:

i. Decide what information will be retained and forgotten. A sigmoid function achieves this by considering the previous state (output) (h_{t-1}) and the current input x_t and calculating the forget gate function f_t using Eq. (3), where W^f and U^f is the weight. If $f_t = 1$ then the information will be retained, if $f_t = 0$ then the information will be removed.

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{3}$$

ii. Updating the contents or contents of a memory cell. In this step, the new information will be selected, stored in the status cell. At the gate input, there are two parts of the sigmoid function choose which value will be updated and the function tanh create vector values new candidate $\hat{C}$ by providing a weight on the value selected based on the level of importance with the range of -1 to 1. After that, both combined to update the status. $\hat{C}$ combined with C_{t-1} to update C_t. Output H_t calculated based on the output function of the sigmoid and tanh function of C_t, which $\hat{C}_t$ is a candidate and the memory cell, W^i, W^g is the weighting parameter. Calculation i_t and $\hat{C}_t$ shown in Eqs. (4) and (5).

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{4}$$

$$\hat{C}_t = \tanh(x_t U^g + h_{t-1} W^g) \tag{5}$$

iii. In the next step the old status cell C_{t-1} is actualized with C_t by multiplying the old status cell by f_t to forget irrelevant information and then adding it with $\hat{C}_t$. This process represents a new candidate value that is scaled based on how many are selected to update each status value, represented by Eq. (6).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \hat{C}_t \tag{6}$$

iv. The final step is to calculate the output. First, the sigmoid function selects the relevant part of the cell state to be transmitted to the output. The state cell is then passed through and multiplied by the sigmoid gate output, so only the portion selected for output is saved. At the end of the cycle, units of the hidden layer h_t representing the output cycle and memory status used the next cycle. Calculation o_t and $h_t$shown in Eqs. (7) and (8).

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{7}$$

$$h_t = \tanh(C_t) \cdot o_t \tag{8}$$

In Briefly, in LSTM, the three gates are trained to learn what information can be retained in memory, how long it can be stored, and when it can be used. Combining multiple memory cells into a block allows them to share the same gate.

2.4.3. Bidirectional long short term memory

The basic idea of Bidirectional LSTM comes from the Bidirectional Recurrent Neural Network (BiRNN) [33]. BiRNN presents each data train sequence forward and backward into a different recurrent network, both of which are connected to the same output layer [34]. The output at any time t depends not only on the previous input, but also on the next input [35]. The hidden layer in the bidirectional LSTM stores two values: the forward calculation and the value for the reverse calculation [36].

Connections in the layer forward are equivalent to multiple stacked LSTM networks, which computes the sequence $\left(\vec{h}_t^L . \vec{c}_t^L\right) from\, t = 1\, to\, T$. On the other hand, in the forward layer the $\left(\overleftarrow{h}_t^L . \overleftarrow{c}_t^L\right)$ and the output is from iterated $t = T\, to\, 1$. Therefore, the mathematical expression of the Lth LSTM backward layer, at time t, can be written in Eqs. (9) to (14), the notation used same as the LSTM but in a different direction where the terms W and b express the weight and bias of the corresponding gate respectively.

$$\overleftarrow{f}_t^L = \sigma\left(W_{\overleftarrow{f}_h}^L h_{t+1}^L + W_{\overleftarrow{f}_x}^L h_t^{L-1} + b_{\overleftarrow{f}}^L\right) \tag{9}$$

$$\overleftarrow{i}_t^L = \sigma\left(W_{i_h}^L h_{t+1}^L + W_{\overleftarrow{i}_x}^L h_t^{L-1} + b_i^L\right) \tag{10}$$

$$\overleftarrow{\tilde{c}}_t^L = \tanh\left(W_{\overleftarrow{\tilde{c}}_h}^L h_{t+1}^L + W_{\overleftarrow{\tilde{c}}_x}^L h_t^{L-1} + b_{\overleftarrow{\tilde{c}}}^L\right) \tag{11}$$

$$\overleftarrow{c}_t^L = \overleftarrow{f}_t^L \cdot \overleftarrow{\tilde{c}}_{t+1}^L + \overleftarrow{i}_t^L \cdot \overleftarrow{\tilde{c}}_t^L \tag{12}$$

$$\overleftarrow{o}_t^L = \sigma\left(W_{\overleftarrow{o}_h}^L h_{t+1}^L + W_{\overleftarrow{o}_x}^L h_t^{L-1} + b_{\overleftarrow{o}}^L\right) \tag{13}$$

$$\overleftarrow{h}_t^L = \overleftarrow{o}_t^L \cdot \tanh(\overleftarrow{c}_t^L) \tag{14}$$

