Lightweight mobile network for real-time violence recognition

Youshan Zhang¹, Yong Li²*, Shaozhe Guo¹

¹ Graduate Student Brigade, Chinese People’s Armed Police Force Engineering University, Xi’an, Shaanxi Province, China, ² College of Information Engineering, Chinese People’s Armed Police Force Engineering University, Xi’an, Shaanxi Province, China

* liyong@nudt.edu.cn

Abstract

Most existing violence recognition methods have complex network structures and high cost of computation and cannot meet the requirements of large-scale deployment. The purpose of this paper is to reduce the complexity of the model to realize the application of violence recognition on mobile intelligent terminals. To solve this problem, we propose MobileNet-TSM, a lightweight network, which uses MobileNet-V2 as main structure. By incorporating temporal shift modules (TSM), which can exchange information between frames, the capability of extracting dynamic characteristics between consecutive frames is strengthened. Extensive experiments are conducted to prove the validity of this method. Our proposed model has only 8.49MB parameters and 175.86MB estimated total size. Compared with the existing methods, this method greatly reduced the model size, at the cost of an accuracy gap of about 3%. The proposed model has achieved accuracy of 97.959%, 97.5% and 87.75% on three public datasets (Crowd Violence, Hockey Fights, and RWF-2000), respectively. Based on this, we also build a real-time violence recognition application on the Android terminal. The source code and trained models are available on https://github.com/1840210289/MobileNet-TSM.git.

Introduction

With the development of Internet technology [1], an increasing number of digital cameras have been deployed in public places to monitor the order of public places or deployed in a private space to protect personal safety. Irrespective of where the digital camera is deployed, it must always identify the video manually. This is difficult to achieve because of the huge cost of monitoring all real-time pictures. In recent years, the application of deep learning in the field of computer vision has been widely studied. Deep learning model can be trained to extract feature information and classify pictures instead of human beings; this has increasingly attracted attention.

Image classification refers to classifying images by extracting different feature information hidden in pixels, which is a basic problem in the field of computer vision, and it is also the basis of complex visual tasks such as behavior recognition, object detection, and image cutting.
As a specific task of behavior recognition, violence recognition also attracted the attention of many scholars. Its development process is similar to that of behavior recognition. Generally, it can be divided into two stages: manually designed descriptors and deep learning methods.

Early violence recognition methods mainly extracted features using manually designed descriptors. Based on the sensitivity of human vision to features, special descriptors were constructed to represent violent behaviors in videos and then classified using different machine learning algorithms. The following research classified violence by judging the presence of blood or abnormal sounds such as gunshots in the video [1–3]. Special descriptors, such as Violent Flows (ViF) [4, 5] and Oriented Violent Flows (OViF) [6], were designed to extract the characteristics of violent behavior, such as large range of action, short occurrence time and large change in movement direction. ViF considered statistics of how flow-vector magnitudes change over time. OViF pointed out that ViF can’t handle magnitudes with the same magnitude but opposite directions well. By depicting the information involving both of motion magnitudes and motion orientations, OViF achieved better performance. Scale Invariant Feature Transform (SIFT) [7], Histograms of Oriented Gradient (HOG) [8], and their variants [9–13] were also learned to identify violent behaviors.

Deep learning has shown its application prospects in computer vision tasks [14–17]. Many classic deep learning models such as VGGNET [18], YOLO [19], ResNet [20] and their variants have also achieved good performance in the field of violence recognition [21–24]. With the development of deep learning, the accuracy on the major public datasets has been constantly refreshed; meanwhile, the volume of the model has become increasingly larger. In particular, after vision transformer (ViT) [25] broke the application barrier in the field of computer vision, a series of research [26–28] based on transformer have showed its potential on many difficult tasks. With numerous parameters and calculations, advanced algorithms can only run in top-level hardware environments. This promoted the development of lightweight research of deep learning to obtain smaller and faster models while ensuring accuracy. The emergence of GhostNet [29], ShuffleNet [30] and MobileNet [31–33] enables edge devices to directly run deep learning models.

As a specific task of behavior recognition, violence recognition belongs to pattern classification in the field of computer vision. Its main task is to identify whether the picture contains violent behavior. Different from the task of behavior recognition, violence recognition must have a more clear and detailed division of human behavior. In recent years, scholars have been committed to the research of violence recognition from two perspectives.

