UNIVERSITY OF LIVERPOOL

Deep Learning in Video Anomaly Detection and Its Applications

Thesis submitted in accordance with the requirements of the University of Liverpool for the degree of Doctor in Philosophy by

Yuxuan Zhao

December 2021

Abstract

With the popularization of the city monitoring system, surveillance videos have been increasingly presented. Traditional methods for video analytic require professionals to monitor the video constantly to find out abnormal events, which leads to a tough and time-consuming task. Therefore, research activities on automatic video anomaly detection are of great practical significance since a feasible detection technique can reduce the large amount of human resources used for monitoring videos. This thesis presents several novel deep learning methods for video anomaly detection. In addition, it provides a potential system for the application of these methods and extension of the video sources. Video anomaly detection is a problem of detecting and classifying anomalies in videos. Anomaly refers to an unusual event or emergency that deviates from what is standard, normal and expected. The kernel of video anomaly detection is the extraction of spatial and temporal features.

Proposed methods in this thesis are all based on the two-stream structure. This structure allows two streams with different inputs. The input is sampled frames for the first stream, while the second stream needs optical flow as its input. The traditional two-stream model extracts spatial features only from the first stream. In addition, all temporal streams are captured and handled from the stream of optical flow. Some significant improvements have been implemented in our models. For the spatial features, since convolutional neural networks (CNN) have been proved to have a good performance, we keep the convolutional structure and replace the basic CNN with advanced CNN (DenseNet). For the temporal features, proposed methods try to extract them from both two streams. In the first stream, 3D convolution (C3D) and Long Short-Term Memory (LSTM) are used to handle a sequence of frames. In the second stream, we implement the DenseNet Structure to improve the performance. These modifications make the whole model too complicated such that the training process would be complex. Therefore, the clip-based video processing method is designed to enhance the efficiency of the training process and reduce the pressure of computation.

Experiments are conducted to validate the performance of the proposed two-stream methods with comparisons along several well-known videos related to deep learning models.

UCF-101 is used to evaluate the general performance of models. FIRESENSE Dataset and UCF-Crime are used to test the performance of video anomaly detection tasks. We also collected a merged dataset to simulate the anomaly detection. Proposed models perform well on all these datasets.

Acknowledgements

First and foremost, I wish to express my deepest thanks and gratitude to my supervisors Prof. Ka Lok Man, Prof. Jeremy S. Smith, and Prof. Sheng-Uei Guan, who had provided unreserved and invaluable professional guidance for my research during the past four years. I appreciate their patience in our weekly/monthly meetings. Their suggestions and help have played a vital role in my research.

I also want to thank my Independent Progress Assessment Panel (IPAP) members, Dr. Dawei Liu, Dr Jieming Ma and Dr. Waleed Al-Nuaimy, for reviewing my annual reports every year. Their questions and feedback during for annual IPAP meetings helped me to solve many research problems.

Furthermore, I would like to acknowledge the financial support from XJTLU and UoL.

I would also like to thank all the staff members at the Xi'an Jiaotong-Liverpool University (XJTLU) and the University of Liverpool (UoL). Many thanks to my colleagues for their useful discussions, suggestions, and help in research and life. They are Dr. Yuechun Wang, Ziqiang bi, Dr. Fangyu Wu, Qi Chen, Jing Qian, Xianbin Hong, Dr. Hang Dong, Dr. Jie Zhang, Dr. David Afolabi, and Dr. Vijayakumar Nanjappan.

Finally, I would like to take this opportunity to express my great gratitude to my wife, Mrs Wei Zhao, for her support and encouragement during the last four years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, the background and motivation of the whole thesis are presented firstly. Then it introduces the aims and objectives, followed by the outline of the contributions. Finally, a general list of the thesis is given at the end of this chapter.

1.1 Background and Motivation

Video anomaly detection is the problem of detecting anomalies in videos. It focuses on finding whether the given video frames exhibit an anomaly or not. Anomalies refer to some unforeseeable events and emergencies, which are presented in Figure 1.1, that deviate from what is standard, normal, or expected. Anomaly detection plays an essential role in smart city management, such as traffic control and criminal investigation. Unlike other anomaly detection tasks that can provide clear unusual signals [12], video anomaly detection requires the analysis of videos.

Due to the requirements of urban security, surveillance videos have been increasingly presented in our cities in order to monitor human activity and prevent unusual events such as traffic accidents and criminal offences. Traditionally, we need professionals to watch the video constantly to find out abnormal events. It always turns into a tough and time-consuming task. Therefore, research activities on automatic video anomaly detection are of great practical significance since a feasible detection technique can reduce large amount

Figure 1.1: Some anomalies in our city

.

of human resources used for monitoring videos, especially for surveillance systems and improve the performance of the detection.

Traditional methods try to use trajectory-based anomaly detection. The principle of these methods is if the objects of interest are not following the learned normal trajectories, the video will be tagged as an anomaly. Such methods have high requirements for monitoring the objects of interest, which may not be suitable for surveillance videos. In addition, these traditional approaches are ineffective when they are used in a different domain. They cannot be adapted to new anomalies that they have not faced before. Since deep learning algorithms have generated a lot of success in computer vision field, recent works have focused on the use of deep learning methods for the task of anomaly detection. Figure 1.2 shows that with the increase of data size, the deep learning methods can achieve higher performance than traditional methods.

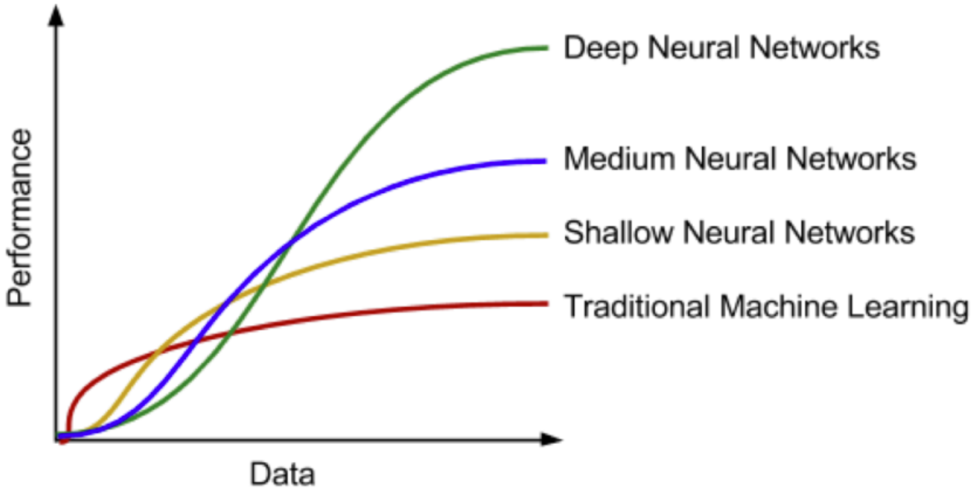


Figure 1.2: Comparisons of deep learning methods and traditional methods [1]

.

Since video anomaly detection is actually a task of video processing, there are two kinds of features that can be extracted by deep learning methods, the spatial features and temporal features. Some existing deep learning models, such as Convolutional Neural Networks (CNN) [13], have been proved that have ideal performance for spatial features. However, for the temporal features, several methods have been presented in the past 10 years, most of them have distinct advantages and disadvantages. For example, for the basic Recurrent Neural Network (RNN) [14], it achieves the target of extracting temporal information by adding a transmission mechanism for hidden layers. However, it still cannot handle long-term videos. Therefore, design of a model for video anomaly detection that can overcome the drawbacks of previous methods becoming a challenging and novelty research activity.

1.2 Aims and Objectives

The main aim of the thesis is the construction of a deep learning model for the video anomaly detection. In addition, the objectives of the thesis are given as follows:

- Spatial feature extraction model based on convolution structure.

  For either images or videos, spatial features are basic components that the video anomaly detection method needs to focus on. The traditional CNNs have been proved that have a good performance on handling such tasks [2]. However, considering the computation power and training time, the new model should be more suitable for video processing tasks.

- Deep Learning method for handling the temporal features.

  The most important difference between image and video is that video contains temporal information. It is also the reason why we focus on the video anomaly detection. The temporal information has a considerable impact on the accuracy of the detection. However, the current models, such as Long Short-term Memory (LSTM) 2.6 and Two-stream model [4], may not extract as many features as possible. So it is necessary to improve them for obtaining better performance.

- Video transmission and processing system for more potential video accesses.

  Although various surveillance cameras can provide video data in cities, there are still some blind spots that surveillance videos cannot cover, such as the indoor environment and remote regions. Since the monitoring coverage of surveillance cameras is limited, more methods to get videos into the system should be developed.

1.3 Contributions

The main focus of the thesis is to achieve better accuracy in video anomaly detection based on the two-steam structure. In addition, a video transmission system has been built. To address the research objectives in Section 1.2, this thesis contributes to the following:

- The LSTM is used in the spatial stream in the two-stream model to replace the original Convolutional Neural Network (CNN) in the traditional structure. LSTM can process the temporal information directly from frames. Therefore, some drawbacks of the optical flow can be solved such as the problem of moving cameras. Some experiments have been done to check which kind of LSTM can get the best feasibility for the video classification tasks in two-stream structure. This contribution also provides a basis for further optimization of the model.

- A combination of Convolution 3D (C3D) [5] and LSTM is used to replace the traditional spatial stream in two-stream model. The new stream improves the performance of feature extractions. Since both C3D and LSTM are complex, many improvements have been done to avoid the problem caused by the complexity of the model. The detailed improvements include:

  - Clip-based video processing method is used before C3D so that the small kernel in C3D can be used directly for long-term video.
  - The C3D structure is simplified with fewer convolution layers.
  - According to previous experimental results on UCF-101, single-directional LSTM is used.
  - The inputs of the LSTM are feature maps of C3D. Therefore, the sequence of inputs is reduced. The problems about gradient and feature missing can be solved.
  - For the fusion layer, average fusion is applied to replace the SVM.

- The Dense structure [3] is applied in the second stream in two-stream models for optical flow. The modified stream needs fewer feature images and parameters.

- The Enhanced Public Video Press (EPVP) is proposed as a system that allows authorized users to capture videos of unforeseeable events using mobile phones. In addition, it provides a web portal for related organizations or government departments to monitor unforeseeable events. For example, after an earthquake video is taken of the disaster, its aftermath could be uploaded to a central database.

The contributions mentioned above have led to a number of peer-reviewed publications, which are presented in various journal papers and conference papers:

SCI Papers:

- Zhao, Y., Man, K.L., Smith, J., Siddique, K. and Guan, S.U., 2020. Improved two-stream model for human action recognition. EURASIP Journal on Image and Video Processing, 2020(1), pp.1-9.
- Zhao, Y., Man, K.L., Smith, J. and Guan, S.U., 2021. A novel two-stream structure for video anomaly detection in smart city management. The Journal of Supercomputing, pp.1-15.

Journal Paper:

- Han, Z., Zhao, Y. and Man, K.L., 2019. Design and Implementation of the Enhanced Public Video Press for Unforeseeable Events Management. Journal of Industrial Information Technology and Application (JIITA), vol.3 no.4, pp.294-297.

Conference Papers:

- Zhao, Y., Gabriela, M. and Man, K.L., 2021. Video Anomaly Detection by the Combination of C3D and LSTM. in Proceedings of International Conference on Digital Contents: AICo (AI, IoT and Contents) Technology 2021.
- Zhao, Y., Zhang, J. and Man, K.L., 2020. LSTM-based Model for Unforeseeable Event Detection from Video Data. in Proceedings of The International Conference on Recent Advancements in Computing in AI, IoT and Computer Engineering Technology(CICET) 2020. pp.41.
- Zhao, Y. and Man, K.L., 2017. Autonomous Alert Generation and Recommendation for Disaster Management. in Proceedings of RESKO Techinical Conference 2017, pp.100-101.

1.4 Thesis Outlines

The thesis is organized as follows:

In Chapter 2, the related work is introduced. This chapter mainly discusses methods for video processing. Section 2.1 shows three models for the spatial features in videos. Section 2.2 shows four models that can handle the temporal features. Finally, in Section 2.3, some existing video anomaly detection methods have been introduced.

Chapter 3 proposes a two-stream model used for video classification. The main feature of this model is the usage of LSTM in the spatial stream. The traditional two-stream model only extracts temporal features from optical flows. With the help of LSTM, the improved version can directly extract temporal features from the RGB frames. For the experiment part, a popular video classification dataset (UCF-101) is used to check the general performance. In addition, a video anomaly dataset (FIRESENSE) is also added in the experiment to verify if the model can handle such tasks.

In Chapter 4, the two-stream model from the previous chapter is enhanced. To fit the video anomaly detection tasks, the single LSTM has been replaced by the combination of C3D and LSTM. Due to the application of C3D, the efficiency of the stream is improved. Considering that the whole model would be too complex, making it hard to be trained, several actions have been taken to simplify the model and the training process.

Chapter 5 presents the EPVP system, which can be seen as an application of video anomaly detection. This system allows more video sources such as videos captured by smartphone cameras, which may extend the current surveillance system in our city.

Chapter 6 summarizes the thesis and discussions and gives some potential directions for future works.

Chapter 2

Literature Review

2.1 Spatial Features Extraction Techniques

2.1.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is one of the most classic structures in deep learning architecture. It has an excellent performance in spatial features extractions for images and videos. The idea of CNN was inspired by the animal visual cortex organization 50 years ago [15]. They found that the structure of cells arrangements in the animal visual cortex helps animals feel the light and detect the visual field. The first Neural Network model that used a similar architecture was ”neocognitron” [16]. This paper provides an unsupervised method, which contains several layers of cells for visual pattern recognition. Compared with modern CNN models, this model does not require a shared wight.

