Deep Learning-based Abnormal Activity Recognition in Video

Jeongtae Kim
Hanyang University
Seoul, Korea
jt0227@hanyang.ac.kr

Yongsu Lee
Hanyang University
Seoul, Korea
lysu3612@hanyang.ac.kr

Abstract

In this paper, we propose a deep learning-based framework for abnormal scene recognition in video, leveraging both optical flow and temporal feature learning to detect irregular events effectively. For abnormal scene detection, we introduce a fusion model combining 3D convolutional networks and optical flow, enabling the model to learn spatial-temporal patterns which are important for identifying abnormal activities. Our experiments utilize the mini-RWF2000 dataset [1], pre-labeled with "Fight" and "Non-Fight" scenarios, to validate our approach. An ablation study further demonstrates the significance of optical flow integration for improved detection accuracy.

1. Introduction

In this section, we introduce the necessity of our model explaining the background of our society.

1.1. Background

Nowadays, there are many CCTVs and recording cameras in the world to detect crime or abnormal scenes. However, because of too many CCTVS and less workers to watch them, it is necessary to make something that can detect abnormal scenes instead of humans or let humans know when there are crimes. As we acknowledge this situation, we aim to make a network that can detect abnormal cases.

2. Model Architecture

We did two different tasks which are optical flow and abnormal detection. For optical flow, instead of using opencv library, we implement this algorithm with some given procedure. This work can be done only with numpy and CPU. The other task is abnormal detection uses torch and GPU.

2.1. Lucas Kanade

When figuring out the motion of objects in images, we need to specify accurate features and motion vectors. First, to find significant corners as features used Harris Corner Detector instead of goodFeaturesToTrack function in OpenCV Library. And then we applied Lukas-Kanade algorithm to identify the motion of features.

2.2. Model

When detecting abnormal scenes, movement is important because usually when detecting abnormal cases, there are some abnormal movements. Therefore, we added two different time sequential data which is Farneback and 3D CNN to let the network learn more data related to time.

2.2.1. Farneback algorithm

The Farneback algorithm estimates optical flow between two frames using polynomial expansion. Each

pixel's neighborhood is approximated by a quadratic polynomial and then computes displacement by analyzing the changes in polynomial coefficients. Noise is reduced through local averaging and large displacements are handled using a multi-scale approach. Also, accuracy of displacement estimation through iterative refinement. Therefore, this method is robust for motion estimation, particularly for varying displacement fields and large-scale motion.

In terms of the simplicity and efficiency, Farneback algorithm has similar assumptions to Lukas-Kanade algorithm. Both algorithms assume 'spatial coherence' that the displacement field varies slowly [3]. In addition to this explicitly stated assumption, we can conjecture that Farneback algorithm also adopted the assumption that the pixel brightness is constant as time goes on (Brightness Constancy) in that it uses a polynomial approximation like as LK algorithm.

The Farneback algorithm is different in that it takes a multi-scale displacement estimation. This approach starts at a coarse scale to obtain a rough displacement estimation and then progresses to a finer scale to get more accurate estimates. Also, as a dense optical flow, Farneback algorithm computes the optical flow vector for every pixel of the frame while as a sparse optical flow, LK algorithm does same things only for distinguishing features like corners. Second-order polynomial approximation is also an important component for Farneback algorithm. For these reasons, Farneback algorithm's computational cost is higher than the LK algorithm, but more sophisticated results are provided.

2.2.2. 3D CNN

For Fusion model, we use 3d convolution and max pooling. In input, there is image size, video time, optical flow & RGB. We divide optical flow and RGB channel and proceed them with 3d convolution. In 3d convolution, we do two kinds of convolution. One is convolution with image size and the other is convolution with time. Activation function, ReLU, is followed by each convolution, but in the last layer of optical flow, sigmoid function is used because of detecting movement in that pixel. After convolving each channel, there are 2 outputs which are RGB and optical flow. Multiplying two outputs made fused feature and we did similar convolution to this fused feature. For classification, we proceed multilayer perceptron which has two dimensions, that match our task for detecting normal and abnormal scenes.

3. Training

This section shows what type of training algorithm and hardware we used.

3.1. Training Data

For dataset, we used mini-RWF2000 [1] which pre labeled with Fight and Non-Fight. For trainset, there are 160 train videos and 40 validation videos with 5 seconds length. In sight of dimension, each video should have the same length. To match the dimensions, we apply padding. Two types of data augmentation are applied to each frame. One is color jittered and the other is flipped. Figs 1,2,3. shows how images change.

Figure 1. Original

Figure 2. Color jittered

Figure 3. Color jittered & flipped

3.2. Hardware

We trained our models by A100 in Google Colab.

3.3. Optimizer and Loss function

For optimizer function, we use SGD with learning rate 0.003 and weight decay 1e-6. In addition, we also used cosine annealing learning rate scheduler for dynamic learning rate. For loss function, we use Cross Entropy Loss.

Figure 4: Result of optical flow of Lukas-Kanade algorithm

4. Results

We have done two different tasks which are optical flow and abnormal detection. Optical flow can be evaluated with qualitative evaluation and abnormal detection can be analyzed with quantitative analysis.