First, a variety of algorithms are fused to meet the needs of violence recognition. Spatio-temporal attention modules and frame-grouping method [34] were proposed to build a violence detection system. By averaging the channels and grouping three consecutive channel-averaged images, short-term dynamics can be well simulated which is very helpful for violence identification. A deep multi-net (DMN) [35] architecture based on AlexNet and GoogleNet was proposed for violence recognition in videos, which used transfer learning to solve the problem of abrupt camera motion. By integrating Fusion Convolutional Neural Network (Fusion-CNN), spatio-temporal attention modules and Bi-directional Convolutional LSTMs (BiConvLSTM) [36], a two-stream architecture was proposed for violence recognition and localization in videos.

Second, the algorithm performance is improved by replacing a part of the basic framework. By changing the 3D convolutional layer to a depth-separable convolutional layer [37], this mix convolution block greatly reduces the computational cost. Ref [24] introduces a semi-supervised approach into pre-trained I3D, which can improve accuracy by removing redundant data and focusing on useful visual information. Ref [38] uses CNN-LSTM to extract visual and
auditory information simultaneously, then, a shared semantic subspace is constructed based on an autoencoder mapping model, which can fuse segment level features.

However, most of the existing studies focus on improving the accuracy of the algorithm, while ignoring the complexity of the algorithm. To meet the requirements of practical application, violence recognition must satisfy high accuracy requirements and speed of real-time recognition. Therefore, to solve the aforementioned problems, a lightweight network model MobileNet-TSM, which was proposed. Our contributions are summarized as follows:

1. We propose a lightweight network model MobileNet-TSM to meet the dual requirements of accuracy and real-time for violence recognition.
2. We integrate the idea of temporal shift module, such that the original model can better extract the characteristic information of violent behavior, and further improve the accuracy of violence recognition.
3. We conducted several experiments on three commonly used violence recognition datasets: Hockey Fights, Crowd Violence and RWF-2000. We compared the advantages and disadvantages of the proposed method with those of the most advanced method in terms of accuracy, model parameters, and model size.
4. We deploy the trained model on the Android terminal and design a mobile app to show the application prospect of the method proposed in this study.

**Related work**

**Depthwise separable convolutions**

Depthwise separable convolutions, mainly proposed by the Google research team, are the soul of Inception and MobileNet series. They solve the main problem of maintaining the accuracy of the network’s classification, while completely reducing the computational cost and memory usage of the model. The lightweight network with smaller size and faster speed enables mobile intelligent terminals and embedded devices to run the neural network model. The main idea of depthwise separable convolutions is to use multiple small-sized convolution cores instead of large-sized cores to simplify the network complexity and reduce the number of parameters.

The main method used in this study refers to the depthwise separable convolution of MobileNet series, and uses the combination of depthwise and pointwise revolution to build the basic module of the network. Fig 1 presents the schematic of depthwise and pointwise revolution.

A depthwise convolution sets a convolution kernel for each input channel, and each convolution kernel performs a convolution operation independently to an input channel, thus greatly reducing the amount of calculation and parameters of the model. Then, pointwise revolution integrates the characteristic information extracted by each convolution core. The

![Fig 1. Schematic of depthwise revolution and pointwise revolution.](https://doi.org/10.1371/journal.pone.0276939.g001)
The number of convolution cores is set equal to the number of output channels, and the size of convolution cores is \(1 \times 1 \times N\), where \(N\) represents the number of input channels. The combined depthwise separable convolution replaces the traditional 3D convolution, thus reducing the amount of calculation by approximately 8 to 9 times.

**Temporal shift module**

Temporal shift module (TSM) was proposed by Lin [39] in 2019, which is specially designed to exchange some position information between adjacent frames by shifting some channels in the time dimension before the convolution layer extracts the characteristic information. After that, 2D convolution kernel can extract spatial and temporal information at the same time, which can achieve the performance of 3D convolution neural network with the complexity of 2D convolution neural network. As shown in Fig 2(a), the original input tensor is stacked by several adjacent frames. We use different color to represent the input video frames at different times. As shown in Fig 2(b), by moving some channels along the temporal dimension in the same input batch, the information between frames is exchanged to a certain extent. When the following 2D convolution neural network is used to extract features, not only the spatial information of the current frame can be extracted, but also part of the temporal information of adjacent frames can be extracted.

