ICIC Express Letters
Part B: Applications
Volume **13**, Number **10**, October **2022**

ICIC International ©2022 ISSN 2185-2766
pp. **1027–1035**

FUSION LIGHTWEIGHT CONVOLUTIONAL NEURAL NETWORKS AND SEQUENCE LEARNING ARCHITECTURES FOR VIOLENCE CLASSIFICATION

Wimolsree Getsopon and Olarik Surinta*

Multi-agent Intelligent Simulation Laboratory (MISL)
Department of Information Technology
Faculty of Informatics
Mahasarakham University
Khamriang Sub-District, Kantarawichai District, Mahasarakham 44150, Thailand
63011261001@msu.ac.th; *Corresponding author: olarik.s@msu.ac.th

Received February 2022; accepted April 2022

Abstract. *Stopping violent incidents in real life is more dangerous for ordinary people. It may harm people's lives. Calling the police is the best choice to stop the violence. We should have an automatic system to recognize violence and warn the police on time. This paper proposes a method to classify violent incidents from video. However, classification of violent videos faces many challenging problems, such as video length, quality, angles and orientations of the recording devices. The proposed method is called fusion MobileNets-BiLSTM architecture. In the first part, we propose to use the lightweight MobileNetV1 and MobileNetV2 to extract the robust deep spatial features from the video so that only non-adjacent 16 frames were selected. The spatial features were transferred to the global average pooling, batch normalization, and time distribution. In the second part, the spatial features from the first part were concatenated and then sent to create the deep temporal features using the bidirectional long short-term memory (BiLSTM). The proposed fusion MobileNets-BiLSTM architecture was evaluated on the hockey fight dataset. The experimental results showed that the proposed method provides better results than the existing methods. It achieved 95.20% accuracy on the test set of the hockey fight dataset.*

Keywords: Violence classification, Lightweight convolutional neural networks, Recurrent neural networks, Sequence learning architectures, Fusion architectures

1. **Introduction.** Video surveillance systems are essential to save human life and reduce the risks of becoming a victim of crime [1]. A crime can happen anywhere and anytime, causing damage to life and property. Most public or private places have established video surveillance systems to monitor human activity and prevent crime. However, using human monitoring through video surveillance may not stop the incident. Therefore, applying computer vision technology to video surveillance systems is crucial to identify in real time and warn related agencies when an abnormal event occurs. The need is to recognize violent activities such as fighting, punching, and kicking from a person or crowd. It is imperative to understand video and efficiently apply it to the real world.

Nowadays, deep learning is developing rapid detection and recognition of violence in surveillance video. When comparing deep learning methods with traditional methods, deep learning methods have strong feature expression ability and minor limitations [2]. Some researchers have developed convolutional neural networks (CNNs) for performing violent video recognition [1,3,4]. Khan et al. [5] presented a violence detection approach using deep learning. The video was segmented into shots and selected representative frames with a maximum saliency score. Then, the selected frames were learned by a lightweight

DOI: 10.24507/icicelb.13.10.1027

deep learning model and classified them as violence or non-violence. Keçeli and Kaya [6] used a pre-trained CNN for deep high-level features extraction that applied an optical flow as the input of the network and classified violent activities by SVM and subspace k-nearest neighbor (SkNN). Karisma et al. [7] used a pre-trained VGG16 model for the feature extraction method and classified it using the support vector machine (SVM) algorithm with the linear kernel. VGG16 extracted 4,096 features and was used as the input to the SVM. The experimental results showed that the VGG16 combined with SVM achieved an accuracy of 96.4%.

Some studies have proposed combining CNN and LSTM networks with learning sequence data from video. Soliman et al. [8] proposed an end-to-end deep neural network model for recognizing violence in video. The VGG16 was used for spatial feature extraction, followed by LSTM for extracting the temporal features. Then, the fully connected and softmax layers were used as classification. Their method achieved the best accuracy of 95.10% on the hockey fight dataset. Ditsanthia et al. [9] proposed a new visual feature descriptor, called multi-scale convolutional features, to partition the video frame into different regions and extract deep features. Then, the features were pooled together to obtain a meaningful feature vector. Finally, the frame-level features were fed into the BiLSTM to classify violence from the video.