In 1987, Time-delay Neural Network (TDNN) [17] was developed for the speech recognition tasks. This model can be seen as the first one-dimension CNN structure. In a TDNN, the weights are shared in a temporal dimension, which reduces computation. In 1988, CNN was used in the computer vision field for the first time [13]. The two-dimension CNN is proposed for the medical images analysis. However, because of the limitation of computing power and datasets, CNNs were not widely used at that time.

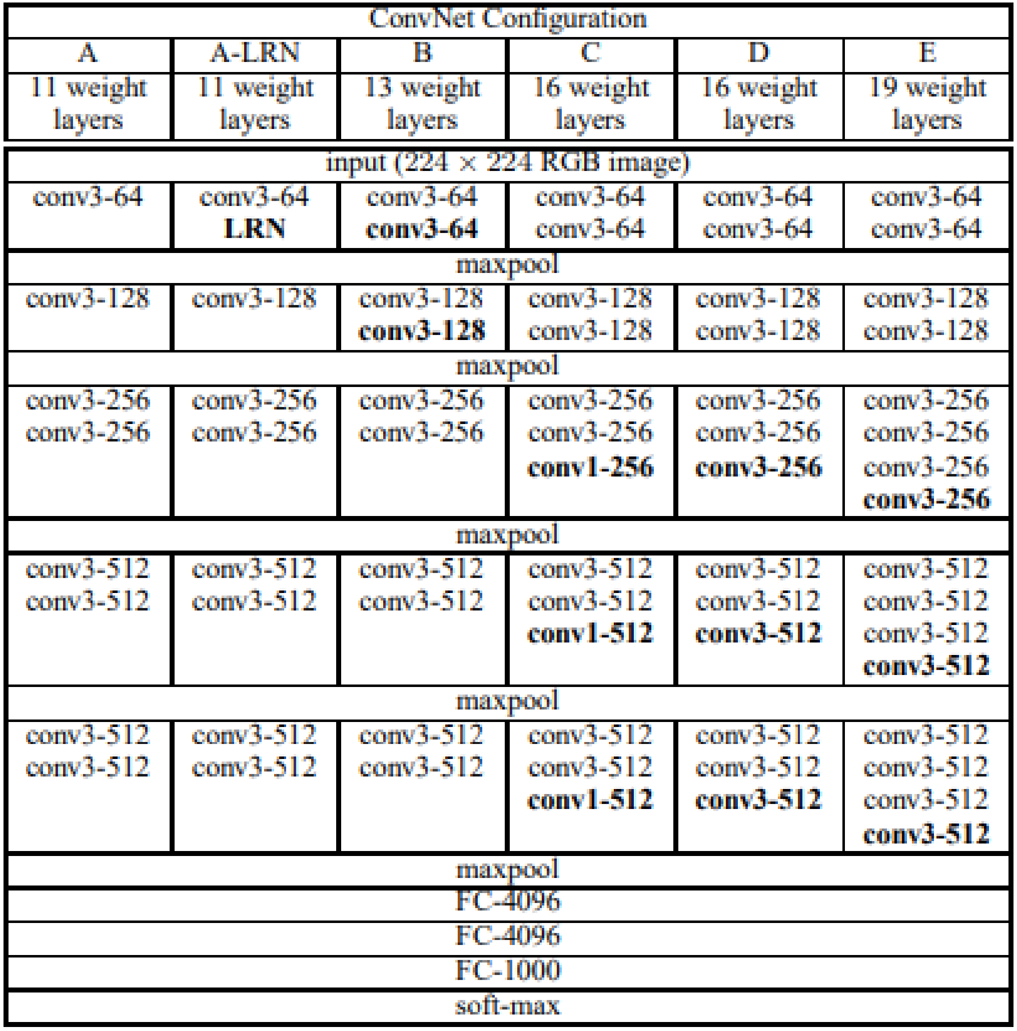
With the rapid development of deep learning and computation techniques, CNNs have become popular in the past 15 years. The GPU-accelerated computing techniques have been exploited to train CNNs more efficiently. Nowadays, CNNs have already been successfully applied to image and video tasks such as handwriting recognition, face detection, behaviour recognition, speech recognition, recommendation systems and image classification.

In CNNs, the convolution has replaced the general matrix multiplication in standard NNs. In this way, the number of weights is decreased, thereby reducing the complexity of the network. Furthermore, as raw inputs, the images can be directly imported to the network, thus avoiding the feature extraction procedure in the standard learning algorithms. It should be noted that CNNs are the first genuinely successful deep learning architecture due to the successful training of the hierarchical layers. The CNN topology leverages spatial relationships to reduce the number of parameters in the network, and the performance is improved using the standard backpropagation algorithms. Another advantage of the CNN model is that it requires minimal pre-processing. Some convolutional Neural Networks used in the following chapters are introduced in the below chapters.

2.1.2 VGG 16

VGG16 is a Convolutional Neural Network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" [2]. The model achieves 92.7% top-5 test accuracy in ImageNet [18], which is a dataset of over 14 million images belonging to 1000 classes.

The structure of the VGG16 is shown in Figure 2.1. It contains 13 convolutional layers and three fully connected layers. All hidden layers are equipped with rectification (ReLU) non-linearity. The size of every convolutional kernel in this structure is 3×3. All 13 convolutional layers can be divided into five blocks. Block 1 contains two layers. The number of channels in this block is 64. Block 2 also has two convolutional layers, while the number of channels is 128. Block 3 contains three convolutional layers, and the number of channels is increased to 512. The last two blocks both have three convolutional layers, and the number of channels is 512. After each block, there is a max-pooling layer, whose size is 2×2. Then three fully connected layers are set after the fifth block. The first two have 4096 channels each. The third contains 1000 channels. The number of channels in fully connected layers can be changed according to the requirements of different datasets and tasks. The final layer is the soft-max layer. The data processing part of the VGG16 is shown in Figure 2.2.



ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.1: The structure of VGG16 [2]

VGG16 mainly improves the previous CNN structure by adding two modifications. Firstly it reduces the size of the convolutional kernel to 3x3. Compared with the big kernel, small kernels can reduce the computation and parameters in the network. Though

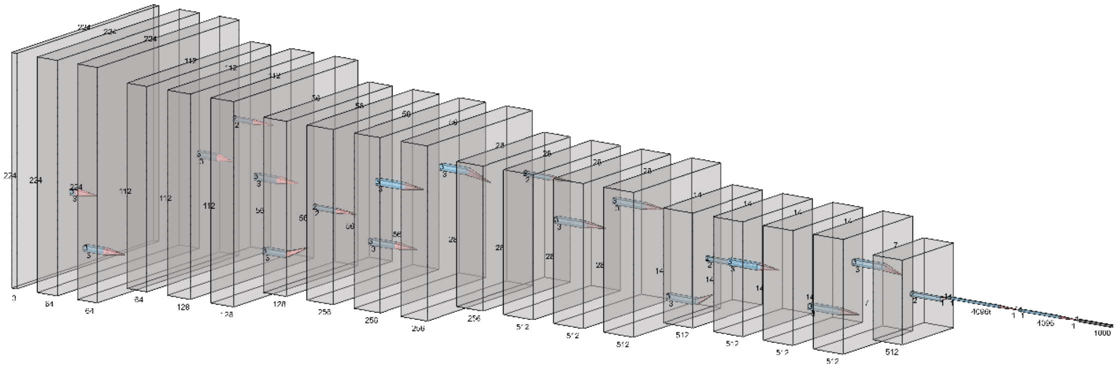


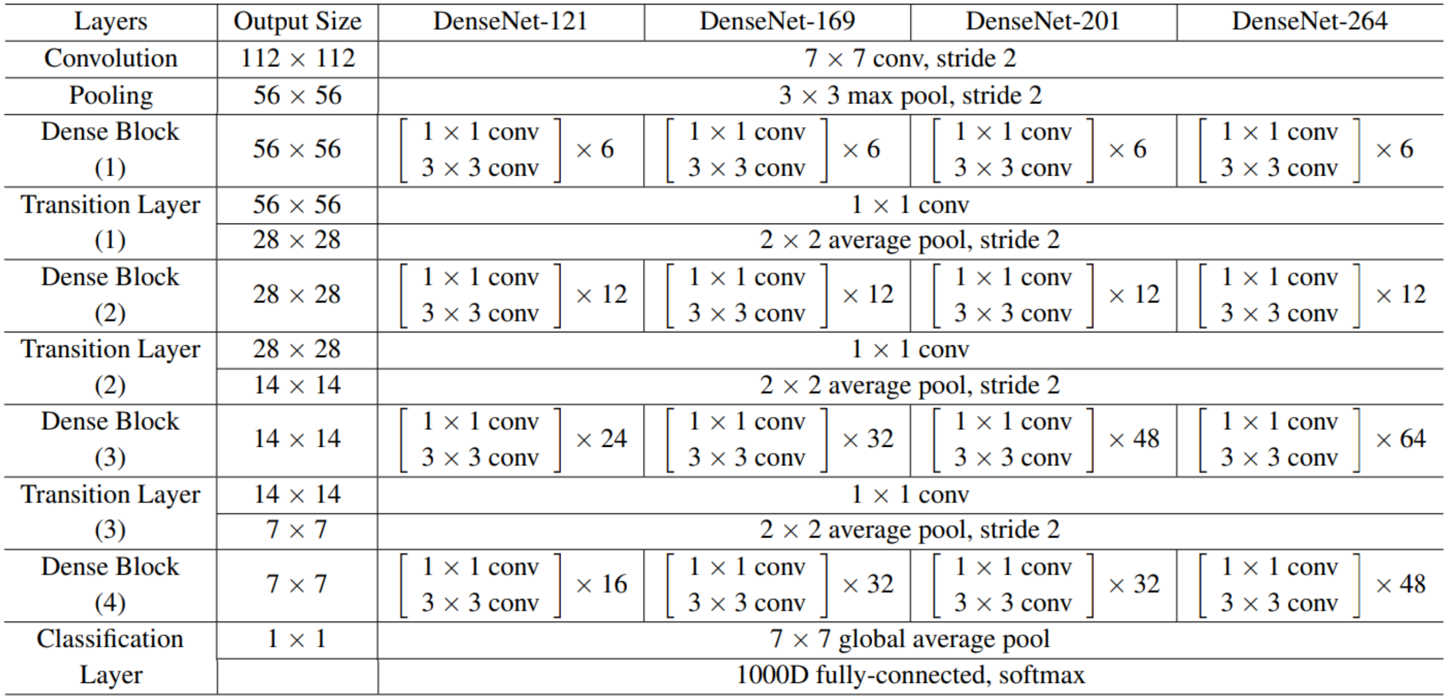
Figure 2.2: How the data be processed in VGG16

one big convolutional kernel can get more spatial information, small kernels can overcome this drawback by increasing the number of kernels. In addition, the structure of VGG16 proves that the performance of a CNN is affected by its depth.

2.1.3 Dense Convolutional Neural Network (DenseNet)

A DenseNet is a type of Convolutional Neural Network that utilises dense connections between layers [3]. Before the DenseNet, the directions of CNNs are mainly in two aspects: increasing the depth of the network or making every layer in the network bigger. However, DenseNet focuses on the features. In traditional CNNs, the n_{th} layer is only connected to the $(n+1)_{th}$ layer. For example, in the VGG16, the 1_{st} convolutional layer is not connected to the 10_{th} layer. With the input increase, this feature may cause the vanishing gradient problem, which affects the final model.

DenseNet uses several dense blocks in its structure. Every dense block contains several convolutional layers. Unlike the VGG16, layers in the same block are related to each other. Therefore, every layer contains the output features of all the above layers in the same block. In general, the relation among different layers is enhanced a lot in the DenseNet. The basic structure is shown in Figure 2.3 [3]. There are four kinds of DenseNets in this figure. All of them keep a similar structure. In general, DenseNet keeps the design of small convolutional kernels. After each average pooling layer, there will be a 1×1 Convolutional layer that is used as the transition layer between two dense blocks.



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 2.3: The structure of DenseNet [3]

The advantage of this model is that it needs fewer feature images than other models. Since each layer receives feature maps from all preceding layers, a network can be thinner and compact. In addition, the relation among layers is enhanced, more information can be collected in a dense block so that fewer parameters and feature images are needed to ensure the stability of the whole training process. Each layer has direct access to the gradients from the loss function and the original input signal, leading to implicit deep supervision.

2.2 Temporal Features Extraction Techniques

2.2.1 Recurrent Neural Network (RNN)

Recurrent Neural Networks are a class of networks that are always used for sequential or time-series data processing tasks such as Natural Language Processing (NLP) and video processing. RNNs and other traditional networks distinguish mainly by the ”memory” [14]. An RNN allows the previous inputs to affect the current output.