For better expression, we assume that the input is an infinite one-dimensional vector \(X^0\) and a \(1 \times 3\) convolution kernel \(W_1 = (a, b, c)\), then the convolution operation \(Y = \text{Conv}(W, X^0)\) can be written as:

\[
Y_i = a \ast x_{i-1} + b \ast x_i + c \ast x_{i+1}
\]

(1)

\(Y_i\) represents the \(i\)-th element in \(Y\). \(x_{i-1}\), \(x_i\), and \(x_{i+1}\) represent the \((i-1)\)-th, \(i\)-th and \((i+1)\)-th elements in the input vector \(X\), respectively. \(X^{-1}\), \(X^{+1}\) represent the infinite one-dimensional vector \(X^0\) shifted back and forth by a unit. The sum of \(x_{i-1}\) is \(X^{-1}\). The sum of \(x_i\) is \(X^0\). The sum of \(x_{i+1}\) is \(X^{+1}\). The shift operation can be written as:

\[
X_{i-1} = X_{i-1}, X^0 = X_i, X_{i+1} = X_{i+1}
\]

(2)

\[
X^{-1} = \sum x_{i-1}, X^0 = \sum x_i, X^{+1} = \sum x_{i+1}
\]

(3)

Fig 2. (a) The original tensor without shift, (b) The tensor after shift.

https://doi.org/10.1371/journal.pone.0276939.g002
After that the convolution block followed will complete the multiply-accumulate operation as follows. Therefore, after passing through temporal shift module, each frame contains some information of adjacent frames.

\[ Y = \sum Y_i = a \sum x_{i-1} + b \sum x_i + c \sum x_{i+1} \]  

(4)

\[ Y = a \sum x^{-1} + b \sum x^0 + c \sum x^1 \]  

(5)

**Partial temporal shift module**

In our previous work, we proposed Partial temporal shift module [40] based on TSM [39]. We tried to improve the TSM algorithm from the perspectives of accuracy and real-time performance to meet the realistic requirements of violence recognition. By reducing temporal shift modules’ insertion, our proposed optimal model had not only reduced the memory usage of hardware, but also achieved higher accuracy on multiple datasets with 77.3% running time. However, the P-TSM algorithm is still too complicated, so that it is still difficult to deploy applications in mobile intelligent terminals on a large scale.

**MobileNet-TSM**

**Explanation**

In order to deploy violence recognition applications on mobile intelligent terminals on a large scale, violence recognition significantly increases the demand for real-time and accuracy of the algorithm, especially the ability of fast recognition. Therefore, we turn our attention to the MobileNet series [31–33], which is a series of lightweight convolutional neural networks specially designed for mobile terminals or embedded devices. We carried out relevant experiments which inserted temporal shift module into MobileNet-V2 [32] and MobileNet-V3 [33], tried to further improve the performance of the model.

In this paragraph, we share two failed experiments to explain why we choose MobileNet-V2 as backbone structure. The first is to use MobileNet-V3 as the backbone structure, which exhibits the same accuracy as that of MobileNet-V2 on two smaller datasets, Hockey Fights and Crowd Violence. On RWF-2000, the highest accuracy we trained is 83.75% which is lower than that of MobileNet-V2. At the same time, MobileNet-V3 does not run as fast as MobileNet-V2 due to its higher model complexity. This runs counter to our initial goal of improvement. Another attempt is to further integrate the attention mechanism. We notice that the squeeze-and-excitation (SE) [41] attention module is integrated in MobileNet-V3 [33], but after inserting temporal shift modules, the model is always unable to fit. The same phenomenon also appears in the experiments combining the convolutional block attention module (CBAM) [42] in MobileNet-V2 [32]. We conjecture that the exchange of information between frames by the temporal shift module confuses the attention mechanism’s feature extraction capability for key point regions. So we selects MobileNet-V2 [32] as the backbone after comparing the parameters and performance of the experiments.