Carneiro et al. [10] focused on using a multi-stream of VGG-16 networks and investigating conceivable feature descriptors of a video, including spatial, temporal, rhythmic, and depth information. Then, the outputs were classified using the ensemble method. Peixoto et al. [11] proposed a fusion model based on visual and audio feature representation to tackle violence detection in video. First, the video frame features were extracted using C3D, CNN-LSTM, and InceptionV4, whereas the audio features were calculated using four standard audio feature extractor methods. Then, the different visual and audio features vectors were fused with a concatenation operation. Finally, a random forest and a softmax function were used as classifiers. The result showed that the classification accuracy increased 6% when combining visual and audio features. Lou et al. [12] proposed an autoencoder mapping method for auditory-visual information fusion, using a CNN-LSTM architecture for feature extraction. Then, the visual and auditory features were integrated into the same shared subspace using an autoencoder model. Next, the output from autoencoder mapping was combined with the concatenation method. Finally, the softmax function was used to identify violent behavior. The result showed that their proposed method improved the performance of violent behavior recognition.

In the above studies, CNN extracted only spatial features. However, information sent to create the deep learning model for video classification is insufficient [13], although many studies use the RNN architecture to learn from the sequence data and increase the performance of the violence recognition. Therefore, for the surveillance system to recognize more accurately, the feature-fusion method receives more attention because the combination of features can significantly improve the efficiency of violence recognition.

The main contributions of the proposed architecture are presented in the following. We proposed the lightweight MobileNets to extract the deep spatial features and bidirectional long short-term memory (BiLSTM), which is a recurrent neural network, to learn from the sequence video frames and extract the temporal features. We proposed the concatenating operation to combine the spatial features that were extracted using the MobileNetV1 and MobileNetV2 before sending the spatial features to the BiLSTM network. The softmax function was used as the classifier of the proposed architecture. Hence, we selected keyframes which were the only 16 non-adjacent frames. However, other methods were examined with 20 and 40 frames. In this paper, all 16 keyframes were input to the proposed fusion lightweight CNNs and sequence learning architecture. The output was classified as violence and non-violence.

This paper is organized as follows. Section 2 presents the proposed fusion lightweight CNNs and sequence learning architecture. The violence video dataset, experimental setup, and experimental results are presented in Section 3. The conclusion and future work are given in the last section.

2. Fusion Lightweight CNNs and Sequence Learning Architecture.

In this section, we present the fusion lightweight CNNs and sequence learning architecture to classify violent incidents from videos.

Overview of the architecture. We divided the proposed architecture into two main parts. For the first part, the deep spatial features are extracted from the violence videos using lightweight MobileNetV1 and MobileNetV2. In addition, we removed the two last layers of MobileNetV1 and V2 and replaced them with global average pooling (GAP), batch normalization (BN), and time distribution layers. Hence, the deep spatial features from MobileNetV1 and V2 were connected with the concatenating operation. For the second part, we proposed the bidirectional long short-term memory (BiLSTM), which is a sequence learning architecture, to learn from the sequence features and extract the robust temporal features. The framework of the proposed architecture is shown in Figure 1. The details of each part are described in the following sections.