The output of BiLSTM can be expressed in Eq. (15) which $W_{\vec{h}y}$, $W_{\vec{h}y}$ are the weight, b_y is bias, $\vec{h}_t$ is the output of layer forward and $\overleftarrow{h}_t$ the backward output of layer [37].

$$y_t = W_{\vec{h}y} \vec{h}_t + W_{\vec{h}y} \overleftarrow{h}_t + b_y \tag{15}$$

2.4.4. Gated recurrent unit

GRU is a simplified version of the LSTM that can achieve the same results as the LSTM, using fewer parameters consisting of two gates, namely the reset gate and the update gate [38]. The update gate controls the previous information brought to the current layer, while the reset gate decides the amount of information to be discarded [39]. The equations for the update gate z_t, reset gate r_t, current memory state $\tilde{h}_t$, and final memory h_t are in Eqs. (16) to (19).

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{16}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{17}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b) \tag{18}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{19}$$

Signs $\odot$ represent the Hadamard product. W_z, W_r, and W is a weighted matrix, x_t is the input vector, $h_{t-1}$is previous hidden state, σ and tanh activation function is sigmoid and tanh, b_z, b_r, and b are biased.

2.4.5. Ensemble decision fusion learning

The ensemble is one type of Ensemble Learning. This method combines various models or classifiers and various types of datasets to increase the predictive power and accuracy of the classification model. The training process for making classification models takes up many resources and requires various experiments with various parameters to produce a good performance. When all the train data are combined, it will take longer. If we have several data sets for the same case, we have also done experiments and got the best model. It would be a pity if the resulting model was ignored so that we had to retrain with a new batch of models. This study uses ensemble decision fusion learning to maximize the classification results by utilizing experimental models with hockey, movie, and crowd datasets. We combined the new model's results with the child dataset to detect child abuse in video surveillance. The decision fusion ensemble we use is majority voting. This method is a powerful way to improve the prediction accuracy of classification models [20]. The ensemble method applied in a study conducted by Saloni Kumari to detect diabetes mellitus showed that it could increase accuracy [21]. In addition, this ensemble method can improve model performance when applied to malware detection, as done in [22].

Majority voting is the simplest but most powerful way and minimizes bias. Majority voting is done by combining the classification results from each classifier and then making decisions based on the most votes in the voting. In the case of binary classification, the number of models combined in this learning ensemble is odd to avoid a balanced number of votes. If a classifier makes an error, this method attempts to complement it with another member of the ensemble, which throws the error on a different object [40]. Each ensemble method requires an appropriate decision-combination strategy to combine the results from a single classifier to produce a final predictive model. The final prediction results are usually determined by majority voting, which refers to hard voting [41]. Each prediction made by base learning is counted as a vote and the combined prediction is determined by a majority vote [42]. Hard voting can be defined mathematically in Eq. (20) which determines the mode of the single classifier result, with y_i the final prediction label of the i-th data and $c_1, c_2, \dots, c_k$ the predicted result of single classifier-1 to single classifier-k.

$$y_i = mode\{c_1, c_2, \dots, c_k\} \tag{20}$$

In this study, training and testing will be carried out on several datasets, the best model from each dataset These are then combined to improve model performance on child datasets using stacking ensemble learning. The decision-making process is done by decision fusion majority voting. The ensemble learning steps carried out in this study will be described as follows:

Step 1. Each dataset is divided into training and test data using k fold validation.

Step 2. The training data from each dataset will be used as input for every single classifier for the training process. Before the training process, the data is preprocessed and feature extracted using the VGG16 algorithm.

Step 3. Test the test data on every single classifier of each dataset.

Step 4. Evaluate the performance of a single classifier on each dataset.

Step 5. The best-performing model from each dataset is selected to be combined in the learning ensemble. The number of models selected must be odd to avoid an even number of votes.