**Main network structure**

The proposed lightweight mobile network that can realize real-time violence recognition is mainly realized by inserting temporal shift modules in the backbone. The structure of the model is presented in Table 1. Conv2d represents two-dimensional convolution. Bottleneck is the main component of the model. Avgpool represents average pooling. Linear represents the full connection layer. t represents the expansion factor. c represents the number of output
channels. n represents that the corresponding bottleneck will repeat n times. s represents that the first layer of each part has a stride s and all others use stride 1. m represents the number of temporal shift modules we insert in the corresponding bottleneck.

In Fig 3, blocks with different color are used to represent different functional layers of the network. Bottleneck, which forms the main body of the network, mainly comprises several temporal shift modules inserted in the basis of MobileNet-V2 [32], as shown in Fig 4. The basic structure first uses pointwise revolution to upgrade the dimension of the input picture, with the expansion ratio of 6, and then uses temporal shift modules to exchange information among the input tensors. Batch normalization and ReLU6 are simultaneously used for normalization and activation. Then, depthwise revolution is used to extract the characteristic information of violent behaviors, and the combination of batch normalization and ReLU6 is also used. The projection layer, which is essentially a pointwise revolution, is used to compress data and tighten the network interface. In most of these bottleneck, shortcut connections are used to improve the performance of the model. Two additional issues must be addressed.

Details

In this section, we explain two details of the network structure.

Why do we need to upgrade the dimension? If we downgrade the dimension of the input tensor, the calculation of the convolution layer will be reduced. The dilemma is that the higher the dimension of the input tensor, the more characteristic information can be extracted from...
the convolution layer, and sufficient characteristic information can generate stable prediction results. Because the depth revolution can greatly reduce the amount of calculation, we can first use the pointwise revolution to upgrade the dimension, and then after the extraction the projection layer is used to downgrade the dimension. Both the pointwise revolution and projection layer have parameters to learn. Therefore, the entire network structure can learn how to better expand and compress the data.

Why do we insert the temporal shift modules in this way? The first reason is that we have considered the location of the temporal shift module. We insert the temporal shift module after the pointwise revolution instead of inserting it between each bottleneck similar to the original TSM algorithm. If we first perform temporal shift and then upgrade the dimension, the same information is extracted by six groups of convolution kernels. The information between adjacent frames exchanged by the temporal shift module will be diluted. The second reason is the performance of the model. Although one research [43] shows that appropriately increasing the number of temporal shift modules can improve the performance of the network, our experiments prove that increasing the number of temporal shift modules is not beneficial, particularly for models with a small capacity. Therefore, to maintain the stability of the model, the method we proposed only inserts the temporal shift module into the bottleneck that meets the following conditions: first, the stripe of the depthwise convolution is equal to 1. Second, the shape of the input characteristic matrix is the same as that of the output characteristic matrix. Third, shortcut connection is used.

Experiments

Datasets

In order to verify the performance of the proposed algorithm, we carried out experiments on three public violence recognition datasets. A summary of the used datasets is shown in Table 2 below.

The Hockey Fights [44] dataset contains 1000 violent and non-violent videos collected from ice hockey game. The training set includes 800 video clips, the verification set includes 100 video clips and the test set includes 100 video clips. The Crowd Violence [4] dataset mainly contains the scenes of crowd, but due to long shooting distance and low resolution, most of the scenes are chaotic and vague. The latest published RWF-2000 [45] dataset contains 2000 surveillance video clips collected from Youtube. The training set includes 1600 video clips, the
verification set includes 200 video clips and the test set includes 200 video clips. Each video clip is 5 seconds and contains 150 frames. It mainly includes violent behaviors such as two persons, multiple persons, and crowds. The scenes are so rich and complicated that it is difficult to recognize. All video clips are obtained through the security camera. Without multimedia technology transformation, they fit the actual scene and have high research value.

**Experimental environment and configuration**

We rented a virtual computer on AI-Galaxy GPU cloud for experiments and using Windows 10 as the operating system. During the entire experiments, we used Pytorch 1.9.0 as the deep learning framework and Intel(R) Xeon(R) Gold 6130 @2.10GHz as the CPU. We use CUDA 11.1 to accelerate the GPU and two NVIDIA RTX 2080 Ti GPUs with 11 GB video memory for parallel computing.