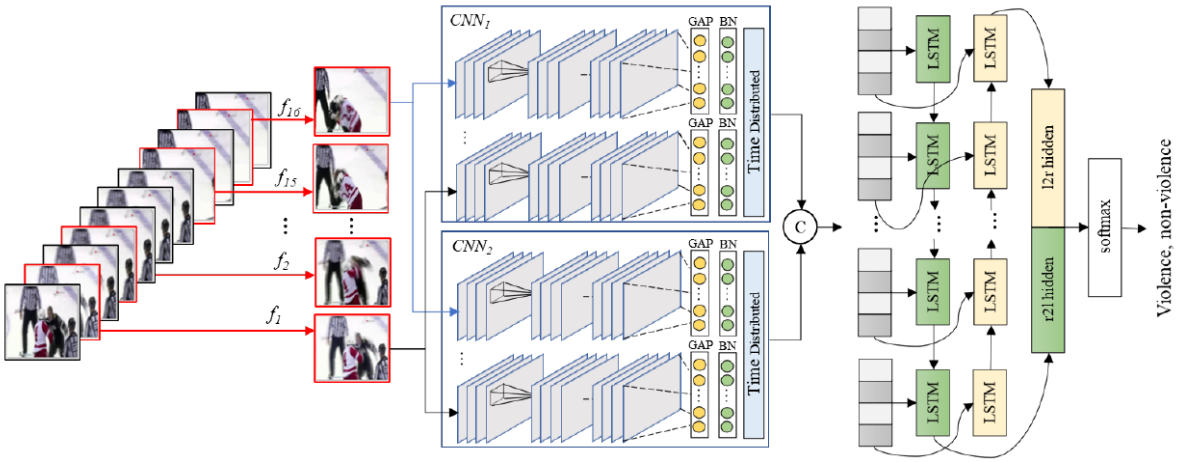

FIGURE 1. Illustration of the fusion lightweight MobileNets and BiLSTM architecture for violence video recognition

2.1. Convolutional neural network architectures.

The details of the CNN architectures proposed in our experiments are as follows.

MobileNetV1, which is the lightweight CNN architecture, has a small number of parameters because the depthwise separable convolution operation was invented [14]. Depthwise convolution was applied to each channel. Then, the pointwise convolution with a 1×1 convolution was performed to change the dimension and create a linear output. In the MobileNetV1 architecture, the depthwise separable convolution was attached to the convolution operation in every layer. Further, the BN and rectified linear unit (ReLU) activation function were combined after each convolution. The model of the MobileNetV1 is much smaller than VGG16 and GoogLeNet.

MobileNetV2 is the improved version of MobileNetV1. Two layers were added in the MobileNetV2 architectures: an inverted residual and a linear bottleneck, to enhance memory efficiency [15]. The inverted residual block contained a convolution layer, depthwise convolution, and convolution layer, respectively, with one stride. The shortcut connection was connected between each residual block the same way as in the residual network. The linear bottleneck block also contained the same layer as the inverted residual layer, but the stride was set as two.

NASNetMobile is the lightweight version of the NASNet. It was designed to explore the best convolutional layer on a small dataset, such as the CIFAR-10 dataset, and then transfer the best layer by stacking the layers together to a large dataset, such as ImageNet [16]. To search for the best convolutional layer, it searches from many sets of convolutional operations, for example, identity, 3×3 convolution, 3×3 depthwise convolution, 3×3 average pooling, and 3×3 dilated convolution, using a recurrent neural network (RNN). NASNet consisted of two main cells stacked together: normal and reduction cells. Although the normal and reduction cells were stacked together, the NASNet architecture could be adjusted by repeating many normal cells with N times.

ResNet50V2 is a modified version of ResNet50 that performs better than the original ResNet50 and ResNet101 on the ImageNet dataset [17]. The difference between the residual block in the original ResNet and the modification ResNetV2 is the number of the convolution operation. The original residual block contained the weight layer, BN, ReLU, weight layer, and BN, respectively. Before combined to the following layer, the ReLU function was performed. While the modified residual block in ResNetV2 contains BN, ReLU, weight layer, BN, ReLU, and followed by weight layer. Hence, it adds to the following layer without applying the ReLU function.

For our experiments, we removed the last two layers of each CNN architecture before extracting the deep spatial features.

2.2. **Sequence learning architectures.** In this study, the sequence information of 16 keyframes that were extracted from the violent video was first extracted using the CNNs and then transferred to the sequence learning architectures. The brief details of the sequence learning architectures are as follows.

Long short-term memory (LSTM) was designed by Hochreiter and Schmidhuber [18] to overcome the error of back-flow problems. LSTM has a memory block, which is a set of recurrently connected blocks, multiplicative units: input, output, and forget gates. The advantage of the LSTM network is that it was proposed to deal with long sequential data, including video, speech, and long text data. The gates were designed to keep or forget information while training the LSTM network. The LSTM learned from the sequence information and extracted the robust temporal features.