The comparison between RNNs and other networks is shown in Figure 2.4. x means the value of the input layer, and o is the value of the output layer. In the hidden layer, U, V and W present three matrices to handle values. S is the value of the hidden layer. In the traditional Neural Networks, the whole structure is just like a line. Every input x is handled by U and V directly, and the structure gets the output. Therefore, the x_1 is only used for the output O_1. However, in the hidden layer of RNN, there is an additional Matrix W to handle the values of previously hidden layers. So the next output O_2 is still affected by the previous input x_1. The detailed working process for the hidden layer is shown in Figure 2.5. In this figure, for the moment t, the network receives the input x_t and outputs O_t. However, O_t is not only decided by x_t, but also S_{t-1}. Since S_{t-1} is gotten from x_{t-1}, the network successfully handles the temporal information between moment t and $t-1$. There are two functions below to decide how the previous inputs affect the current output:

$$O_t = g(V \cdot S_t) \tag{1}$$

$$S_t = f(U \cdot x_t + W \cdot S_{t-1}) \tag{2}$$

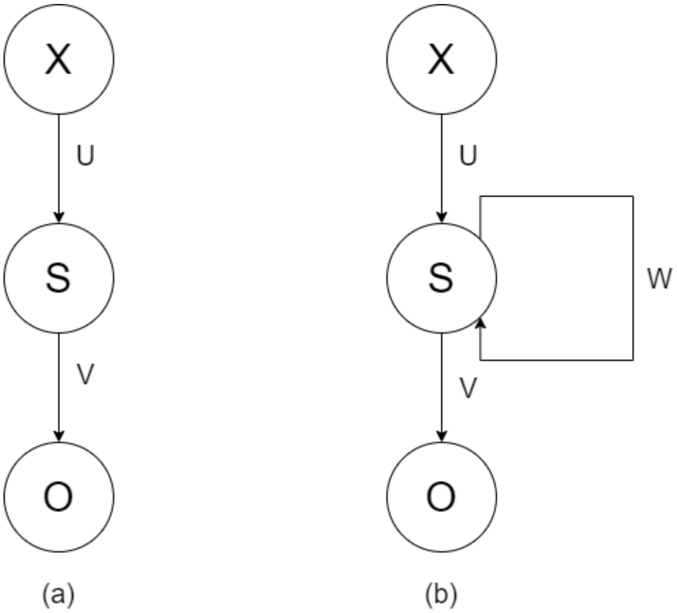


Figure 2.4: (a) Traditional structure for a artificial Neural Network, (b) Structure for a simple RNN

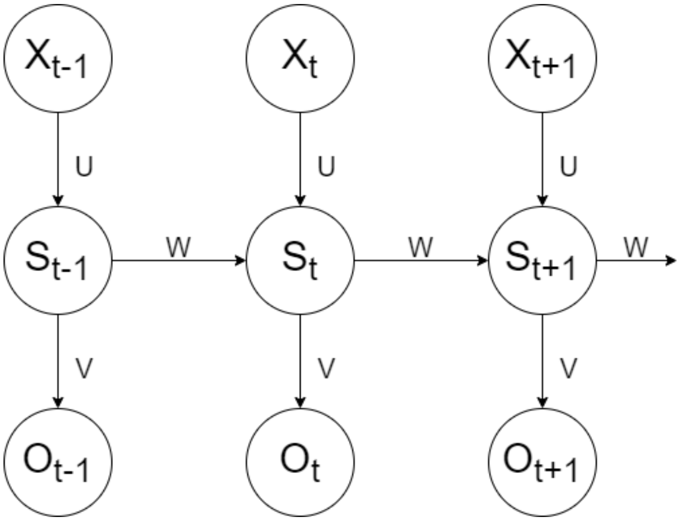


Figure 2.5: Unfolded structure for a simple RNN

2.2.2 Long Short-term Memory (LSTM)

The RNN solve the problem of handling sequential data. However, the basic structure of RNN could easily face the problem of gradient disappearance and gradient explosion when the sequential data are too long. Because all RNN units share a same set of parameters, features of earlier input may be deleted or decreased. In Figure 2.6, the input x_1 is multiplies by W for many times as a part of S. Features in x_1 will be reduces by above multiplication, which makes it hard to affect the output O_{t+1}. This problem is severe when we use the RNN to handle video-related tasks.

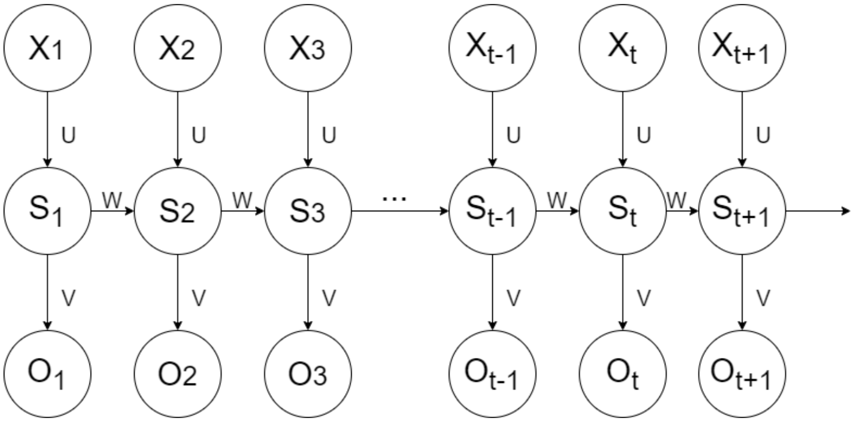


Figure 2.6: Potential problems when the input is too long

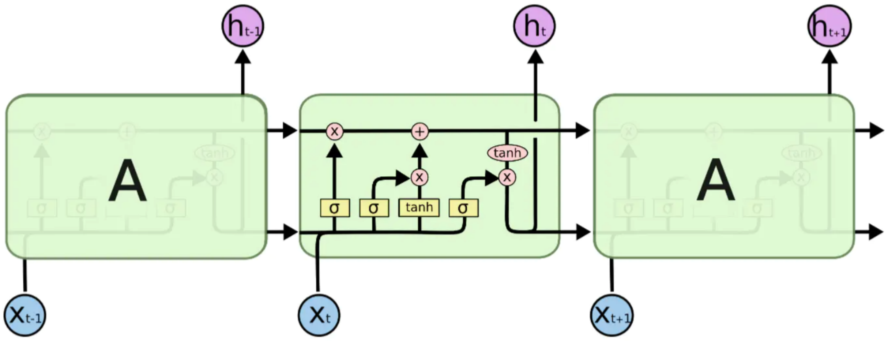


Figure 2.7: The structure of LSTMN

As the replacement, Long Short-term Memory (LSTM) is one of the solutions to improve the basic RNN structure [19]. In a basic RNN, the only state that will be transferred to the next moment is the hidden state S_t. However, LSTM introduces a new state called cell state (C_t) for the transmission. In every LSTM unit, there are gates to judge if the input is necessary to be kept, updated or transferred to the output. Therefore, LSTM can handle inputs and hidden states for an indefinite length of time, which solves the problem of the traditional RNNs. The structure of LSTM is shown in Figure 2.7. The detailed introduction of LSTM is in Chapter 3.

2.2.3 Two-stream Based Model

Besides the RNNs and LSTM, the two-stream model is another choice for temporal features extraction. Unlike the LSTM, which tries to get information from sequential data, the two-stream model uses optical flow for the temporal features. The basic structure is shown in Figure 2.8 [4]. The whole model can be divided into two streams. For the first stream (spatial stream), its input is a random sampled frame of the video. The model used in this stream is a simple CNN with four convolutional layers and two fully connected layers. Therefore, the stream is essential for an image classification structure. The second stream (temporal stream) still uses a ConvNet structure. However, unlike the structure in the spatial stream, the input is formed by optical flow displacement fields between frames. The general procedure of how the optical flow is generated is shown in Figure 2.9. The optical flow contains both the horizontal and vertical motions information in the video. Therefore, this stream is used to extract temporal features. Each stream outputs its softmax scores for later fusion.

The idea of the two-stream structure is quite straightforward. Since every video can directly be decomposed into spatial and temporal components, the structure handle these two parts separately. Below are some advantages of this structure:

- It provides the basic structure for handling the spatial and temporal features separately.

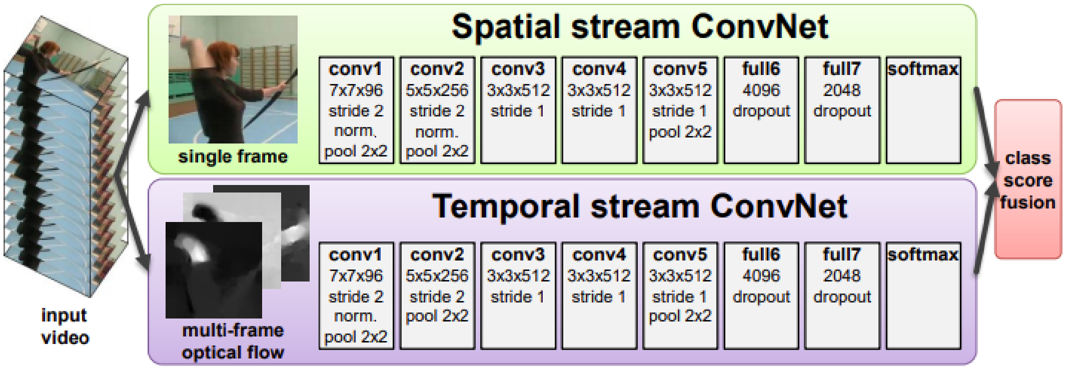


Figure 2.8: Two-stream structure for video classification [4]

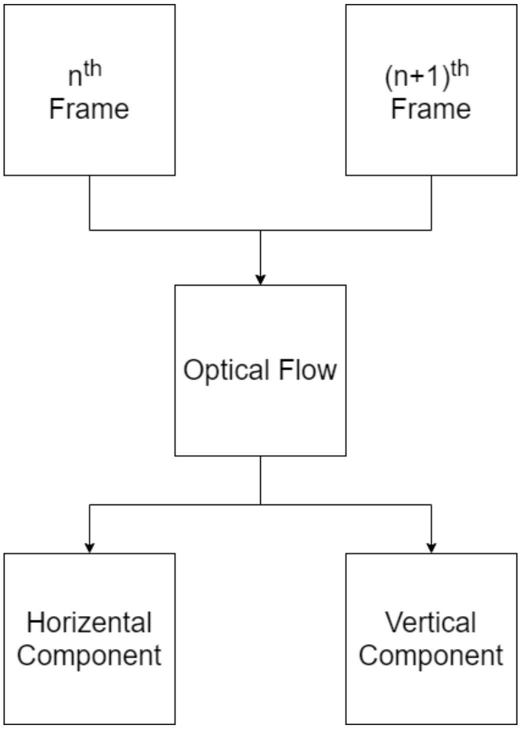


Figure 2.9: Description of an optical flow

- The performance of the two-stream model is better than the LSTM.

However, it also has some drawbacks that needed to be overtaken:

- It requires additional computing power and time to generate optical flow.
- The CNN structures in both two streams are not very advanced.
- For the long-term videos, the traditional two-stream structure cannot achieve ideal performance.
- Different methods for generating optical flows may get different results.
- The process of generating optical flow is easily affected by the moving of the camera, which means the model can only achieve the best performance in videos taken by still cameras. Otherwise, it needs compensation algorithms to handle this problem. Normally, the mean flow subtraction is used to avoid this situation. This method deletes the average value of the optical flow for every motion vector, so the influence of cameras can be reduced. However, for the valuable motions, their temporal features will also be reduced, which may affect the performance.

2.2.4 Convolutional 3D (C3D)

Because of the wide usage of CNN in computer vision tasks, the problem of how to extract temporal features by convolutional structure has been discussed. Mainstream CNNs ignore the temporal information in videos. Therefore, they can only focus on static images or short time sequence images. 3D convolutional networks (C3D) is a classic model which extends the traditional CNN. Figure 2.10 [5] shows the differences between 2D and 3D convolutional kernels. With a 2D kernel, no matter it is used in a still image or a sequence of images (video), the output is a 2D feature map (Figure 2.10: (a) and (b)). However, the 3D kernel can extract temporal information by the additional dimension (Figure 2.10: (c)).

Figure 2.11 gives the structure of the original C3D model. The whole structure is similar to the 2D CNN. It has eight convolution layers and two fully connected layers. All 3D convolution kernels are set to be 3 x 3 x 3, corresponding to length (number of input frames), height and width. All pooling kernels are set to be 2 x 2 x 2. The first convolution layer has 64 filters. After every pooling layer, the number of filters in a convolution layer is doubled.

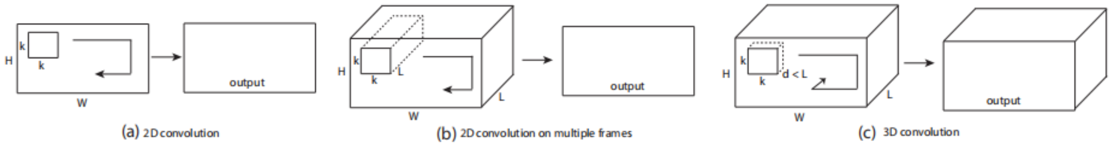


Figure 2.10: 2D and 3D convolution operations. a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal [5]

.

The C3D inherits some advantages of traditional 2D CNNs. The kernels in the model are all the same so that the pressure of computation can be reduced. The model has a good performance for local feature extraction. With the increase of the number of convolutional layers, it can extract global features. However, the structure inevitably becomes more complex due to the additional dimension, and the training process becomes complicated. The features for every 3D convolutional layer are hard to explain, which make C3D a black box model [20].

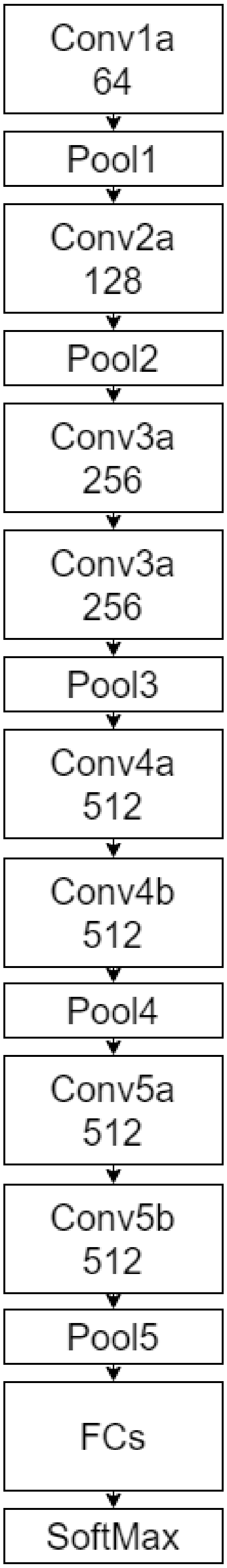


Figure 2.11: C3D architecture [5]

Table 2.1: Comparisons of feature processing methods

Methods	Advantages	Disadvantages
VGG16	1. It reduces the kernel size. 2. It increases the depth of traditional CNNs.	1. Only spatial features can be extracted.
DenseNet	1. The relationships of convolution layers are enhanced. 2. The model needs fewer feature images and parameters. It has a good anti-overfitting performance.	1. Current deep learning platforms cannot have good support for the DenseNet structure, which leads to the waste of GPU memory.
LSTM	1. It solves the problem that the basic RNN cannot handle long-term video.	1. LSTM requires more computer powers than CNNs.
Two-Stream Model	1. It requires additional computing power and time to generate optical flow. 2. The performance of the two-stream model is better than the LSTM.	1. The CNN structures in both two streams are not very advanced. 2. The process of generating optical flow is easily affected by the moving of the camera.
Convolutional 3D	1. It inherits some advantages of traditional 2D CNNs. 2. It can handle the temporal features.	1. The structure becomes complex.