During the experiment, we used the pre-trained model of MobileNet-V2 provided by PyTorch to reduce the computational complexity of network training. We set the initial learning rate as 0.000375 and used Adam optimizer. The momentum decay rate used is 0.9. The input modality of the picture is RGB. The batch size of the input picture is 8. The shift division of the temporal shift module is also set to 8. During training, when the curve of the model on the test set significantly fluctuated, we set the initial learning rate as 0.0005 or 0.0001 and change the learning rate adjustment method. We adjusted the learning rate to 85% of the original learning rate every two epochs to train a better model.

**Ablation experiment**

**Accuracy of TSM module insertion experiments.** First, we first tested the accuracy of the baseline model, namely MobileNet-V2 [32], on three public datasets of violent behaviors, and then, we tested the model by inserting temporal shift modules as original TSM [39] (Baseline + TSM #x2212; 1). We inserted and conducted experiments as we designed (Baseline + TSM − 2). The experiments’ accuracy is presented in Table 3.

The baseline model using method 2 to insert the temporal shift module achieved 97.959% accuracy on the Crowd Violence, 97.5% on the Hockey Fights, and 87.75% on the RWF-2000. Compared with the baseline model, the accuracy increased by 5.102%, 2.5%, and 4.75% respectively. The baseline model using method 1 to insert the temporal shift module also had a certain improvement in accuracy compared with that of the baseline model, but the performance was not as good as that of method 2. The experiments showed that the addition of the temporal

| Dataset            | Year of release | Clips include | FPS | Frame resolution |
|--------------------|-----------------|---------------|-----|------------------|
| Crowd Violence [4] | 2012            | 246           | 25  | 320 x 240        |
| Hockey Fights [44] | 2011            | 1000          | 25  | 360 x 288        |
| RWF-2000 [45]     | 2021            | 2000          | 30  | 300 x 240, 320 x 240, 480 x 360, 920 x 720, 1280 x 720 |

Table 3. Introduction of used datasets for violence recognition.

|                 | Crowd Violence | Hockey Fights | RWF-2000 |
|-----------------|---------------|--------------|----------|
| Baseline        | 92.857%       | 95%          | 83%      |
| Baseline+TSM-1  | 95.95%        | 97.2%        | 87%      |
| Baseline+TSM-2  | 97.959%       | 97.5%        | 87.75%   |

https://doi.org/10.1371/journal.pone.0276939.t003
shift module could effectively improve the baseline’s capability of extracting spatio-temporal information. Through cross-channel information interaction, the bottleneck of the model was made more sensitive to the temporal information of violent behaviors, thereby improving the recognition accuracy.

**Comparison of optimal methods.** To further compare the performance of our model with that of the state-of-the-art behavior recognition algorithm, we tested it on different datasets in the same experimental environment. Table 4 presents the specific performance of different algorithms.

As shown in the table, the proposed model, which inserts the temporal shift module, achieves similar or even better results than those of the other state-of-the-art behavior recognition algorithms on three public violence recognition datasets. The proposed method achieves 97.959% accuracy on the Crowd Violence which is higher than most of the algorithms. Only 0.191% lower than SAM, which achieved the best performance on the Crowd Violence. On Hockey Fights, our methods achieved 97.5% accuracy that was slightly lower than that of the two-cascade TSM [48]. There is a gap of 2% compared with SAM that obtains the optimal accuracy. On RWF-2000, the method only achieved 87.75% accuracy which is 3.25% lower than P-TSM. In terms of accuracy, our proposed method is still inferior to the state-of-the-art algorithms.

MiNet-3D [37] also proposed a lightweight model for violence recognition, which uses a combination of 3D convolution and 2D convolution as the basic framework. The model was lightened by using depthwise separable convolutions instead of 3D convolutions. The method we proposed uses MobileNet-V2 as the basic framework and integrates temporal shift module to improve the performance of the model. MobileNet-V2 uses deep separable convolution as its main structure.

As shown in Fig 5, we can see that the accuracy does not significantly improve, but our model has less parameters and lower computational complexity. The biggest advantage of our proposed method lies in the complexity of the model, which is discussed in detail in the next paragraph.