Bidirectional LSTM (BiLSTM) is a sequence learning architecture that processes sequence information in two directions [19]. It consists of two independent LSTM networks: forward state and backward state. The forward state takes the input in a forward direction. At the same time, the backward state takes in a backward direction. The outputs of the two states are connected to the same output.

Gated recurrent unit (GRU) was introduced by Cho et al. [20] and has the same function as the LSTM network. The previous sequence information is controlled by reset and update gates. The reset and update gates were designed to control the previous sequence information. Further, the update gate combined the input and forget gates into a single gate. The GRU network has fewer hyperparameters to adjust. Thus, it trains the model faster than the LSTM network [21].

3. Experimental Setup and Result.

3.1. **Violence video dataset.** We evaluated the proposed method on a benchmark violent video dataset that was collected from hockey games of the national hockey league (NHL) in North America, namely the hockey fight dataset [22]. The hockey fight dataset includes two classes and contains 500 violent videos and 500 without violence. Each hockey video consists of 41 frames with 720×576 pixels resolution. Examples of violent and non-violent videos are shown in Figure 2.

(a)

(b)

FIGURE 2. Some examples of (a) violent video and (b) non-violent video of the hockey fight dataset

3.2. **Experimental setup.** We implemented the proposed framework using Keras API based on the TensorFlow backend. All experiments were performed on Windows OS with Intel Core i9, 32GB of RAM, and NVIDIA RTX2070 GPU.

In the experimental setup, we first used a pre-trained model of four state-of-the-art CNN architectures to train on the hockey fight dataset, including MoblileNetV1, MobileNetV2, ResNet50V2, and NASNetMobile. The hyperparameters of the CNNs were set as follows: SGD optimizer, the momentum of 0.9, batch size of 4, and train with 100 epochs. We also performed different learning rates (0.01, 0.01, 0.001, 0.0001, and 0.00001) to find the lowest loss value while training. To extract the deep features, we then deleted the last layer of each architecture, which was the fully connected (FC) and softmax layers and replaced it with three layers: global average pooling (GAP), batch normalization (BN), and time distribution layers. Second, the deep features were sent to the recurrent neural networks (RNNs), including LSTM, GRU, and BiLSTM. The softmax function was used as a classifier. The hockey fight dataset was divided into training and test sets that contained 750 and 250 videos, respectively.

3.3. **Experiments with frames selection.** To show the performance of the CNN and RNN architecture on the hockey fight dataset, we proposed to use the MobileNetV2 architecture to train and extract deep features from all frames, which was 40 frames for each video. Subsequently, the deep features were combined with the LSTM network, called MobileNetV2-LSTM. We trained the MobileNetV2-LSTM model for 12 hours and 19 minutes. The result showed that it achieved 93.73% accuracy on the test set.

Existing violence recognition systems were designed to extract 16, 20, and 40 frames from the video [6,8-10]. In this experiment, we trained MobileNetV2-LSTM by choosing only 16 frames from the video. Consequently, we experimented on choosing the key frame from different frame numbers (see Table 1). As a result, the computational time was reduced and was three times faster than when training with 40 frames. It trained approximately four hours.

The accuracy results of different frame numbers are shown in Table 1. We compared four keyframe numbers (see Table 1, Experiments 1-4). It can be seen from Table 1 that frame numbers 5, 7, 9, ..., 35, which are 16 frames, are the best keyframes in our experiments on the hockey fight dataset. It obtained 88.80% on the test set.

TABLE 1. Experimental results with different frames using MobileNetV2-LSTM

Experiments	Frame numbers	Accuracy (%)
1	1-16	83.20
2	13-28	87.60
3	25-40	88.00
4	5, 7, 9, ..., 35	**88.80**

Discussion of experiments with frames selection. We found that the best performance was obtained when selecting non-adjacent frames. However, when the non-adjacent frames were selected, the CNN-LSTM model was trained from the redundant information. For the hockey fight dataset, we then selected every two frames. Also, training the CNN-LSTM model using 16 keyframes was much faster than training with the whole frames. An example of the adjacent and non-adjacent frames is illustrated in Figure 3.