2.3 Anomaly Detection Methods

Video anomaly detection is the problem of detecting anomalies in videos. Unusual events do not happen very often, which make a person hard to watch the videos and recognize the potential abnormal things happening in the video. Therefore, it is necessary to develop the method to find the patterns that do not conform to what is considered normal in videos. We call such tasks anomaly detection [6]. The target is to develop a robust algorithm that can automatically monitor and detect unusual events in videos.

An example of a simple anomaly case that can be seen in Figure 2.12 where the normal regions are denoted by N and anomalies are those denoted by O. As seen in the figure, anomalies tend to lie outside what is normal clearly. However, these anomalies can be close to normality which O_2 illustrates.

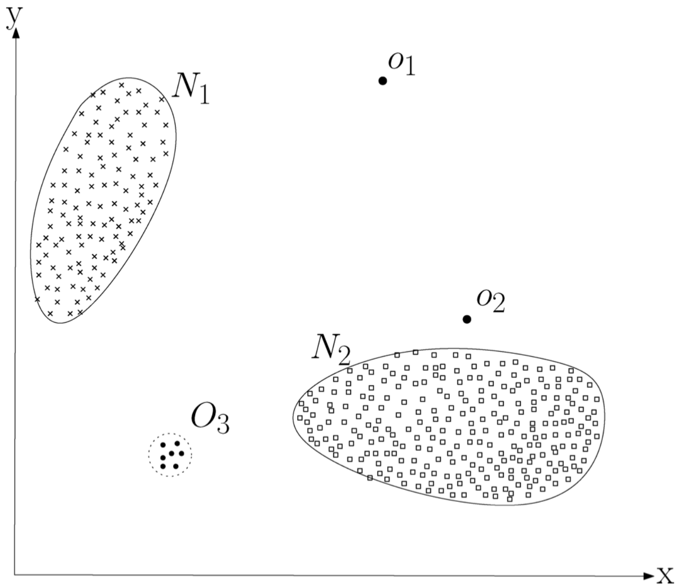


Figure 2.12: A case of anomaly detection [6]

Video anomaly detection approaches can be mainly divided into two directions. Traditional methods focus on the clustering-based detection [21, 22, 23, 24, 25] and low-level feature extraction. Since deep learning techniques have revolutionized in the field of computer vision [26, 27], deep learning-based methods have become the mainstream to solve video anomaly detection problems in recent years. These methods aim to train models to learn the features that detect anomalies according to these features.

2.3.1 Trajectories-based Methods

The principle of the clustering-based methods is the fact that an anomaly is always sudden and appears unusual features in a large range of videos [22]. Therefore, these methods can learn regular trajectories from normal events in a video stream. An unusual event will be detected if it cannot follow learned trajectories [21]. In order to improve the performance of the clustering, two models can be built to handle the spatial changes and movements in the video [24].

2.3.2 Low-level Feature Extraction Methods

However, learning trajectories from normal events are complicated for traditional clustering methods. In addition, to solve the problem that clustering-based methods are too dependent on the moving objects, low-level feature extraction methods focus on low-level presentations in the video such as the change of greyscale, moving vectors [28] and textures [29].

2.3.3 Deep Learning-based Methods

As the volume of video data increases because of the development of smart city, it becomes nearly impossible for above traditional methods to scale to such large scale data to find outliers. Since deep learning has been applied successfully in many fields, deep learning-based methods have become popular in such tasks.

Using reconstruction error is the most popular direction among deep learning methods [30, 31, 32, 33] for video anomaly detection. A model of normal videos is learned so that abnormal events will always show higher reconstruction errors than normal events since they are not as close as normal samples to the training data. Models in video anomaly detection tasks follow the basic structure of image-based models such as CNNs. To adapt this basic structure from the image to the video, methods for temporal features process are added such as LSTM, 3D Convolutional Neural Network, and Two-Stream Model [4].

Besides the reconstruction error, future frame prediction chosen by some models also uses autoencoders. They use autoencoders to generate anomalous frames [34, 35]. The basic structure of autoencoder is shown in Figure 2.13. The autoencoder does the work of frame prediction. It generates new frames with the same statistics as the training video. Since anomalies can be viewed as events that do not conform with certain expectations, they should not be similar to the predicted frame. According to it, the method can judge whether there is an outlier or not. Methods that follow this principle is called Generative Adversarial Network (GAN) [36].

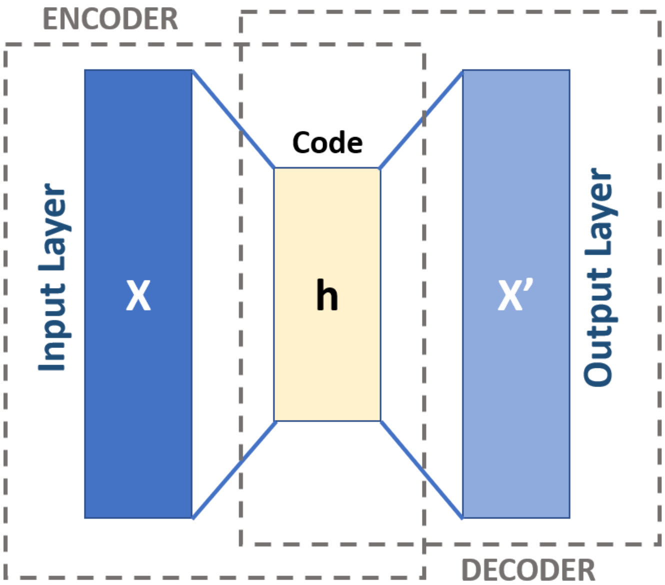


Figure 2.13: Structure of an autoencoder [7]

In addition, classifiers [37] and scoring methods [38, 39] are also used for video anomaly detection. The task can be considered a binary classification problem. The classifiers are designed to produce accurate and robust features for both normal and unusual videos. Similarly, if we consider this task as a regression problem, an anomaly score can be used to determine how likely the video is to be abnormal.

2.4 Summary

In this chapter, some previous related works are discussed as supplementary materials to the following chapters. The basic architecture of the two-stream model and LSTM is used in both Chapter 3 and Chapter 4. VGG16 is a comparison method in Section 3.6. C3D is modified in Section 4.2 to enhance the first stream of the two-stream model. Some anomaly detection methods are used as comparison studies in Section 4.3. A general comparison of different methods in this chapter is shown in Table 2.1.

Chapter 3

Improved Two-stream Based Model for Video Classification Tasks

Some contents of this chapter have been published in the following papers:

- Zhao, Y., Man, K.L., Smith, J., Siddique, K. and Guan, S.U., 2020. Improved two-stream model for human action recognition. EURASIP Journal on Image and Video Processing, 2020(1), pp.1-9.

- Zhao, Y., Zhang, J. and Man, K.L., 2020. LSTM-based Model for Unforeseeable Event Detection from Video Data. in Proceedings of The International Conference on Recent Advancements in Computing in AI, IoT and Computer Engineering Technology(CICET) 2020. pp.41.

3.1 Introduction

The video classification task is one of the most classic problems in the computer vision field. This section introduces a solution of using an improved two-stream structure to solve the video classification tasks. The videos in this part are mainly related to unforeseeable events and human actions.

Action recognition aims to recognize the motion and actions of objects. In the human action recognition field, vision-based action recognition is one of the most popular and essential problems [40]. It requires approaches to track and distinguish the behaviour of the subject through videos. Human action recognition is used in some surveillance systems and video processing tools [41]. In addition, the model of solving this kind of problem can also be used for other video classification tasks by transfer learning. Therefore, it is necessary to develop a new model to improve recognition accuracy.

With the development of surveillance systems, more video data can be captured in our city every day. These videos record almost every corner of the city and may include valuable information for the daily management of the city. Therefore, the demand for video processing capabilities increases significantly, especially in unforeseeable event management, such as fire detection and violence warning. The analysis of these videos provides an efficient method to distinguish the unforeseeable events in them, which is a quick response and saves human effort. The idea is to build the classifier by deep learning methods, such as different neural networks, to label the events in videos and extract potentially dangerous things from them. The critical challenge is to process videos and handle different unforeseeable events labels.

Based on the rapid development of computer vision and neural networks, vast improvements have been achieved in the video classification field [42, 43]. By using CNNs, spatial features from RGB video frames can be easily extracted, which is similar to its functions in image recognition [44, 45]. However, the critical challenge of such tasks is how to obtain and handle temporal features. Compared with still images, the video contains valuable temporal information that can enhance the accuracy of the action recognition [46, 47]. How to get and use the temporal features has become an important task in the video classification problem.

Current solutions can be divided into two approaches. One approach is to use models that can extract the temporal features, such as LSTM, in the final model [4]. LSTM uses three gates to decide which cell can be passed to the next layer or forgotten. Thus, it can keep the temporal information in the video. However, the input size of an LSTM would be much bigger than a CNN. Therefore, the training speed of such methods can be slow if they rely on the LSTM. A single LSTM-based model still has space for improvement according to experimental results shown in Section 3.2.

Another approach is to add an extra input stream, which can be the extracted temporal features using a CNN [4]. Optical flow is a popular input in this approach. It is a set of images, which presents the relative motion between the object and background in the video. Thus, the optical flow contains the features in a time sequence. Since there are two inputs for the CNN, the traditional method is to train two independent CNNs, one handling the RGB frames, and another one managing the optical flow. Then it combines the results of both training streams to get a final recognition result. This structure is known as the two-stream CNN model. However, the two-stream CNN model still has an apparent defect. The model does not contain the original temporarily information in the RGB video frames. Though the optical flow contains the temporal features, it only records the movements of pixels in the x-axis and the y-axis.

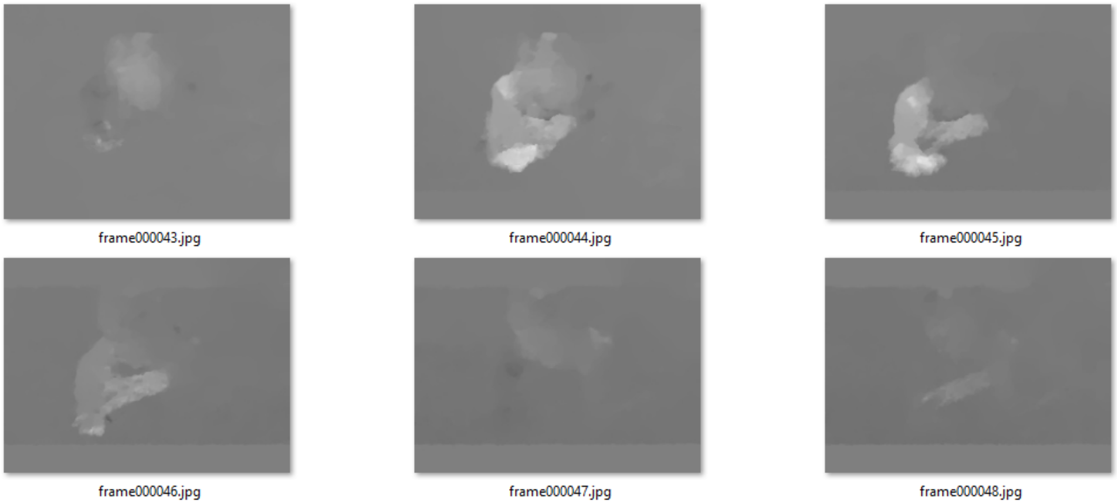


Figure 3.1: Part of an optical flow of a video in UCF-101. It is a set of grayscale images extracted from the RGB images. It identifies the motion of the subject in the video

Therefore, in this chapter, we aim to solve the limitations of previous solutions. The idea is to make a combination of these two approaches. The new model keeps the temporal stream so that a CNN can still process temporal features from the optical flow. For the spatial stream, we use an LSTM-based model to replace the traditional CNN in order to extract more temporal features from the RGB frames.

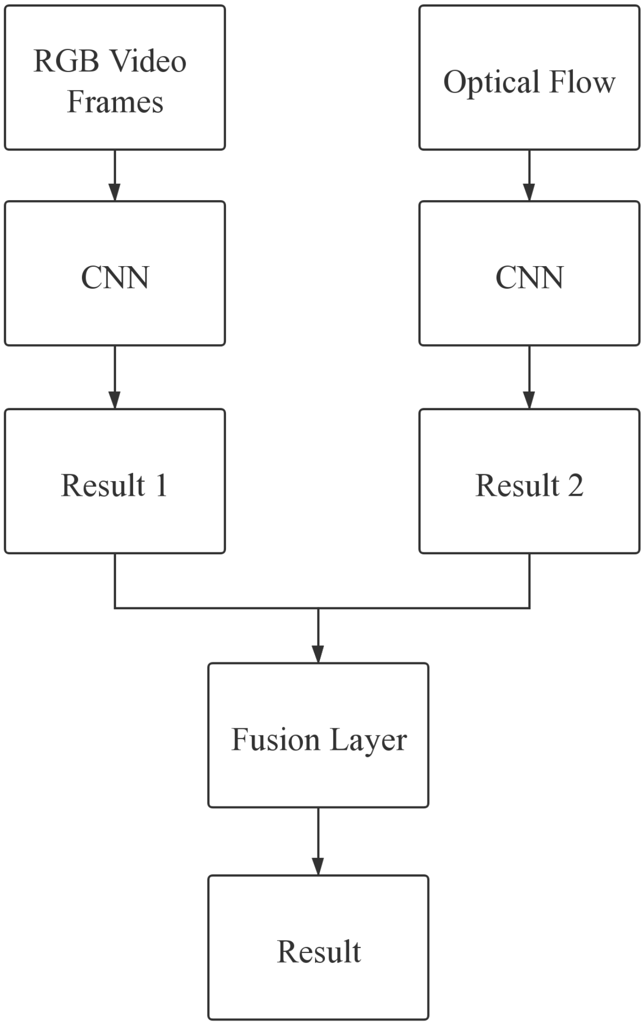


Figure 3.2: The structure of the two-stream CNN model

3.2 The overall Structure

In this section, the proposed model can be decomposed into three modules. They are a spatial stream with the LSTM, a temporal stream with a DenseNet and a fusion layer with Support Vector Machine (SVM) [48].