Step 6. The results of the testing of each selected model are used as input to ensemble learning.

Step 7. Conduct the voting process by assigning a class to each data based on the most votes.

Step 8. Evaluate the performance of the learning ensemble.

3. Results and discussion

The authors developed a model to detect violence against children on video data in this research. For initialization, a model is developed using hockey, movie, and crowd dataset [43-46]. Then to improve performance, a deep learning ensemble is carried out on the child dataset. The child dataset dan experimental setup is explained at the beginning of this section, followed by experiment results.

3.1 Dataset and experimental set up

This study compiled a dataset collected manually from several online platforms. We download videos that contain violence against children and some videos that do not contain violence. Videos containing acts of violence against children consist of abuse by parents or caregivers. For videos that do not contain acts of violence, we take videos of children playing. Figure 6. represents the sample image from the video in the child dataset. The collected videos were trimmed into a similar duration of 1-2 seconds. Then each of the video footage is manually labeled. This dataset consists of 332 videos, of which 145 are violent, and 189 are non-violent. This study uses 5-fold validation with a ratio of 4:1 for the distribution of the dataset. This research obtained 266 training and 66 videos as test data.

(a) (b)

Figure. 6 Sample images of child dataset with label: (a) violence and (b) non-violence

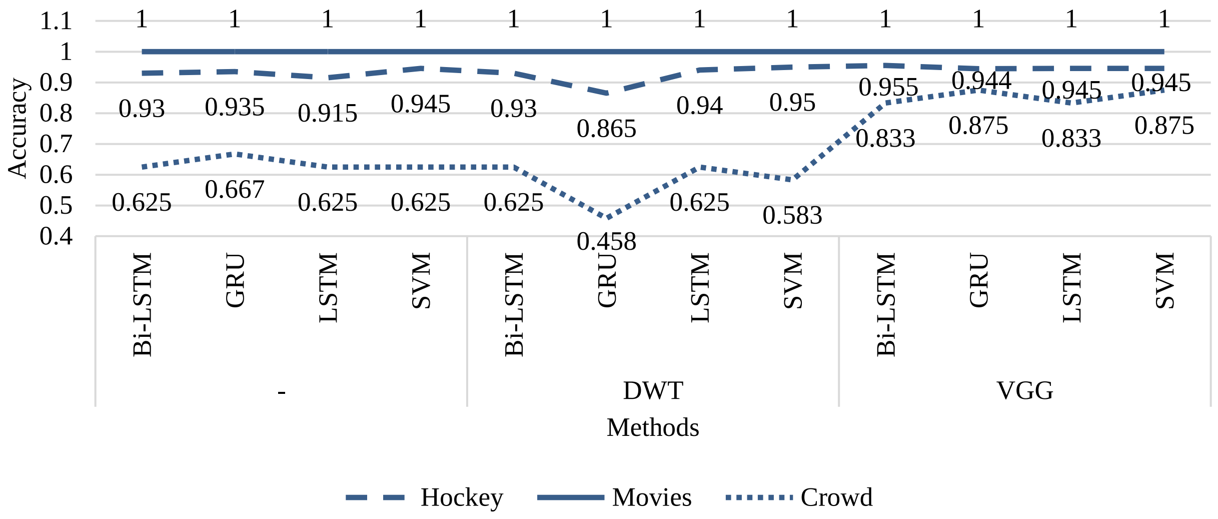


Figure. 7 Graph of accuracy on hockey, movies, and crowd dataset

3.2 Experimental results

Fig. 7 shows the accuracy of the hockey, movie, and crowd datasets. Based on Fig. 7, the best accuracy for the hockey dataset was obtained when using the VGG-BiLSTM combination method with a value of 0.955, which is better than the results obtained in [11, 23, 24]. The experimental results obtained for the movie dataset are outstanding. All combinations of algorithms can classify videos with an accuracy of 100%, and the author obtained the same result in [11], who was able to classify the movie dataset without anything wrong. This result is obtained because most of the videos focus on scenes of violence at close range, and there is rarely noise like the crowds in the crowd dataset.