(a)

(b)

FIGURE 3. Illustration of the (a) adjacent and (b) non-adjacent frames of the hockey fight dataset

3.4. **Experiments with different CNN architectures.** As with the experimental results described above, the best frames were selected from the frames selection experiment, including 16 frames of frame numbers 5, 7, 9, ..., 35. We evaluated the performance of the CNNs and LSTM using four state-of-the-art CNN architectures: MobileNetV1, MobileNetV2, ResNet50V2, and NASNetMobile. The different learning rates were examined and only the best learning rate was reported for each CNN in this experiment. For evaluation, the training set was used for 5-fold cross-validation (5-CV) to avoid overfitting and the test set was for final evaluation.

We present the experimental results with various CNN architectures combined with the LSTM network in Table 2. MobileNetV2-LSTM achieved an accuracy of 92.76% with cross-validation on the hockey fight dataset and 91.60% on the test set. Results also significantly outperformed the other CNN-LSTM models (t-test, $p < 0.05$). The MobileNetV2-LSTM spent around 21 minutes and 7 seconds for the training and test times, respectively. In contrast, the very deep networks (ResNet50V2 and NASNetMobile) performed worse on accuracy and computation.

Discussion. We found that the proposed CNN-LSTM architectures can address the overfitting problem because the accuracies of the 5-CV and test set were not different. With the MobileNetV2 architecture, a very small learning rate value was used to reach the

TABLE 2. The average accuracy (%) and the standard deviation of CNN architectures combined with the LSTM network obtained on cross-validation and test sets

Models	Learning rate	5-CV	Test accuracy (%)	Training time (∼mins)	Testing time (∼sec/video)
ResNet50V2-LSTM	0.01	77.33 ± 0.0472	77.60	33	11
NASNetMobile-LSTM	0.0001	82.67 ± 0.0550	87.60	31	34
MobileNetV1-LSTM	0.00001	92.00 ± 0.0354	**92.00**	22	5
MobileNetV2-LSTM	0.0001	**92.76 ± 0.0369**	91.60	21	7

lowest loss value. Further, the computational time decreased when the lightweight CNNs (MobileNetV1 and V2) were performed. In the following experiments, MobileNetV1 is proposed in combination with different RNN architectures: LSTM, BiLSTM, and GRU.

3.5. **Experiments with fusion MobileNets and RNN architectures.** To examine the effect of the combination between MobileNets and RNN architectures, we combine the deep features extracted using MobileNetV1 and MobileNetV2 with concatenating and adding operations. Then, the deep combination features were transferred to the RNN architectures and classifier with a softmax function (see Figure 1). Furthermore, the proposed model was trained with 1,000 epochs.

We present the accuracy results of the combined operations, including concatenating and adding, as shown in Table 3. We also compared the fusion MobileNet and RNN architecture results with the experiments in Section 3.4. The fusion MobileNet and RNN models outperformed the single CNN models by approximately 2% on the test set. However, they spent much more training time, because they had to train on both MobileNet architectures.

TABLE 3. The accuracy (%) and computational times of violence recognition experiments on the hockey fight dataset

Combined operations	Feature sizes	RNNs	Test accuracy (%)	Training time	Testing time (∼sec/video)
concatenating	16 × 2048	LSTM	94.80	3h:38m	3
		BiLSTM	**95.20**	8h:44m	5
		GRU	94.00	4h:6m	2
adding	16 × 1024	LSTM	94.80	3h:22m	2
		BiLSTM	94.80	7h:35m	4
		GRU	94.40	3h:36m	2

It can be seen from Table 3 that the concatenating operation created robust deep features with the size of 16×2048 and achieved better accuracy when combining MobileNet models with BiLSTM architecture. It achieved an accuracy of 95.20% on the test set of the hockey fight dataset. However, the adding operation created only 16 × 1024 deep features and achieved 94.80% accuracy when combined with RNNs. The performance was slightly decreased (only around 0.4%) when compared with concatenating operation. Most importantly, the testing time shown was almost equal.