The overall structure of the model and the general training process is shown in Figure 3.3. Firstly, the training data are RGB video frames and optical flow. The training process can be divided into three parts. For each video, a sequence of sampled RGB video frames is processed by the spatial stream. The LSTM-based model in this stream trains the network and gets a recognition result by marking grades for different labels. According to the sequence of frames, the corresponding optical flow is inputted and processed in the temporal network. DenseNet is used in this stream for training. Because the whole training process is by supervised learning, every optical flow is also labelled. DenseNet also provides its recognition results. So far, there are two results from the above streams. Finally, the fusion layer fuses the results of the two streams to get the final recognition result.

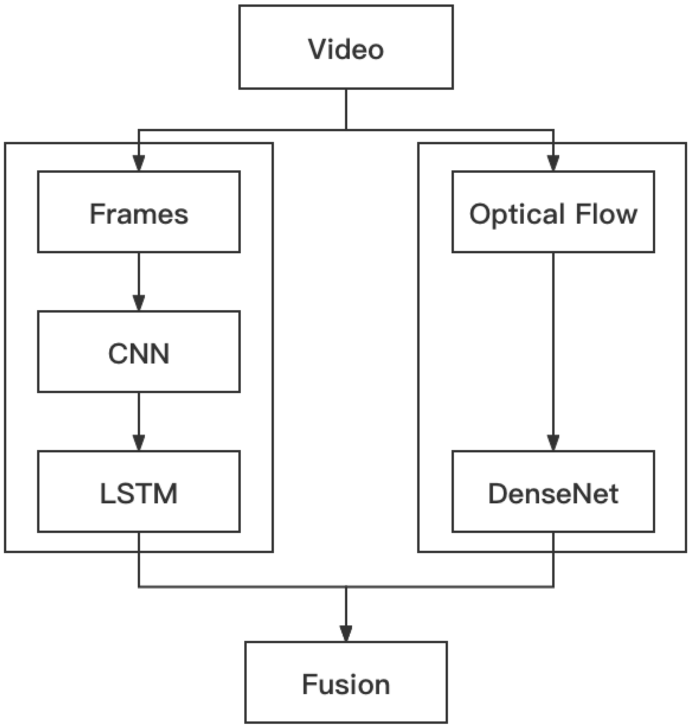


Figure 3.3: The overall structure of the proposed two-stream model

3.3 Stream for the RGB Frame

There is an LSTM-based model in the spatial stream, which uses a convolutional neural network for feature extraction and an LSTM network to do further classification.

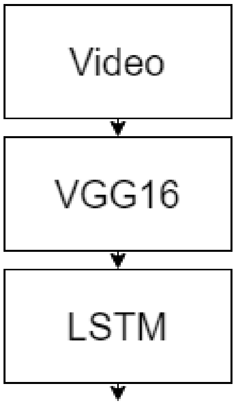


Figure 3.4: The proposed structure of the spatial stream

3.3.1 Video Processing

Unlike the image classification tasks, video classification requires extracting the same number of frames from videos with different total frames so that the size of the input of the stream can be unified. The traditional method is to set a specific value K as the number of input frames. According to K and the total number of the input video (N), the model can pick one frame from every S frame by calculating S = N/K (round down). After it reaches K frames, it stops picking frames. When the number of input frames is set to be 16, and the video has 32 frames, it is easy to pick one frame from every two frames. However, if we have 47 frames, according to the calculation by 47/16, the method will still get one frame from every two frames. As a result, after 16 frames are picked, the last 15 frames in the video are dropped by the method just like Figure 3.5, which may cause the loss of features.

In the proposed model, it still calculates the S by N/K. However, the method keeps picking one frame from every S frame until the end of the video. Then M is set to be the number of picked frames. According to the equation T = M/(M-K) (round down), some picked frames will be deleted. For example, if N is 47 and K is 16, S should be 2. Then

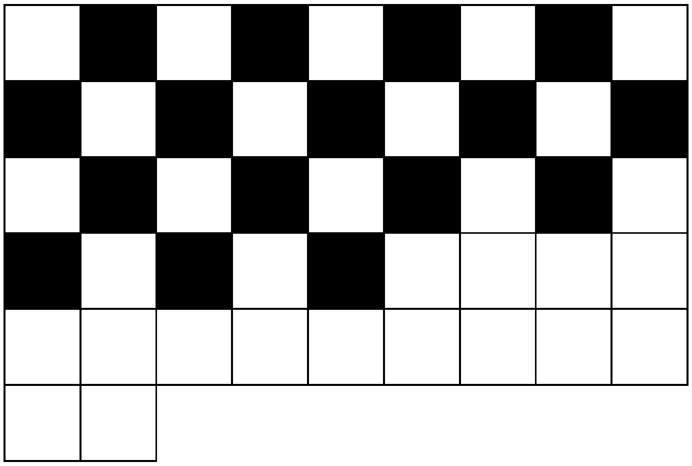


Figure 3.5: The sampling method for traditional models: Picking 16 from 47 frames. black squares mean the picked frames and white squares present the dropped frames

23 frames are picked because the method does not stop after picking 16 frames. The T equals 23/(23-16) = 3, which means the method will delete one frame from every 3 frames. Compared with Figure 3.5, Figure 3.6 shows that the new method can extract 16 frames from the video more evenly.

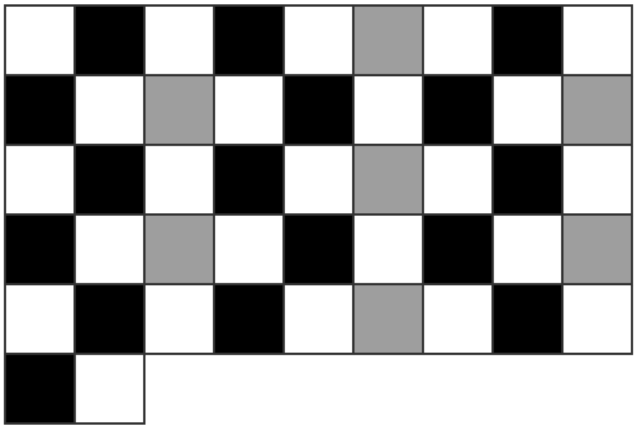


Figure 3.6: The sampling method in the proposed model: Picking 16 from 47 frames. Black squares mean the picked frames, White squares present the dropped frames, Gray squares present deleted frames in the second process

3.3.2 Spatial Feature Extraction

The Visual Geometry Group (VGG16) network is modified slightly in this model for the spatial feature extraction work [2]. VGG16 is a CNN provided by Oxford University and has been widely used in the image classification field. It is pre-trained on the ImageNet dataset [18], and the weights and layer configuration are available on the official website. In VGG16, there are sixteen hidden layers, including thirteen convolutional 2D layers and three fully connected layers.

The input RGB video frames are resized to 224*224 to fit the default input size of this model. In the original VGG16 model, the last fully connected layer is used for the image classification. Thus it has been deleted in the proposed model. Finally, this model outputs the input frames' sequence of features and passes these features to the next stage. Assume that the dimension of the feature captured by the CNN part is N (N is 4096 in VGG16). For the video with K frames, the CNN part will output a sequence of N-dimensional features with the length of K. This sequence of features will be the input of the following LSTM part in the first stream.

3.3.3 Temporal Feature Extraction

In consideration of the fact that CNN is good at extracting spatial features, it is necessary to utilize temporal features from these RGB video frames. Since the input of the whole model is a sequence of features from the CNN part, an LSTM is built in the current model [19]. The LSTM network in this model is set to be a single-directional structure. It contains one LSTM layer and two fully connected layers. Figure 3.9 shows the structure of the kernel of the LSTM layer, where σ and tanh represent the activation functions, C and h represent the cell state and the hidden state separately; x is the input signal.

The LSTM uses three gates to determine which information is valid. Gates can be seen ways to control how the information through optionally. The below equations describe how the three gates work in the LSTM. W in these equations represents the matrix of parameters.

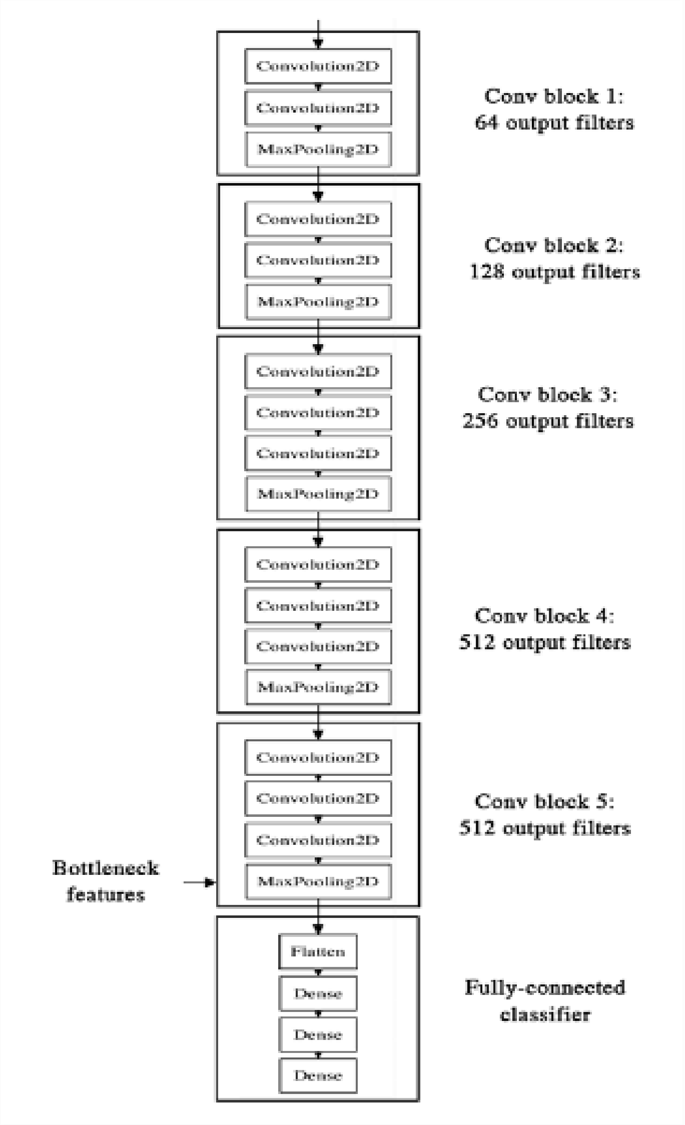


Figure 3.7: The structure of original VGG16 [2]

.

Related Equations [19]:

$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t]) + b_i \tag{1}$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t]) + b_f \tag{2}$$

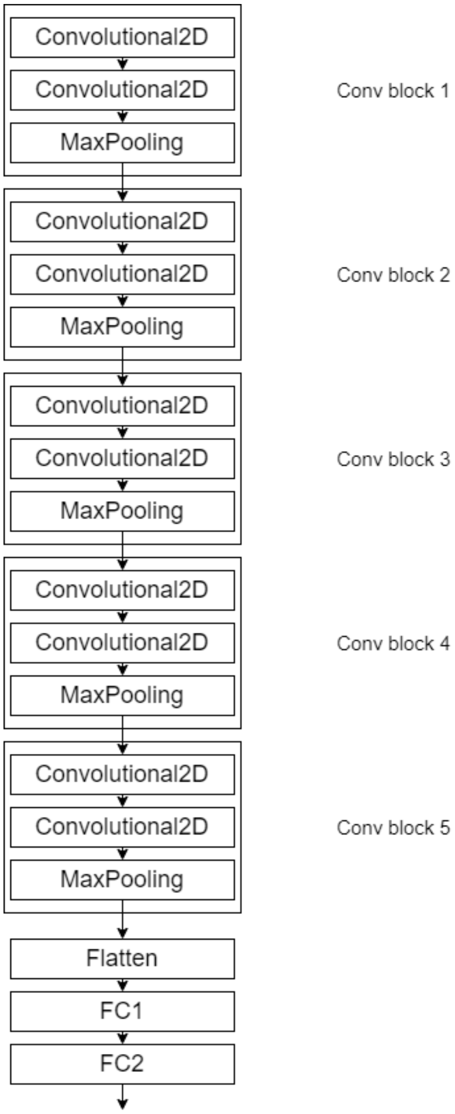


Figure 3.8: The structure of the CNN part in the proposed model

$$\widetilde{c_t} = \tanh(W_c \cdot [C_{t-1}, h_{t-1}, x_t]) + b_c \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{c_t} \tag{4}$$