In general, the best feature extraction method for hockey and crowd datasets is VGG16, with an average accuracy of 0.947 for hockey datasets and 0.854 for Crowd datasets. In addition to providing the best accuracy, the VGG16-SVM and VGG16-GRU methods also obtain the best Gmean on the crowd dataset, 0.874. It can be said that the two combinations of algorithms can effectively classify the two classes compared to others. The best initialization model for each dataset in Fig. 7 will be combined with the best model from the child dataset to enhance the ensemble learning method.

Table 2 and Fig. 8 show the experimental results on the child dataset. Wavelet-LSTM shows the best specificity in this dataset which is 0.947. In contrast, the accuracy and the Gmean are only 0.773 and 0.712. The G-Mean value is crucial to consider considering that this child dataset is unbalanced because the G-Mean value shows the performance to classify the two classes in a balanced way. The best accuracy when training on this dataset was obtained using a combination of the VGG16-LSTM and VGG16-GRU method, which was 0.909. In addition to getting the best accuracy, both algorithms also provide the best sensitivity and Gmean of 0.929 and 0.911.

Fig. 8 shows the time required to test the test data on the child dataset. The fastest average test time obtained using the VGG16 feature extraction method is 0.679 seconds. Judging from Figure 8, the testing time required for data without feature extraction takes a long time; it is different when combined with feature extraction methods. It shows that feature extraction speeds up testing time. Fig. 9 shows the confusion matrix and scatter plot of the child dataset. The confusion matrix displayed represents the combination of methods with the best accuracy from each classification algorithm. Fig. 8 and 9 imply that the VGG16 method used as feature extraction provides good accuracy and test time performance.

Table 2. Table of experimental results child dataset

Method	F.Ext	Accu-racy	Sensi-tifity	Speci-ficity	G-Mean
SVM	VGG18	0.894	0.893	0.895	0.894
	Wavelet	0.788	0.643	0.895	0.758
	-	0.803	0.679	0.895	0.779
LSTM	VGG16	0.909	0.929	0.895	0.911
	Wavelet	0.773	0.536	**0.947**	0.712
	-	0.742	0.786	0.711	0.747
BiLSTM	VGG16	0.894	0.857	0.921	0.889
	Wavelet	0.803	0.714	0.868	0.788
	-	0.773	0.857	0.711	0.780
GRU	VGG16	0.909	0.929	0.895	0.911
	Wavelet	0.833	0.714	0.921	0.811
	-	0.697	0.464	0.868	0.635
Ensembel Deep Learning	VGG16	**0.924**	**1**	0.868	**0.932**