**Comparison of the computational cost.** To better demonstrate the advantages of the proposed method, we tested more related data to further compare the performance of the algorithms. We compared the computational cost of the proposed method with that of the existing algorithms, as presented in Table 5. Fig 6 shows the comparison of estimated total size.

| Algorithm          | Crowd Violence | Hockey Fights | RWF-2000 |
|--------------------|----------------|---------------|----------|
| ResNet-50 [20]     | 93.878%        | 95.5%         | 84%      |
| 3D-CNN [46]        | 94.3%          | 94.4%         | 82.75%   |
| LRCN [47]          | 94.57%         | 97.1%         | 77%      |
| I3D [48]           | 88.89%         | 97.5%         | 85.75%   |
| MiNet-3D [37]      | 91.41%         | 94.71%        | 81.98%   |
| AR-Net [49]        | 95.918%        | 97.2%         | 87.3%    |
| TSM [39]           | 95.95%         | 97.5%         | 88%      |
| TEA [50]           | 96.939%        | 97.7%         | 88.5%    |
| Two-cascade TSM [43]| 96.939%        | 98.05%        | 89%      |
| SAM [36]           | 98.15%         | 99.1%         | 89.1%    |
| SSHA [24]          | -              | 98.7%         | 90.4%    |
| P-TSM [40]         | 96.939%        | 98.5%         | 91%      |
| Baseline+TSM-2(ours) | 97.959%       | 97.5%         | 87.75%   |

https://doi.org/10.1371/journal.pone.0276939.t004
We used torchsummary to calculate models’ parameters and estimated total size. Since torchsummary can control all variables of model computational complexity, and the calculation results can be simply reviewed instead of manual calculation. For better comparison, we set the input of the summary function is set to a fixed value. We set the picture size as $224 \times 224$, batch size as 8, and number of input channels as 3. We used Tensorboard to record relevant data that provides the time used after training and testing for 100 epochs.

As observed from the Table 5, compared with the existing behavior recognition algorithms, including the latest violence recognition algorithms, the proposed method significantly reduced the complexity of the model. The use of depthwise separable convolution considerably reduced the number of parameters to be trained. Our proposed model has only 8.49MB parameters and 175.86MB estimated total size. Compared with the state-of-the-art violence recognition algorithm, our proposed model reduces the used parameters by an order of magnitude. Furthermore, the training time was substantially reduced, and only takes 32 minutes and 41 seconds, which is a quarter of the time two-cascade TSM [43] taken to complete one.

Table 5. Complexity of different models.

| Algorithm                | Params       | Estimated total size | Time cost  |
|--------------------------|--------------|----------------------|------------|
| 3D-CNN [46]              | 297.56MB     | 2647.70MB            | 4h8m23s    |
| LRCN [47]                | 237.83MB     | 1212.93MB            | 3h5m7s     |
| I3D [48]                 | 46.88MB      | 1000.20MB            | 2h3m48s    |
| TSN [51]                 | 89.69MB      | 390.05MB             | 46m41s     |
| TSM [39]                 | 89.69MB      | 397.71MB             | 1h53m26s   |
| TEA [50]                 | 91.95MB      | 479.78MB             | 3h10m48s   |
| Two-cascade TSM [43]     | 89.69MB      | 397.71MB             | 2h8m20s    |
| P-TSM [40]               | 89.69MB      | 396.57MB             | 1h27m38s   |
| Baseline+TSM-2(ours)     | 8.49MB       | 175.86MB             | 32m41s     |

https://doi.org/10.1371/journal.pone.0276939.t005
hundred epochs of training and testing. Fig 6 shows that the estimated total size of the model also greatly reduced, providing favorable conditions for the deployment in mobile intelligent terminals.

**Android smartphone implementation.** We used the Xiaomi MIX2S smartphone, Android system version 10, and Qualcomm Snapdragon 845 processor. It is worth mentioning that the android building Gradle used is 3.5.0, and the versions of “pytorch-android-lite” and “pytorch-android-torchvision” had to be the same as the PyTorch version when the deep learning model is training. Due to compatibility issues, we can only use PyTorch version 1.9.0 to do our work.