$$o_t = \sigma(W_0 \cdot [C_t, h_{t-1}, x_t]) + b_0 \tag{5}$$

$$h_t = o_t \cdot \tanh{(C_t)} \tag{6}$$

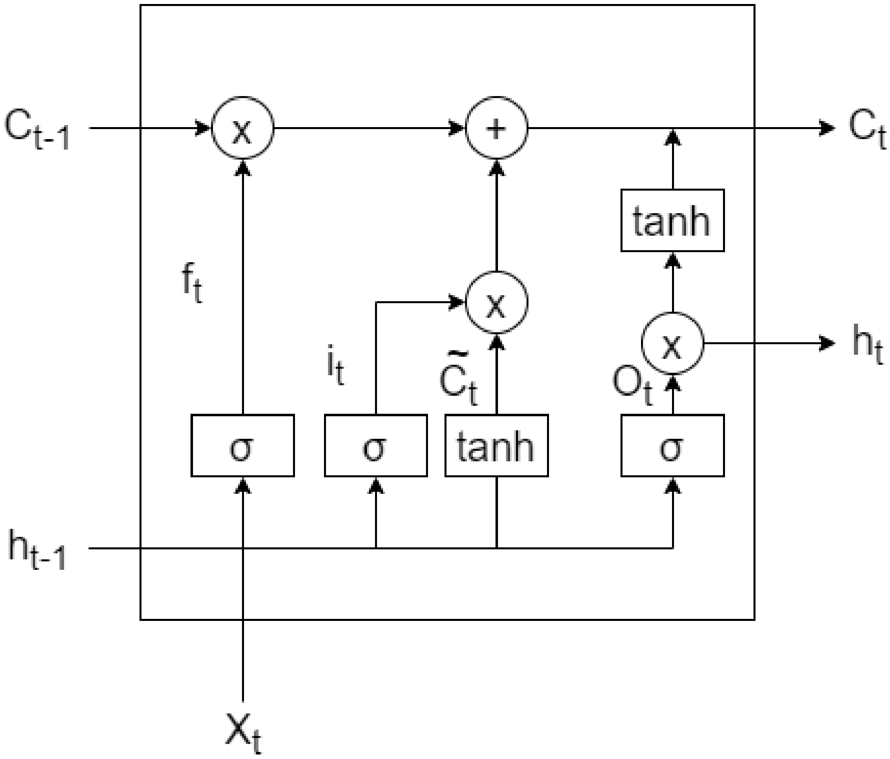


Figure 3.9: The structure of the LSTM

Firstly, the forget gate chooses the information which will not be used in the current cell. The working process of this gate is shown in Figure 3.10. The corresponding equation of the gate is equation (1). The forget gate is implemented by a sigmoid layer. It receives the previous output h_{t-1} and the current input x_t. Then the layer outputs a value between 0 and 1 for the last cell state C_{t-1} to decide whether the value of C_{t-1} and f_t can be kept or deleted.

The next step is to get the cell state for current cell by the input gate. The working process of this step is shown in Figure 3.11. The corresponding equations of the gate are equatios (2) and (3). An input gate, which is implemented by a sigmoid layer, decides which values the cell will update. The i_t in the Figure 3.11 is represented as its output. Then a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state. Finally, $\tilde{C}_t$ and i_t are combined to update the value of the current state.

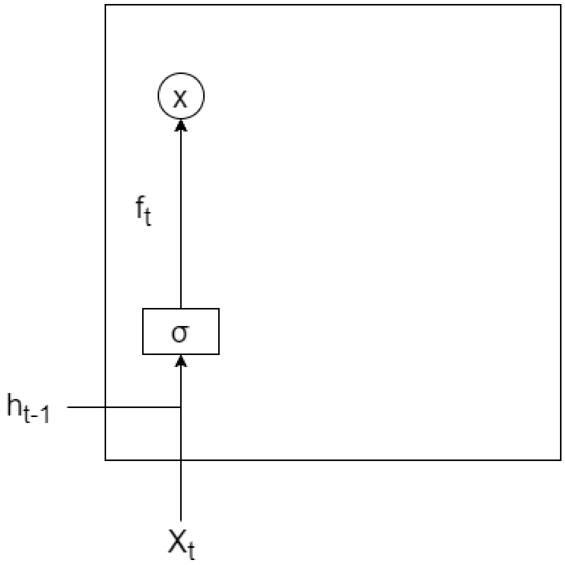


Figure 3.10: Figure for equation (1)

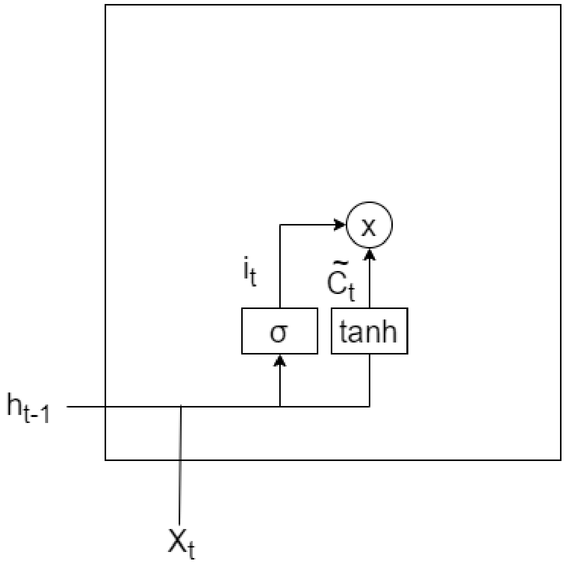


Figure 3.11: Figure for equation (2) and (3)

The cell state of current cell is also affected by the forget gate. The working process of this step is shown in Figure 3.12. The corresponding equation is equation (4). The results of forget gate and input gate are combined together to get the final C_t as the cell state, which will be transferred to the next LSTM unit.

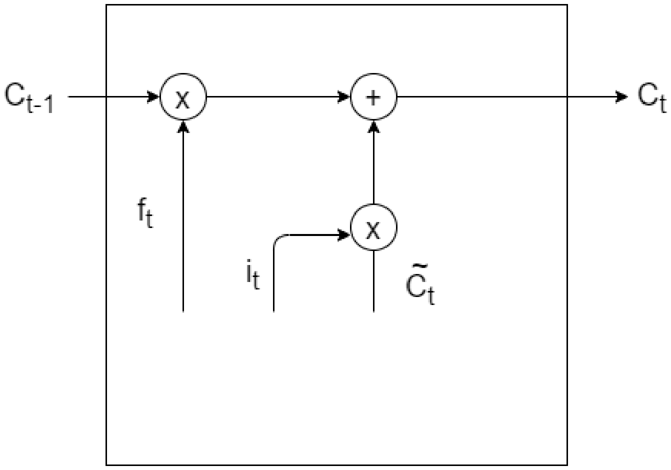


Figure 3.12: Figure for equation (4)

The last step is to determine the output of the cell. The process is shown in Figure 3.13. The corresponding equations are equations (5) and (6). Another sigmoid layer is applied in this layer to decide what parts of the cell state are going to be output. Then the tanh function is used to push the values between −1 and 1. Finally, the output value of this function is multiplied by the output of the sigmoid gate so that we only output the parts we decided to.

Since the training speed of an LSTM is much slower than that of CNN, if the input data of this stream is the standard sampled files of UCF-101, the training time will be too long. According to the tests, if the input data of the spatial stream is the image set provided by the Graz University of Technology, the training time of each epoch will be more than 2 hours under the processing of an Nvidia RTX2080ti GPU card. Therefore, a sampled script is added before the whole model to extract 25 frames from each video.

3.4 Stream of the Optical Flow

This section describes the Convolutional Neural Network, which is used in the temporal stream. The difference for this stream is that it uses a stack of optical flow images. As

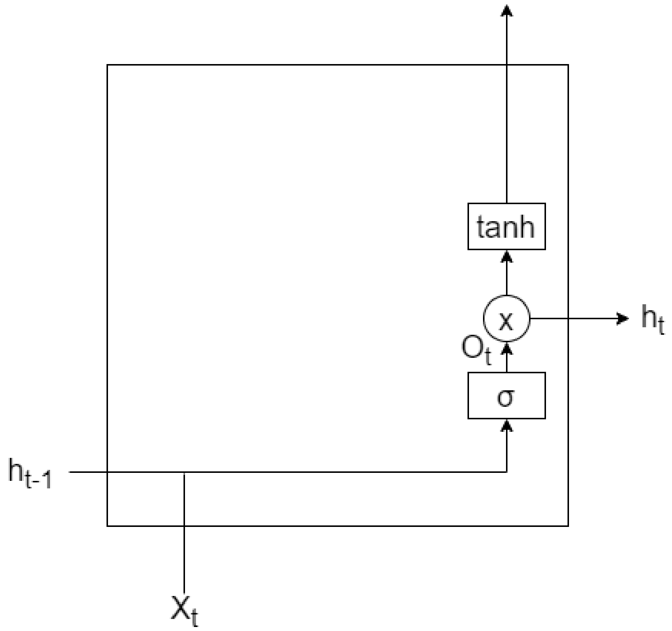


Figure 3.13: Figure for equation (5) and (6)

shown in Figure 3.14 [4], there are five variations of the optical flow-based input. In this work, (d) and (e) are chosen as the input data. The optical flow is generated by the same method in traditional two-stream CNN model [49], which outputs the optical flow based on intensity and gradient. A stack of optical images, which contains ten x-channel and ten y-channel images, is considered as an input. Therefore, the input shape is (20,224,224).

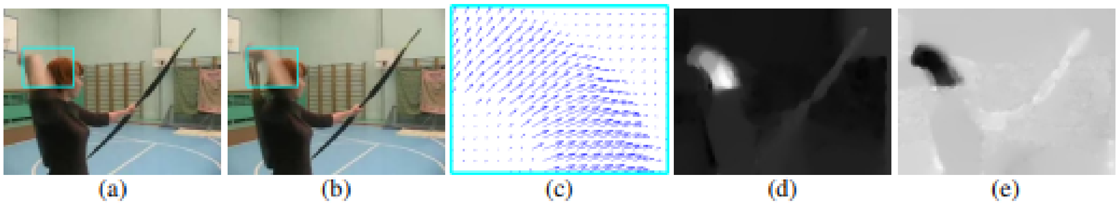


Figure 3.14: Optical flow: (a) & (b): RGB video frames, which are similar to the input data of the spatial stream. However, there are some rectangles highlighting the moving area in the frame. (c): the dense optical flow in the outlined area; (d): y-channel images of the displacement vector; (e): x-channel images of the displacement vector

3.4.1 Feature Extraction

The DenseNet is implemented in the temporal stream [3]. DenseNet uses several dense blocks in its structure, and every dense block contains several convolutional layers. Unlike the VGG net, layers in the same block are related to each other. Therefore, every layer contains the output features of all the previous layers in the same block. The relation among different layers is significantly enhanced in the DenseNet. Figure 3.15 [3] shows the basic structure of this network. The advantage of this model is that it needs fewer feature images than other models. In traditional CNNs, the convolution layer is connected one by one, which means the $(n+1)_{th}$ layer is only affected by the n_{th} layer. In DenseNet, due to the enhanced relationship between layers, more information can be collected in a single dense block. Therefore, we do not need many parameters and feature images to ensure the stability of the whole training process. Furthermore, the vanishing gradient problem will be solved because of the dense connections.

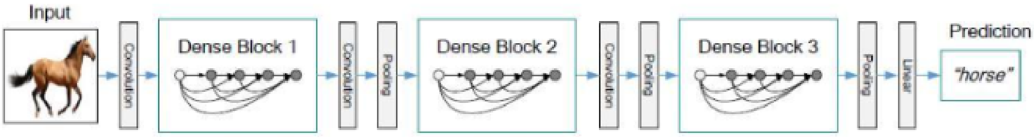


Figure 3.15: The structure of the Densenet

A basic DenseNet-121 is used in the proposed model. It contains four dense blocks and 58 convolutional layers. Since the optical flow of UCF-101 in this research is large enough, it is easy to avoid the over-fitting problem. Therefore, in the final classifier of this stream, we do not need to create several softmax layers for different datasets. As a replacement, we keep the original softmax layer for the UCF-101 classification.

3.5 Fusion Layer

Although the proposed model modifies both the spatial stream and the temporal stream, the outputs of these two streams are not changed. Each stream outputs its classification scores separately. The choices of the method in the fusion layer can be similar to the traditional two-stream CNN model. According to previous experiments of the two-stream

CNN model[4], the SVM has a better performance than the average method. The general procedure is shown in Figure 3.16.

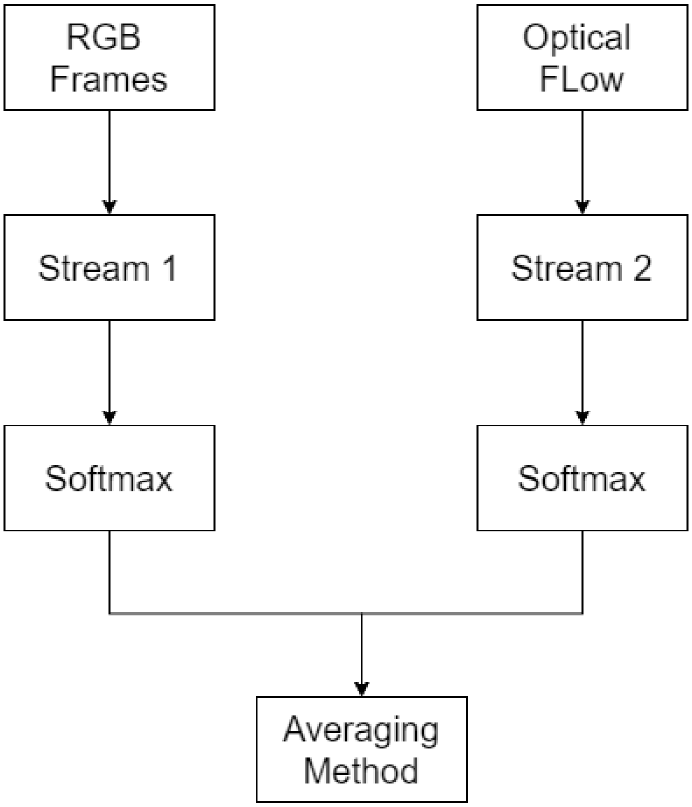


Figure 3.16: The procedure of fusion in the proposed method

3.6 Experiments

3.6.1 Experimental Setup

In this section, detailed setup of the experiment is introduced, which include the environment and datasets we use in the experiment.