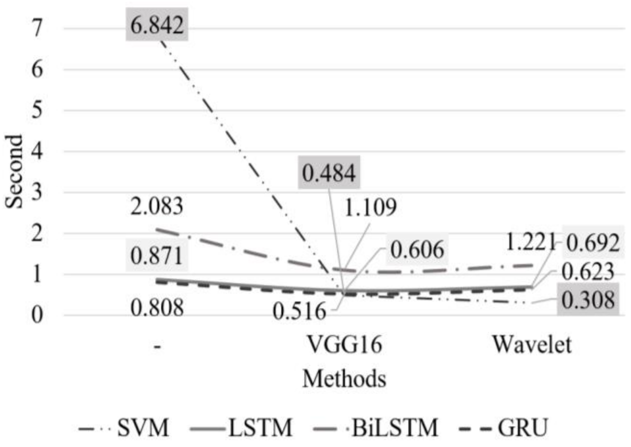


Figure. 8 Testing time on child dataset

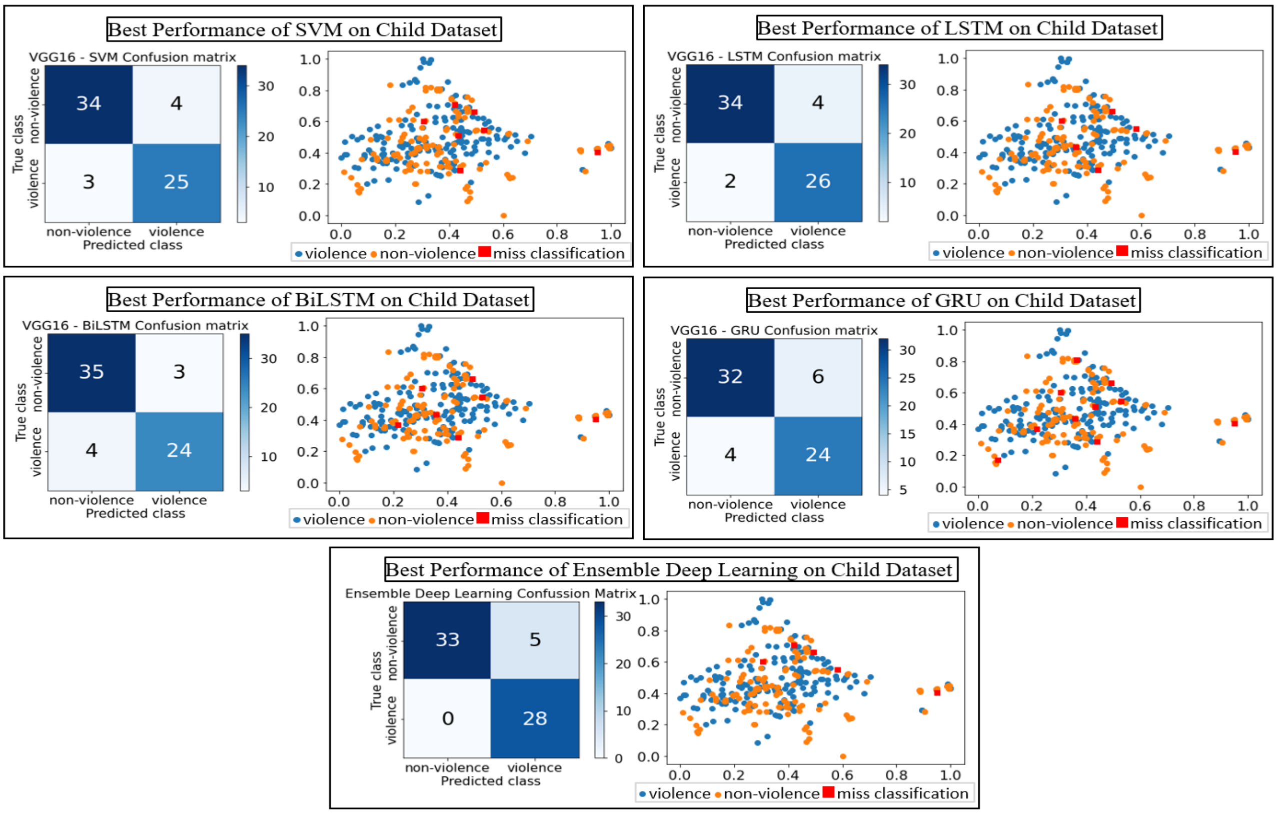


Figure. 9 Confusion matrix and scatter plot on child dataset

This study also conducted Ensemble Deep Learning to improve performance. The Ensemble Deep Learning method combines the best models from the Hockey, Crowd, Movie, and Child datasets. We use a combination of the best methods from the Hockey dataset, namely VGG16-BiLSTM, while from the Movie dataset, and two algorithms, namely VGG16-SVM and VGG16-GRU. Choosing the models is good performance for the movie dataset and the fastest testing time. The crowd dataset uses the model from VGG16-SVM and VGG-GRU, and the child training dataset uses the model from VGG16-LSTM and VGG16-GRU as ensemble learning models. The child seven selected models were applied to detect violence on the child dataset as many as 66 videos. Testing results from each model are then collected, and a voting process to determine the class of each video.

The Ensemble Deep Learning method can increase accuracy by 0.015 compared to the best when using only one method combination. In addition, the sensitivity value obtained also increased by 0.071 to 1, and the Gmean value increased from 0.911 to 0.932. The time required to test this ensemble method is 11,119 seconds, longer than when using only one algorithm combination.

However, this testing time is still within the limit, given that the performance provided is better when compared to using only one method combination. The ensemble learning method's scatter plot and confusion matrix is in Fig. 9.