Discussion of experiments with fusion MobileNets and RNN architectures. When using the combined operations: concatenating and adding, the deep feature sizes of the concatenating operation were larger one time than the adding operation. However, the training time was different, by about only one hour. We can use the fusion MobileNet and RNN architectures to classify violence from real time because it is recognized quickly and with high accuracy. So, extending the complex architecture does not affect the recognition time.

3.6. **Comparison of the fusion MobileNets and BiLSTM architecture and the existing methods.** This section presents the experimental results of various methods, as shown in Table 4.

Table 4. The comparison of the proposed method with existing methods

Methods	No. of frames	Data splitting Train:Test (%)	Testing accuracy (%)
Multiscale convolutional features [9]	40	80:20	83.19
Salient frame extraction and MobileNet [5]	N/A	75:25	87.00
Short-term traffic flow prediction [8]	20	80:20	88.20
Multi-stream CNN [10]	40	90:10	89.10
Optical flow and AlexNet [6]	20	80:20	94.40
Our proposed method	**16**	**75:25**	**95.20**

Table 4 compares the results of our proposed method with the existing methods on the hockey fight dataset. It shows that our proposed fusion MobileNets-BiLSTM architecture outperformed the existing methods with an accuracy of 95.20%. As a result, the existing method trained their models with more frames than our proposed method. The existing method trained with 20 and 40 frames, while our model trained with 16 frames. We also trained the model with less training set than the other methods, except research [5].

4. **Conclusions.** In this paper, we proposed the fusion MobileNets-BiLSTM framework to recognize violent events from the sport of hockey. We first selected MobileNetV1 and MobileNetV2, which are lightweight convolutional neural networks (CNNs), that aim to extract the robust deep features and then convert the deep features to perform with the bidirectional long short-term memory (BiLSTM) by adding three layers: the global average pooling, batch normalization, and time distribution. Second, the concatenating operation was proposed to fuse the robust deep features that are extracted by the lightweight MobileNets before transferring them to the BiLSTM network. For the hockey videos, we extracted video frames by selecting only 16 frames that were non-adjacent to avoid the proposed architecture training from the redundant information. Interestingly, the results showed that selection with the non-adjacent frames outperforms other selection frame methods. Furthermore, our results showed better accuracy than the results presented in existing works. The proposed fusion MobileNets-BiLSTM framework achieved an accuracy of 95.20% on the test set of the hockey fight dataset.

In future work, we first aim to reduce the training and testing time by decreasing the video frames. For this, we will study the instance selection method [23]. Second, we found that applying the optical flow [6] showed the appropriate results. We will also propose the optical flow method for selecting the non-adjacent frames.

Acknowledgment. This research project was financially supported by Mahasarakham University.

REFERENCES

[1] W. Lejmi, A. Ben Khalifa and M. Mahjoub, A novel spatio-temporal violence classification framework based on material derivative and LSTM neural network, *Traitement du Signal*, vol.37, no.5, pp.687-701, doi: 10.18280/ts.370501, 2020.

[2] J. Yang, F. Wang and J. Yang, A review of action recognition based on convolutional neural network, *Journal of Physics: Conference Series*, vol.1827, no.1, 12138, doi: 10.1088/1742-6596/1827/1/012138, 2021.

[3] D. Kreuter, H. Takahashi, Y. Omae, T. Akiduki and Z. Zhang, Classification of human gait acceleration data using convolutional neural networks, *International Journal of Innovative Computing, Information and Control*, vol.16, no.2, pp.609-619, doi: 10.24507/ijicic.16.02.609, 2020.

[4] A. F. Siregar and T. Mauritsius, Ulos fabric classification using android-based convolutional neural network, *International Journal of Innovative Computing, Information and Control*, vol.17, no.3, pp.753-766, doi: 10.24507/ijicic.17.03.753, 2021.

[5] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik and M. Y. Lee, Cover the violence: A novel deep-learning-based approach towards violence-detection in movies, *Applied Sciences*, vol.9, no.22, 4963, doi: 10.3390/app9224963, 2019.

[6] A. S. Keçeli and A. Kaya, Violent activity detection with transfer learning method, *Electronics Letters*, vol.53, no.15, pp.1047-1048, doi: 10.1049/el.2017.0970, 2017.