3.6.1.1 Experimental Environment

In our experiments, the final accuracy is the average accuracy of all three splits. The ImageNet is used for the pre-trained part for both two streams. Finally, we evaluate the spatial stream, the temporal stream, and the whole model separately to check the individual performance of every component in the proposed model. The detailed environment is shown in Table 3.1. We use Keras and an Nvidia RTX2080ti GPU for the experiments. The GPU is powered by the Turing GPU architecture, which is helpful for deep learning research. Besides, it has an 11 GB GDDR6 frame buffer.

Table 3.1: Experimental Environment

Component	Name
GPU card	Nvidia RTX 2080ti
Platform	Keras
Language	python

3.6.1.2 Datasets

The UCF-101, the FIRESENCE database and the merged dataset of above two datasets have been used so far for the experiment and evaluation part.

The proposed model in this chapter is evaluated on the UCF-101 human action recognition dataset [50]. It contains 13320 labelled videos that belong to 101 human action categories, such as punching, boxing, and walking. All these 101 categories can be divided into five types: Human-Object Interaction, Body-Motion Only, Human-Human Interaction, Playing Musical Instruments, and Sports. All videos in this dataset are realistic and collected from YouTube. The UCF-101 dataset does not have a pre-divided training set and testing set. It gives the three official guides of training and testing splits for both action recognition and action detection. The UCF-101 is used for comparison since it is a very popular dataset for human action recognition. We followed the split methods and calculated the average accuracy of all three splits. In addition, both Top 1 and Top 5 accuracy are concluded in the final result.

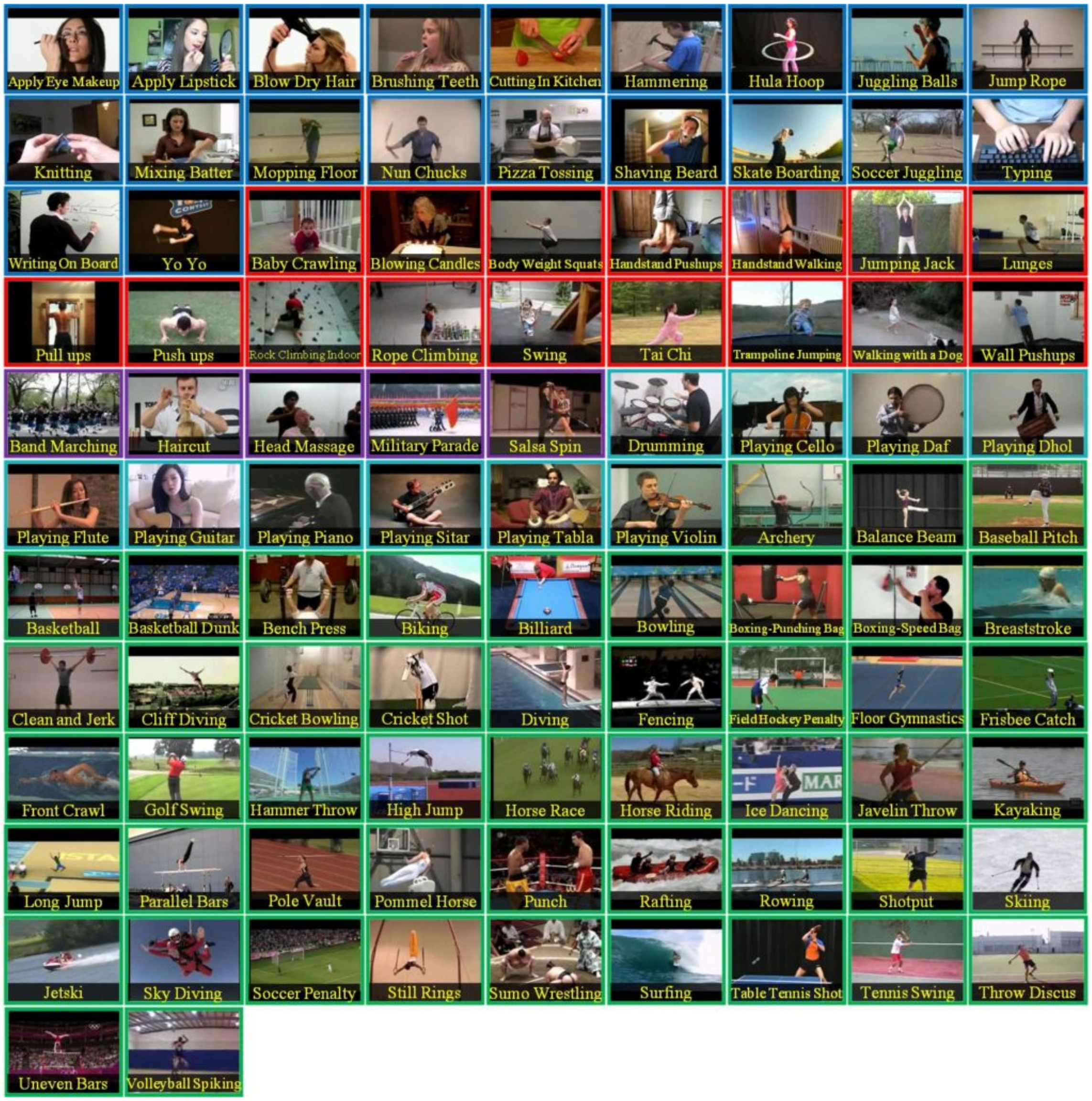


Figure 3.17: Screenot of 101 categories in UCF-101

.

FIRESENCE database [51] contains 49 labelled videos, including 11 fire videos and 13 serious smoke situations. There are other 25 videos without the above situations in the dataset as well. This dataset can be used for fire and smoke detections. This chapter assumes that both fire and smoke situations belong to the positive video and others belong to the negative videos. For the FIRESENCE database, there are only two classes. The positive class includes fire and smoke videos. The negative class contains other videos. The experiment thus becomes a binary classification problem.

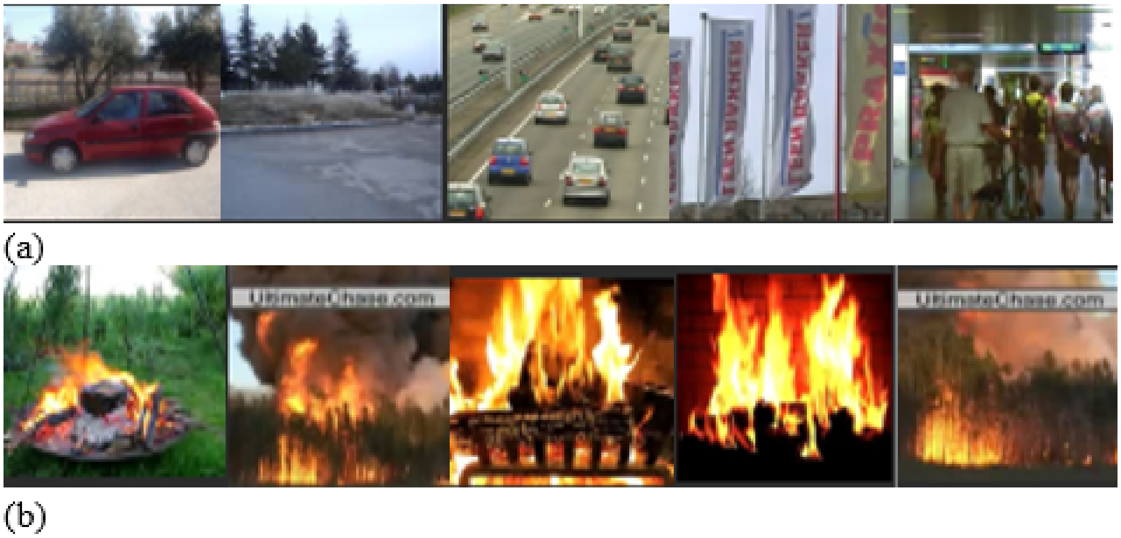


Figure 3.18: FIRESENSE data set: contains (a) negative videos and (b) positive videos
.

Finally, the merged database contains both the classes in the UCF-101 and the FIRESENCE database. This dataset is used to simulate the complex situations the model may encounter. For the UCF-101, we select two dangerous behaviours: punching and armed fighting. In addition, four classes of safe videos are selected as the comparison set. All these classes will be the input of the model. Since the data set contains both UCF-101 and FIRESENSE database, it does not have the initial training and testing sets. Thus the ‘test-size’ is set to be 0.3, which means for each class, 30% videos are randomly selected for validation and others are used for training. For every epoch, both accuracy and loss of the training and validation will be outputted to present the performance of the model.

3.6.2 Results and Discussion

In this section, we compare the experimental results between the proposed model and other state-of-the-art methods. We also discuss the implications and current limitations of our work.

3.6.2.1 Results on the UCF-101

We compare the accuracy of different neural networks on the UCF-101. Five scenarios are considered: (a) the spatial stream ConvNet, which is used in Google's first Two-stream CNN model [4]. (b) VGG16, (c) VGG16 and a bidirectional LSTM, and (d) VGG16 and a single-directional LSTM. In our experiments, all methods are pre-trained by ImageNet. In addition, the dropout is set to be 0.5. Besides the Top 1 accuracy, the Top 5 accuracy of LSTM-based models is also shown in Table 3.2.

Table 3.2: Different model accuracy on video frames of UCF-101

Model	Top 1	Top 5
Spatial Stream ConvNet [4]	72.7%	
VGG16	32.1%	51.3%
Inception V3 [52]	54.55%	79.92%
VGG16+LSTM(Bidirectional)	88.1%	96.72%
VGG16+LSTM(Single directional)	90.81%	98.61%

According to the results in Table 3.2, the method which uses a simple VGG16 has a poor performance. The spatial stream ConvNet can improve the top 1 accuracy by more than 40%. Furthermore, two LSTM-based models used in the proposed model can improve by another 16% on that basis. Between these two LSTM-based models, the single-directional LSTM seems to have better accuracy than the bidirectional one; However, the difference is less than 2%.

Here we compare the performances of different methods in the temporal stream. From Table 3.3, we can see the DenseNet can obtain the highest Top 1 accuracy, which is 3% higher than the ResNet101. As in the previous experiment, VGG16 get the lowest Top 1 accuracy.

We compare the performance of the proposed two-stream model with state-of-the-art methods on UCF-101. The performance is measured by the average accuracy on all three

Table 3.3: The Top 1 and Top 5 accuracies of the opticla flow CNNs

Model	Top 1	Top 5
ResNet101 [53]	76.1%	
VGG16	30.1%	46.5%
DenseNet121(Proposed Method)	79.63%	80.12%

splits of the UCF-101 dataset. In the experiment, we use the control variable method. If the model used Kinetics for the pre-trained part, it would undoubtedly get a higher recognition accuracy. To make the comparison fair, all models in these experiments are only pre-trained on ImageNet. However, we keep the CNN backbone of each method different so that the final accuracy can be authentic. In addition, only the Top 1 accuracy is considered in this experiment because most of the methods in this table do not provide the Top-5 accuracy for comparison.

Table 3.4: State-of-the-art performance comparison on UCF-101. The accuracy is the average accuracy for all three splits of the dataset

Model	CNN Backbone	UCF-101
Two Stream CNN [4]	VGG16	88.7%
Conv + LSTM[47]	AlexNet	69.1%
C3D[5]	VGG11	82.3%
RGB-I3D[54]	Inception v1	84.5%
TSN [55]	Inception v2	86.4%
3D Hybrid Model[56]	C3D	89.4%
Two-stream Model(Proposed Model)	DenseNet	92.5%

Table 3.4 presents a quantitative comparison of the experimental results. According to this table, the proposed two-stream model gets the highest Top-1 recognition accuracy amongst all methods, which is 92.5%. Compared with the state-of-the-art two-stream CNN method, the proposed model outperforms it by more than 3% with similar basic architectures. Besides the improvement of networks in both two streams, the input of the

first stream is changed from a single frame to a sequence of frames. It solves the problem that for the traditional two-stream CNN model, the randomly selected frame may not contain enough information for the video classification tasks.

The traditional LSTM-based model achieves 69.1% accuracy, which is 23% less than our proposed model. Compared with another traditional approach to video classification, C3D, the accuracy of our method is 10% higher. In addition, we also compare other state-of-the-art methods such as RGB-I3D and TSN. Benefiting from the advanced temporal stream, the proposed model can also achieve higher recognition accuracy than these methods. Besides the Top-1 accuracy, in the spatial stream, our method only uses 25 sampled frames as the input, while all state-of-the-art methods use the standard frame dataset of UCF-101, which has much more frames of each video. Though the input volume becomes smaller, the proposed method still has a better performance.

In summary, the proposed model achieves higher recognition accuracy in both the spatial stream and the temporal stream than the traditional two-stream CNN model. In addition, compared with other state-of-the-art approaches, the proposed model achieves the highest overall Top-1 accuracy.

3.6.2.2 Results on the FIRESENSE Database

Since it is a binary classification problem, the performance of the model is recorded using True-Positive Rate (TPR) and False-Positive Rate (FPR). Equations of these two concepts are shown below. Table 3.5 is a comparison table of abbreviations in the above equations and their full names.

$$TPR = \frac{TP}{TP+FN} \tag{7}$$

$$FPR = \frac{FP}{FP+TN} \tag{8}$$

The performance of the current model is compared with two existing methods that use the same data set: 1) multi-features and rule-based classification [4], and 2) spatiotemporal consistency energy [48].