To deploy our trained model on the Android terminal, we first have to convert the format of the model. Built-in functions provided by Pytorch 1.9.0, optimize-for-mobile and save-for-lite-interpreter, were used to convert the format from .ph to .ptl. On the Android terminal, we can directly call the converted model.

The Android app mainly implemented two simple functions to demonstrate the practical performance of the proposed method. Function 1: Identify the existence of a violent behavior in the picture. After clicking the function button, the picture saved in the machine was read, converted into a tensor, and sent into the pre-trained deep learning model for prediction. Subsequently, the prediction result was output. Function 2: Identify the real-time image captured by the smartphone. After clicking the real-time detection function button, the app called the camera of the terminal to capture the real-time image, converted it into a tensor, and then input it into the pre-trained model for real-time detection and output prediction. Simultaneously, the current delay was showed.

**Conclusion**

To solve the problem of huge computational cost of the existing violence recognition algorithm, in particular, not meeting the needs of real-time mobile terminal recognition, this
paper proposes a lightweight network called MobileNet-TSM. The experimental results on three public datasets, that is, Hockey Fights, Crowd Violence, and RWF-2000 showed that the proposed method significantly reduced the model complexity compared with the existing violence recognition algorithms. It could achieve a performance similar to that of the state-of-art algorithm with a relatively less number of training parameters and transplant it to smart mobile devices, verifying the feasibility of its application and showing a broad application prospect. In the future research of real-time violence recognition on mobile intelligent terminal, compatibility is a problem that needs to be solved including the deployment of deep learning models and mobile intelligent terminals with different operating systems.

**Author Contributions**

**Investigation:** Yong Li.

**Methodology:** Youshan Zhang.

**Resources:** Yong Li.

**Software:** Youshan Zhang.

**Writing – original draft:** Youshan Zhang.

**Writing – review & editing:** Shaozhe Guo.

**References**

1. Chen LH, Su CW, Hsu HW. Violent scene detection in movies. International Journal of Pattern Recognition and Artificial Intelligence. 2011; 25(08):1161–1172. https://doi.org/10.1142/S0218001411009056

2. Giannakopoulos T, Pikakis A, Theodoridis S. A multimodal approach to violence detection in video sharing sites. In: 2010 20th International Conference on Pattern Recognition. IEEE; 2010. p. 3244–3247.

3. Kim HD, Ahn SS, Kim KH, Choi JS. Single-channel particular voice activity detection for monitoring the violence situations. In: 2013 IEEE RO-MAN. IEEE; 2013. p. 412–417.

4. Hassner T, Itcher Y, Kliper-Gross O. Violent flows: Real-time detection of violent crowd behavior. In: 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops. IEEE; 2012. p. 1–6.

5. Vashistha P, Bhatnagar C, Khan MA. An architecture to identify violence in video surveillance system using ViF and LBP. In: 2018 4th international conference on recent advances in information technology (RAIT). IEEE; 2018. p. 1–6.

6. Gao Y, Liu H, Sun X, Wang C, Liu Y. Violence detection using oriented violent flows. Image and vision computing. 2016; 48:37–41. https://doi.org/10.1016/j.imavis.2016.01.006

7. Lowe DG. Distinctive image features from scale-invariant keypoints. International journal of computer vision. 2004; 60(2):91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

8. Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). vol. 1. Ieee; 2005. p. 886–893.

9. Das S, Sarker A, Mahmud T. Violence detection from videos using hog features. In: 2019 4th International Conference on Electrical Information and Communication Technology (EICT). IEEE; 2019. p. 1–5.

10. Bilinski P, Bremond F. Human violence recognition and detection in surveillance videos. In: 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS). IEEE; 2016. p. 30–36.