4. Conclusion

This study explores some methods to detect violence from surveillance video. We have compiled the dataset containing video footage of acts of violence against children. In addition to using the compiled dataset, experiments also use three other datasets: the hockey dataset, the movie dataset, and the crowd dataset. The best accuracy and G-Mean obtained for the hockey dataset is 0.955 using VGG16- BiLSTM. The combination of VGG16-SVM and GRU provides the best accuracy and G-Mean for crowd datasets, although SVM requires less testing time with a time difference of 0.33 seconds. GRU is more memory efficient because it does not need to store kernel gram matrix. The VGG16-LSTM shows the best accuracy in the child dataset, with G-mean results reaching 0.909 and 0.911. All the algorithms provide excellent performance for the movie dataset.

Overall, feature extraction using deep learning gives outstanding performance when combined with all algorithms in performance and time. The results obtained showed an increase; the accuracy value increased by 0.015 to 0.924, the sensitivity value increased by 0.071 to 1, and the Gmean value increased by 0.021 to 0.932. The highest Gmean and sensitivity were obtained from all datasets with accuracy results when VGG16 was a feature extraction method. VGG16 is also able to increase accuracy up to 0.417. An ensemble deep learning method is used to improve the performance of the child dataset, an. The classification combined the best models from each dataset with VGG as the extraction feature. The testing time required for this deep learning ensemble method is longer, 11,119 seconds. The proposed method shows good results but still needs to be developed to improve accuracy and faster. Studies can be designed and implemented in real-time CCTV violence detection in the future.

Conflicts of Interest

In accordance with IJIES policy and our ethical obligation as researchers, we are reporting that We do not have a financial and/or business interests related to this topics, and do not receive funding from a company that may be affected by the research reported in the enclosed paper. I have disclosed those interests fully to IJIES, and have in place an approved plan for managing any potential conflicts arising from this arrangement. We have no conflicts of interest to disclose.

Author Contributions

The authors contribution as follow: Conceptualization, methodology, EMI; software, EMI and K; validation, EMI and K; formal analysis, EMI; investigation, EMI; data curation, K; writing—original draft preparation, K and EMI; writing—review and editing, K and EMI; visualization, K; supervision, EMI.

Acknowledgments

This research was supported by the Grant of Higher Education Excellence Research on Applied Research with approval no. B/12126/UN38.9/LK.04.00/2021 and No: 193/SP2H/AMD/LT/DRPM/2021, funded by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia. Implementation of research in the second year of DRPM funding in 2022.

References

[1] World Health organization (WHO), "Global status report on preventing violence against children 2020", 2020. [Online]. Available: https://www.who.int/publications-detail-redirect/9789240004191.

[2] S. M. Hammami and M. Alhammami, "Vision-based system model for detecting violence against children", *MethodsX*, Vol. 7, pp. 104-108, 2020.

[3] World Health organization (WHO), "Violence against children", 2020. https://www.who.int/news-room/fact-sheets/detail/violence-against-children (accessed Nov. 11, 2021).

[4] United Nations Children's Fund, "Protecting Children from Violence in the Time of COVID-19 : Disruptions in prevention and response services", 2020. [Online]. Available: https://www.unicef.org/media/74146/file/Protecting-children-from-violence-in-the-time-of-covid-19.pdf.

[5] World Health organization (WHO), "Addressing Violence Against Children, Women and Older People During The COVID-19 Pandemic: Key Actions," 2020. [Online]. Available: https://www.jstor.org/stable/pdf/resrep28197.pdf.

[6] S. Das, A. Sarker, and T. Mahmud, "Violence Detection from Videos using HOG Features", No. December, pp. 20-22, 2019.

[7] K. Deepak, L. K. P. Vignesh, and S.

Chandrakala, “Autocorrelation of gradients based violence detection in surveillance videos”, *ICT Express*, Vol. 6, No. 3, pp. 155-159, 2020.

[8] P. Yadav, P. Regundwar, A. Wyawahare, P. Pawar, and J. Madake, “An Intelligent System to Detect Violent Mob Activities”, In: *Proc. of 2020 IEEE 17th India Counc. Int. Conf. INDICON 2020*, 2020.

[9] B. Peixoto, B. Lavi, P. Bestagini, Z. Dias, A. Rocha, and P. Milano, “Multimodal Violence Detection in Videos”, In: *Proc. of ICASSP 2020 - 2020 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2957-2961, 2020.