4.1. Qualitative evaluation

In Fig. 4, we can see the frame of how optical flow acts to vehicles in highway by Lukas-Kanade algorithm. It seems algorithm detects the feature's motion or the object's motion well although some sparse motion vectors of disappeared feature are detected. We improved optical flow by optimizing hyperparameters such as quality level, window size, threshold, and so on.

4.2. Quantitative analysis

Figure 5. Train loss of base model

Figure 6. Train accuracy of base model

Figure 7. Validation loss of base model

Figure 8. Validation accuracy of base model

By using base model [1], the outputs are shown in Fig. 5,6,7, and 8. The train loss converges, and train accuracy achieves about 0.75 and validation accuracy achieves about 0.75 equally.

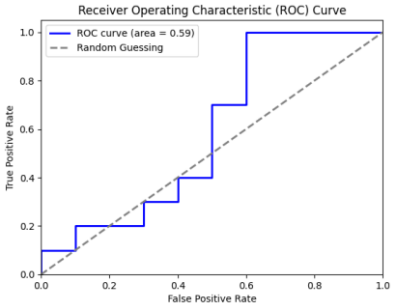

Figure 9. ROC of test dataset by base model

5. Discussion

5.1. Ablation Study

We propose an ablation study to see the result of adding optical flow. Instead of multiplying the camera channel (RGB channel) and optical flow, we only use camera channel, and the following result is in Fig. 10 and 11. The base model without optical flow achieves higher train accuracy than base model but base model gains higher validation accuracy than base model without optical flow, which means base model has more generalization ability.

In optical flow, the sign is determined based on the direction but in processing of optical flow, there is normalization to 0 and 255 which makes negative direction to zero and no movement to medium value. However, we only consider the intensity of optical flow not the direction so before normalization, we apply the absolute value to all components to maintain the intensity of movement. We can get higher accuracy in train dataset and validation accuracy is similar as seen in Fig. 12 and 13.

Figure 10. Train accuracy of base and without optical

Figure 11. Validation accuracy of base and without optical

Figure 12. Train accuracy of base and absolute optical

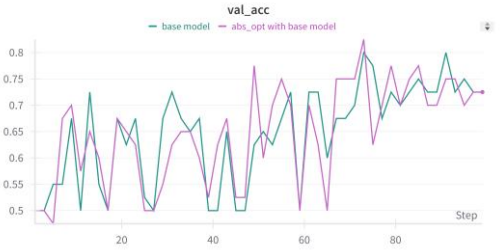

Figure 13. Train accuracy of base and without optical

5.2. Customized Model

The reason why we add farneback algorithm to this model is because we want to concentrate pixel of movement. Instead of adding inductive bias of optical flow using sigmoid function as weight, we propose

cross attention between optical flow and camera channel. For the dimension, in Transformer [2], there is encoder-decoder attention which is cross attention. Query comes from decoder and key and value come from encoder and they calculate attention with embedding vector after making dimensions as batch size, sequence length, embedding dimension. To mimic this dimension, we made (B, T, C, H, W) to (B, T, C*H*W). B means batch size, T means time, C means channel, H means height, and W means width. In this way, we aim to let our model learn spatial relationships between optical flow and camera channel. In addition, we add classification token as learnable parameters. Next, we used multi head self-attention to attend all information in every sequence using time dimension as key, query, value. The result of using attention is shown in Fig. 14, 15. The accuracy of the train and validation is lower than that of base model. The reason for this result is because the dataset is quite small to train attention which uses much more parameters than just multiplying in base model.

Figure 14. Train accuracy with attention

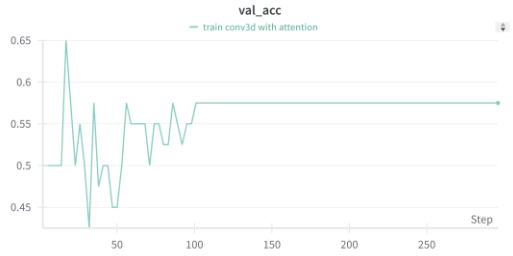

Figure 15. Validation accuracy with attention

5.3. Different dataset

We also did inference our model for new dataset which is Smart-city CCTV Violence Detection Dataset (SCVD) [5] to verify the background of our research. From Fig 15, we successfully detect the violence by CCTV.

Figure 15. Detection of violence by CCTV

References

[1] Brown, Tom, et al. Language Models are Few-Shot Learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:1877–1901, 2020.

[2] Vaswani, Ashish, et al. Attention is All You Need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30:5998–6008, 2017.FirstName Alpher, FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? *Journal of Foo*, 14(1):234–778, 2004.

[3] Horn, Berthold KP, and Brian G. Schunck. Two-Frame Motion Estimation Based on Polynomial Expansion. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[4] Kumar, Vinayak, et al. Towards Smart City Security: Violence and Weapon Detection in Surveillance. *Proceedings of the 2020 IEEE International Conference on Image Processing (ICIP)*, 3214–3218, 2020.