11. Zhou P, Ding Q, Luo H, Hou X. Violence detection in surveillance video using low-level features. PLoS one. 2018; 13(10):e0203668. https://doi.org/10.1371/journal.pone.0203668 PMID: 30281588

12. Mahmoodi J, Salajeghe A. A classification method based on optical flow for violence detection. Expert systems with applications. 2019; 127:121–127. https://doi.org/10.1016/j.eswa.2019.02.032

13. Senst T, Eiselein V, Sikora T. A local feature based on Lagrangian measures for violent video classification. In: 6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15). IET; 2015. p. 1–6.
14. Li W, Li S, Yin L, Guo X, Yang X. A novel visual classification framework on panoramic attention mechanism network. IET Computer Vision. 2022.;
15. Peng L, Jidong Z, Zhen Z, Wei W, Yamei D. Submission for IET research journal papers person re-identification based on feature erasure and diverse feature learning. IET Computer Vision. 2022.;
16. Yang F, Li W, Liang B, Han S, Zhu X. Multi-stage attention network for video-based person re-identification. IET Computer Vision. 2022.;
17. Ye L, Li W, Zheng L, Zeng Y. Lightweight and Deep Appearance Embedding for Multiple Object Tracking. IET Computer Vision. 2022.;
18. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014.;
19. Redmon J, Farhadi A. YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 7263–7271.
20. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.
21. Shripriya C, Akshaya J, Sowmya R, Poonkodi M. Violence Detection System Using Resnet. In: 2021 5th International Conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE; 2021. p. 1069–1072.
22. Mohammad H, Nazerfard E. Video Violence Recognition and Localization using a Semi-Supervised Hard-Attention Model. arXiv preprint arXiv:220202212. 2022.;
23. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: More features from cheap operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 1580–1589.
24. Kang MS, Park RH, Park HM. Efficient spatio-temporal modeling methods for real-time violence recognition. IEEE Access. 2021; 9:76270–76285. https://doi.org/10.1109/ACCESS.2021.3083273
25. Mumtaz A, Bux Sargano A, Habib Z. Fast learning through deep multi-net CNN model for violence recognition in video surveillance. The Computer Journal. 2022; 65(3):457–472. https://doi.org/10.1093/comjnl/bxaa061
26. Asad M, Jiang H, Yang J, Tu E, Malik AA. Multi-Level Two-Stream Fusion-Based Spatio-Temporal Attention Model for Violence Detection and Localization. International Journal of Pattern Recognition and Artificial Intelligence. 2022; 36(01):2255002. https://doi.org/10.1142/S0218001422550023
27. Wang W, Dong S, Zou K, Li W. A Lightweight Network for Violence Detection. In: 2022 the 5th International Conference on Image and Graphics Processing (ICIGP); 2022. p. 15–21.
39. Lin J, Gan C, Han S. Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 7083–7093.

40. Zhang Y, Li Y, Guo S, Liang Q. Not all temporal shift modules are profitable. Journal of Electronic Imaging. 2022; 31(4):043030. https://doi.org/10.1117/1.JEI.31.4.043030

41. Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 7132–7141.

42. Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 3–19.

43. Liang Q, Li Y, Chen B, Kaikai Y. Violence behavior recognition of two-cascade temporal shift module with attention mechanism. Journal of Electronic Imaging. 2021; 30(4):043009. https://doi.org/10.1117/1.JEI.30.4.043009

44. Bermejo Nievas E, Deniz Suarez O, Bueno García G, Sukthankar R. Violence detection in video using computer vision techniques. In: International conference on Computer analysis of images and patterns. Springer; 2011. p. 332–339.

45. Cheng M, Cai K, Li M. RWF-2000: an open large scale video database for violence detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). IEEE; 2021. p. 4183–4190.

46. Ji S, Xu W, Yang M, Yu K. 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence. 2012; 35(1):221–231. https://doi.org/10.1109/TPAMI.2012.59

47. Donahue J, Anne Hendricks L, Guadarrama S, Rohrbach M, Venugopalan S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 2625–2634.

48. Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 6299–6308.

49. Meng Y, Lin CC, Panda R, Sattigeri P, Karlinsky L, Oliva A, et al. Ar-net: Adaptive frame resolution for efficient action recognition. In: European Conference on Computer Vision. Springer; 2020. p. 86–104.

50. Li Y, Ji B, Shi X, Zhang J, Kang B, Wang L. Tea: Temporal excitation and aggregation for action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 909–918.

51. Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, et al. Temporal segment networks for action recognition in videos. IEEE transactions on pattern analysis and machine intelligence. 2018; 41(11):2740–2755. https://doi.org/10.1109/TPAMI.2018.2868668 PMID: 30183621