[10] S. Akti, G. A. Tataroglu, and H. K. Ekenel, “Vision-based Fight Detection from Surveillance Cameras”, In: *Proc. of 2019 9th Int. Conf. Image Process. Theory, Tools Appl. IPTA 2019*, 2019.

[11] A. Jain and D. K. Vishwakarma, “Deep neuralNet for violence detection using motion features from dynamic images”, In: *Proc. of 3rd Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2020*, No. Icssit, pp. 826-831, 2020.

[12] M. Perez, A. C. Kot, and A. Rocha, “Detection Of Real-World Fights In Surveillance Videos”, In: *Proc. of ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2662-2666, 2019.

[13] C. Cappa and I. Jijon, “COVID-19 and violence against children: A review of early studies”, *Child Abus. Negl.*, Vol. 116, No. P2, p. 105053, 2021.

[14] L. J. Thun, P. L. Teh, and C. B. Cheng, “CyberAid: Are your children safe from cyberbullying?”, *J. King Saud Univ. - Comput. Inf. Sci.*, 2021.

[15] T. Han, J. Zhang, Z. Zhang, G. Sun, L. Ye, H. Ferdinando, E. Alasaarela, T. Seppänen, X. Yu, and S. Yang, “Emotion recognition and school violence detection from children speech”, *Eurasip J. Wirel. Commun. Netw.*, Vol. 2018, No. 1, 2018.

[16] M. Khan, M. A. Tahir, and Z. Ahmed, “Detection of Violent Content in Cartoon Videos Using Multimedia Content Detection Techniques”, In: *Proc. of 21st Int. Multi Top. Conf. INMIC 2018*, pp. 1-5, 2018.

[17] M. M. Moaaz and E. H. Mohamed, “Violence Detection In Surveillance Videos Using Deep Learning”, *Informatics Bull. Fac. Comput. Artif. Intell. Helwan Univ.*, Vol. 2, No. 2, pp. 1-6, 2020.

[18] E. M. Imah, E. S. Dewi, and I. G. P. A. Buditjahjanto, “A Comparative Analysis of Machine Learning Methods for Joint Attention Classification in Autism Spectrum Disorder Using Electroencephalography Brain Computer Interface”, *Int. J. Intell. Eng. Syst.*, Vol. 14, No. 3, pp. 412-424, 2021, doi: 10.22266/ijies2021.0630.34.

[19] Karisma, E. M. Imah, and A. Wintarti, “Violence Classification Using Support Vector Machine and Deep Transfer Learning Feature Extraction”, *International Seminar on Intelligent Technology and Its Applications*, 2021, pp. 337-342, 2021.

[20] K. Raza, “Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule”, *U-Healthcare Monitoring Systems, Elsevier Inc.*, pp. 179-196, 2019.

[21] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier”, *Int. J. Cogn. Comput. Eng.*, Vol. 2, No. January, pp. 40-46, 2021.

[22] D. Gupta and R. Rani, “Improving malware detection using big data and ensemble learning”, *Computers and Electrical Engineering*, Vol. 86, Elsevier Ltd, p. 106729, 2020.

[23] S. Roshan, G. Srivathsan, K. Deepak, and S. Chandrakala, *Violence Detection in Automated Video Surveillance: Recent Trends and Comparative Studies*, 2020.

[24] M. Sharma and R. Baghel, *Video Surveillance for Violence Detection Using Deep Learning*, Vol. 37. 2020.

[25] F. J. R. Segador, J. A. Á. García, F. Enríquez, and O. Deniz, “ViolenceNet: Dense Multi-Head Self-Attention with Bidirectional Convolutional LSTM for Detecting Violence”, *Electronics*, Vol. 10, No. 13, p. 1601, 2021.

[26] J. Panyavaraporn and P. Horkaew, “DWT / DC T-based Invisible Digital Watermarking Schem e for Video Stream”, *2018 10th Int. Conf. Know l. Smart Technol.*, pp. 154-157, 2018, [Online]. Available: https://doi.org/10.1109/KST.2018.8426150.

[27] A. Zaarane, I. Slimani, A. Hamdoun, and I. Atouf, “Real-Time Vehicle Detection Using Cross-Correlation and 2D-DWT for Feature Extraction”, *J. Electr. Comput. Eng.*, Vol. 2019, 2019.