[7] 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 (ISITIA)*, pp.337-342, doi: 10.1109/ISITIA52817.2021.9502253, 2021.

[8] M. M. Soliman, M. H. Kamal, M. A. El-Massih Nashed, Y. M. Mostafa, B. S. Chawky and D. Khattab, Violence recognition from videos using deep learning techniques, *The 9th International Conference on Intelligent Computing and Information Systems (ICICIS)*, pp.80-85, doi: 10.1109/ICICIS46948.2019.9014714, 2019.

[9] E. Ditsanthia, L. Pipanmaekaporn and S. Kamonsantiroj, Video representation learning for CCTV-based violence detection, *The 3rd Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, pp.1-5, doi: 10.1109/TIMES-iCON.2018.8621751, 2018.

[10] S. A. Carneiro, G. P. da Silva, S. J. F. Guimarães and H. Pedrini, Fight detection in video sequences based on multi-stream convolutional neural networks, *The 32nd Conference on Graphics, Patterns and Images (SIBGRAPI)*, Rio de Janeiro, Brazil, pp.8-15, doi: 10.1109/SIBGRAPI.2019.00010, 2019.

[11] B. Peixoto, B. Lavi, P. Bestagini, Z. Dias and A. Rocha, Multimodal violence detection in videos, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.2957-2961, doi: 10.1109/ICASSP40776.2020.9054018, 2020.

[12] J. Lou, D. Zuo, Z. Zhang and H. Liu, Violence recognition based on auditory-visual fusion of autoencoder mapping, *Electronics*, vol.10, no.21, 2654, doi: 10.3390/electronics10212654, 2021.

[13] H. Chen, C. Hu, F. Lee, W. Yao, L. Chen and Q. Chen, A supervised video hashing method based on a deep 3D convolutional neural network for large-scale video retrieval, *Sensors*, vol.21, no.9, 3094, doi: 10.3390/s21093094, 2021.

[14] A. G. Howard, M. Zhu, B. D. Chen, W. Wang, T. Weyand, M. Andreetto and H. Adam, MobileNets: Efficient convolutional neural networks for mobile vision applications, *ArXiv*, arXiv:abs/1704.04861, 2017.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, MobileNetV2: Inverted residuals and linear bottlenecks, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.4510-4520, 2018.

[16] B. Zoph, V. Vasudevan, J. Shlens and Q. V. Le, Learning transferable architectures for scalable image recognition, *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.8697-8710, doi: 10.1109/CVPR.2018.00907, 2018.

[17] K. He, X. Zhang, S. Ren and J. Sun, Identity mappings in deep residual networks, *European Conference on Computer Vision (ECCV)*, Springer, vol.9980, doi: 10.1007/978-3-319-46493-0_38, pp.630-645, 2016.

[18] S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Computation*, vol.9, pp.1735-1780, doi: 10.1162/neco.1997.9.8.1735, 1997.

[19] A. Graves and J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Networks*, vol.18, no.5, pp.602-610, doi: 10.1016/j.neunet.2005.06.042, 2005.

[20] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Computing Research Repository (CoRR)*, arXiv:abs/1406.1078, 2014.

[21] T. Toharudin, R. Pontoh, R. Caraka, S. Zahroh, Y. Lee and R. Chen, Employing long short-term memory and Facebook prophet model in air temperature forecasting, *Communication in Statistics-Simulation and Computation*, pp.1-24, doi: 10.1080/03610918.2020.1854302, 2021.

[22] E. B. Nievas, O. Déniz-Suárez, G. García and R. Sukthankar, Violence detection in video using computer vision techniques, *International Conference on Computer Analysis of Images and Patterns (CAIP)*, Springer, Berlin, Heidelberg, pp.332-339, doi: 10.1007/978-3-642-23678-5_39, 2011.

[23] J. Olvera-López, J. Carrasco-Ochoa, J. F. Martínez-Trinidad and J. Kittler, A review of instance selection methods, *Artificial Intelligence Review*, vol.34, pp.133-143, doi: 10.1007/s10462-010-9165-y, 2010.