[28] E. Rezende, G. Ruppert, T. Carvalho, A. Theophilo, F. Ramos, and P. D. Geus, “Malicious Software Classification Using VGG16 Deep Neural Network’s Bottleneck Features”, *Adv. Intell. Syst. Comput.*, Vol. 738, pp. 51-59, 2018.

[29] M. C. Frunza, “Support Vector Machines”, *Solving Modern Crime in Financial Markets*,

Vol. 2, pp. 205-215, 2016.

[30] X. S. Yang, "Support vector machine and regression", *Introduction to Algorithms for Data Mining and Machine Learning*, pp. 129-138, 2019.

[31] J. Li and Y. Shen, "Image describing based on bidirectional LSTM and improved sequence sampling", *2017 IEEE 2nd Int. Conf. Big Data Anal. ICBDA 2017*, pp. 735-739, 2017.

[32] J. Brownlee, *Long Short-Term Memory Networks With Python*, Vol. 1, No. 1. 2017.

[33] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Stacked bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction", *arXiv*, pp. 1-11, 2018.

[34] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures", *Neural Networks*, Vol. 18, No. 5-6, pp. 602-610, 2005.

[35] E. Ditsanthia, L. Pipanmaekaporn, and S. Kamonsantiroj, "Video Representation Learning for CCTV-Based Violence Detection", *TIMES-iCON 2018 - 3rd Technol. Innov. Manag. Eng. Sci. Int. Conf.*, pp. 1-5, 2019.

[36] S. Yang, "Research on network behavior anomaly analysis based on bidirectional LSTM", In: *Proc. of 2019 IEEE 3rd Inf. Technol. Networking, Electron. Autom. Control Conf. ITNEC 2019*, No. Itnec, pp. 798-802, 2019.

[37] Y. Yu, X. Si, C. Hu, and J. Zhang, "Hierarchical Reinforcement Learning for Sequencing Behaviors", *Neural Comput.*, Vol. 31, No. 7, pp. 1235-1270, 2018.

[38] A. Sen and K. Deb, "Categorization of actions in soccer videos using a combination of transfer learning and Gated Recurrent Unit", *ICT Express*, 2021.

[39] S. Zhang, M. A. Aty, Y. Wu, and O. Zheng, "Modeling pedestrians' near-accident events at signalized intersections using gated recurrent unit (GRU)", *Accid. Anal. Prev.*, Vol. 148, No. July, 2020.

[40] Y. Yang, "Ensemble Learning", *Temporal Data Mining Via Unsupervised Ensemble Learning*, pp. 35-56, 2017.

[41] A. Mohammed and R. Kora, "An effective ensemble deep learning framework for text classification", *J. King Saud Univ. - Comput. Inf. Sci.*, No. xxxx, 2021.

[42] R. Cai, T. Han, W. Liao, J. Huang, D. Li, A. Kumar, and H. Ma, "Prediction of surface chloride concentration of marine concrete using ensemble machine learning", *Cem. Concr. Res.*, Vol. 136, No. April, p. 106164, 2020.

[43] R. Nievas, E. Bermejo, Suarez, O. Deniz, Garcia, G. Bueno, and Sukthankar, "Hockey Fight Detection Dataset", *Computer Analysis of Images and Patterns*, 2011. https://academictorrents.com/details/38d9ed996a5a75a039b84cf8a137be794e7cee89 (accessed Mar. 03, 2021).

[44] E. B. Nievas, O. D. Suarez, G. B. Gracia, and R. Sukthankar, "Movies Fight Detection Dataset", 2011. https://academictorrents.com/details/70e0794e2292fc051a13f05ea6f5b6c16f3d3635 (ac cessed Jun. 23, 2021).

[45] I. S. Gracia, O. D. Suarez, G. B. Garcia, and T. K. Kim, "Fast fight detection", *PLoS One*, Vol. 10, No. 4, pp. 1-19, 2015.

[46] T. Hassner, Y. Itcher, and O. K. Gross, "Violent -Flows Database", 2012. https://www.openu.ac.il/home/hassner/data/violentflows/ (accessed Ju n. 23, 2021).