Table 3.5: Comparison table of abbreviations in Equation (7), (8) and their full names

Abbreviation	Full Name
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative

As shown in Figure 3.19, the proposed model has an average true-positive rate of 96.18% and an average false-positive rate of 0%. Compared with multi-features and rule-based classification and the spatiotemporal consistency energy, the performances of the proposed model are better in both TPR and FPR. In addition, the proposed model can be used for other unforeseeable events identification while the other two can only handle fire detection. As this dataset is insufficient to simulate the actual situation, more datasets are needed for further evaluation.

3.6.2.3 Results on the Merged Dataset

We merge the FIRESENSE data set and the UCF-101 to simulate more complicated situations that contain fire and violent behaviours. In addition, there are some safe situations in the merged data set. In the UCF-101, two violent behaviours, ‘punch’ and ‘nunchucks’, are selected. The whole FIRESENCE database is contained in this experiment, including negative videos. Therefore, the new dataset contains the following labels: ‘punch’, ‘nunchucks’, ‘fire’ and ‘negative’. Since there are only four labels, the Top 5 accuracy is not considered in this experiment. The results are shown in Table 3.6.2.3. Compared with the first experiment in Table 3.2, the result is similar. Single directional LSTM has better accuracy than the bidirectional model. With the help of the stream of optical flow, the final proposed model can improve the accuracy to 96.7%. Since the size of the merged dataset is much smaller than UCF-101, both the Top 1 accuracy and the training speed have been improved in this experiment.

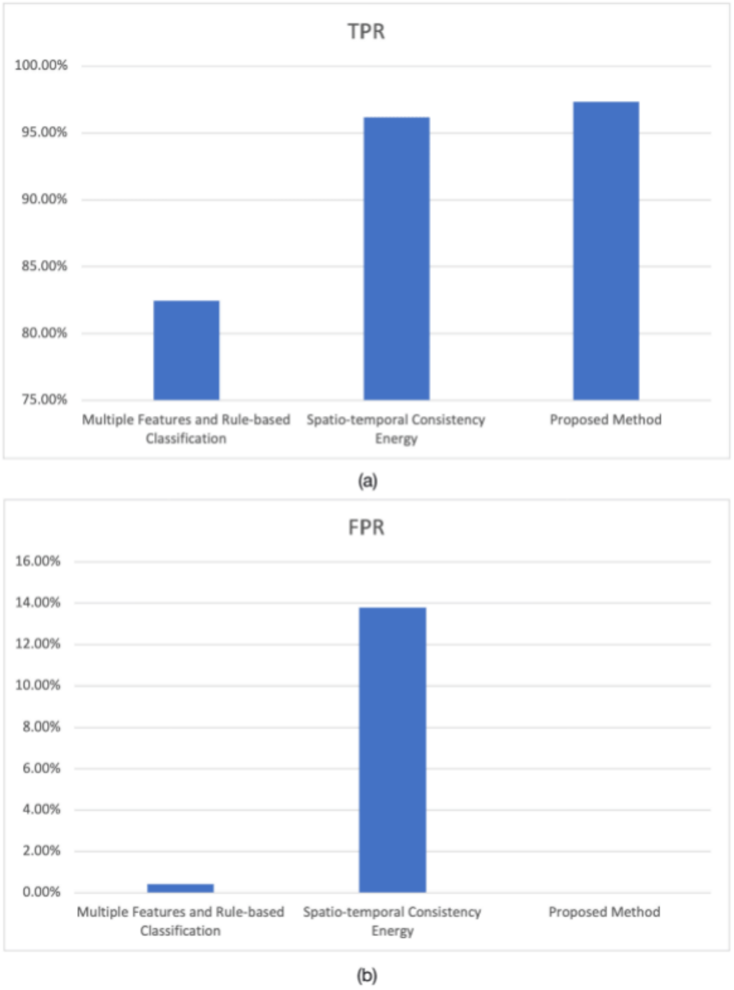


Figure 3.19: Comparison of different methods: (a) TPR (b) FPR

3.7 Discussion

This chapter presents an innovative two-stream model for video human action recognition. The model enhances the function of the spatial stream. According to the results of the experiments, each stream of the proposed model can have a good recognition performance. Finally, the whole model can achieve higher top 1 accuracy than previous deep learning models.

Table 3.6: The Top 1 accuracies and the processing speeds of each step under different methods on the dataset that merge UCF-101 and FIRESENSE database together

Model	Accuracy
VGG16 + LSTM(Bidirectional)	91.7%
VGG16 + LSTM	94.5%
Proposed Model	96.7%

Here are some potential impacts of this chapter:

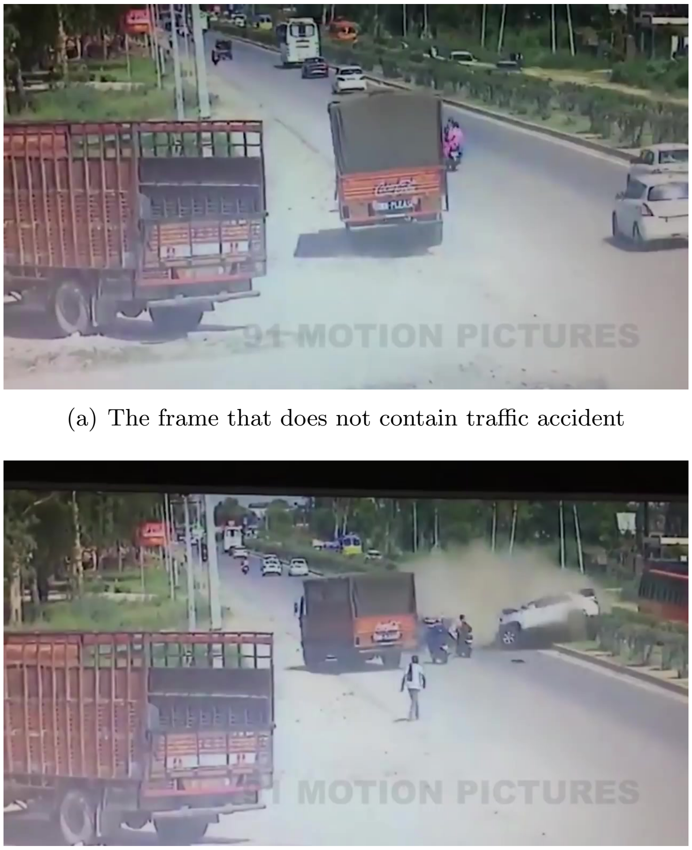
- It provides a new solution for the temporal features extraction problem. It shows that even if the LSTM-based model in the spatial stream is a combination of two basic networks, the two-stream model can still have a high recognition accuracy. In the future, this structure can have further improvements.

- The chapter shows that the optical flow can still be improved if we use advanced CNN.

- The proposed model can be applied in video description tasks with the help of natural language description methods [57].

- The proposed model can be used for smart city surveillance such as the unforeseeable event detection and traffic control [58].

- With the improvement of the first stream, the current model allows a sequence of frames as the input. It solve the potential problem that a random frame may loss valuable features. It is extremely useful for the detection on surveillance videos. As shown in Figure 3.20, two frames are both in the video of a traffic accident. However, (a) does not contain the traffic accident. If the spatial stream of the traditional two-stream CNN model chooses this frame as its input, the output will be the wrong result. However, the proposed model can avoid this situation.



(a) The frame that does not contain traffic accident

(b) The frame that contains traffic accident

Figure 3.20: Two frames in the same video with different information.

Though the proposed method outperforms the state-of-the-art methods, it still has limitations that are needed to be solved improved in the future.

- The proposed model increases the complexity compared with either the LSTM method or the two-stream CNN. The whole model needs more parameters in both streams. Compared with traditional methods, both the LSTM-based model and the DenseNet require more training time. This drawback limits the size of the input, especially for the spatial stream. Currently, the input data for each video are 25 RGB frames. If the training speed was improved, more frames could be added for the recognition. As a result, the model could achieve higher accuracy in experiments.

- The fusion layer is not well developed. We still use the SVM, which is a traditional method in the fusion layer. This part still has room for improvement.

3.8 Summary

We propose an innovative deep learning model, which is used for video classification. The basic structure of this model is the two-stream structure. However, unlike the traditional two-stream CNN model, the proposed method extracts both spatial and temporal features from the RGB video frames in the spatial stream. In order to achieve this objective, we use the LSTM-based model to replace the traditional convolutional neural network in its spatial stream. Furthermore, we implement a DenseNet to improve the performance of the temporal stream. According to the experimental results concerning the traditional two-stream model and other neural networks, the key achievement of our proposed method is to obtain the highest top 1 accuracy among the human action recognition tasks of the UCF-101 dataset. For the FIRESENCE dataset, this model also performs better than traditional fire detection methods. Besides, it can get more than 95% accuracy for the simulated unforeseeable events dataset.

Chapter 4

A Novel Two-stream Structure for Anomaly Detection in Videos

The contents of this chapter have been published in the following papers:

- Zhao, Y., Man, K.L., Smith, J. and Guan, S.U., 2021. A novel two-stream structure for video anomaly detection in smart city management. The Journal of Supercomputing, pp.1-15.

- Zhao, Y., Gabriela, M. and Man, K.L., 2021. Video Anomaly Detection by the Combination of C3D and LSTM. in Proceedings of International Conference on Digital Contents: AICo (AI, IoT and Contents) Technology 2021.

4.1 Introduction

Video anomaly detection is the problem of detecting anomalies in videos. Anomaly refers to some unforeseeable events or emergency that deviates from what is standard, normal or expected. Anomaly detection plays a vital role in smart city management, such as traffic control and criminal investigation. Unlike other anomaly detection tasks that can provide clear unusual signals [12], video anomaly detection requires the analysis of videos. Traditionally, we need professionals to monitor the video constantly to figure out abnormal events. It always turns into a tough and time-consuming task. Therefore, research activities related to this task are of great practical significance since a feasible detection technique can reduce the amount of human resources used to monitor videos, especially for surveillance systems.

Anomaly detection in videos mainly faces the following challenges. Firstly, unusual events always happen with an extremely small probability. It makes relevant datasets difficult to be established. In addition, it causes the situation that the emphasis of related research activities can only be the features of normal videos. It affects the performance of classifiers in models and makes the approach hard to provide correct detection results when the unusual video lies closely to normal video. Another factor exacerbating this phenomenon is that the differences between different anomalies may be huge, making it hard to extract general features from anomalies. Finally, video-based detection tasks are more complex than image-based tasks. Besides the spatial information, such as RGB data and grey-scale histogram, that both videos and images contain, methods used to handle videos should also manage the temporal information. Particularly, anomaly detection techniques are always used to analyse massive video data in the surveillance system. To solve this problem, methods have been proposed and developed over the last decade. Traditional methods [22, 23, 24, 37, 28, 29] focus on using clustering and classification approaches to judge if there is any abnormal event in videos. The kernel is to find anomalies from normal trajectories. Other methods focus on the deep learning-based models [30, 31, 32, 33, 34, 35]. These methods always provide complex models, which are hard to explain and require powerful hardware settings. The models or results produced by traditional approaches have better interpretability. However, their performances are not as good as deep-learning based methods due to the ability of features extraction. The past decade has seen a growth in computer hardware, making deep learning-based models the majority choices among research activities in recent years.

This chapter proposes a model based on the two-stream structure to handle the anomaly detection problem. Plenty of improvements and changes have been introduced to adapt the traditional two-stream model to this specific task. For the structure of the model, the original two-stream model uses a spatial stream to extract spatial features in RGB frames of the video and uses a temporal stream to get temporal features from the optical flow of the video. Therefore, the model may ignore the temporal features of frames and the spatial features of the optical flow. To solve this drawback, we extract both spatial and temporal features from every stream. Considering that the combination of different methods may improve the overall performance of the model for image and video processing [59], the structure combines outputs of Long Short-Term Memory (LSTM) [19, 60], CNN [2], and C3D [5] to replace the 2D convolutional models in the spatial stream. In addition, the structure of the DenseNet is in the stream of optical flow to enhance the connections among convolutional layers. Finally, the fusion layer is also improved to adapt to the new model. Considering the massive video data, we cannot handle them frame by frame directly for the video processing part. It may lead to the problem of computing power. Our approach divides each video into clips and extracts C3D features for every clip to reduce the impact of large data input.

We carry out experiments on the UCF-Crime [61]. It is a large video dataset, which consists of long untrimmed surveillance videos which cover 13 real-world anomalies, including Abuse, Arrest, Arson, Assault, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. It is a popular dataset that is used by plenty of research activities. We use Area under Curve (AUC) as the main evaluation standard.

4.2 Methodology

The proposed method is based on the two-stream structure for video classification [26]. Figure. 4.1 shows the general structure of this new model. It follows the basic structure and inputs of the two-stream CNN model. In Chapter 3, we have proposed a two-stream model which uses an LSTM-based network as its first stream. Although, unlike RNNs, LSTM has three gates to decide the output of every unit, it still faces problems when the sequence of the input data is too long. Because every gate in an LSTM shares the same parameter, the earlier features can still be ignored. The proposed model uses C3D to get both spatial and temporal features from clips for the stream of RGB frames. Then an LSTM is used to get the video-based features. Because inputs of LSTM are feature maps from C3D, the length of input is decreased. The stream of optical flow uses a DenseNet structure to enhance the relationship between different CNN layers. Finally, there is a fusion layer to fuse the outputs of both two streams and get the final detection result. In the proposed model, the input video should be turned into RGB frames and optical flow first. Then the frames will be processed by C3D and LSTM in the first stream. The CNN processes the optical flow in a dense structure. The fusion layer fuses both results from these two streams. It will also output the final detection result. Several new strategies are proposed to improve its performance of detection and adapt the massive input data. The specific modifications include the following aspects.

- Changing the processing part of the video.
- Modifying the feature extraction from 2D CNN to C3D.
- Simplifying the fusion layer to improve the detection speed.

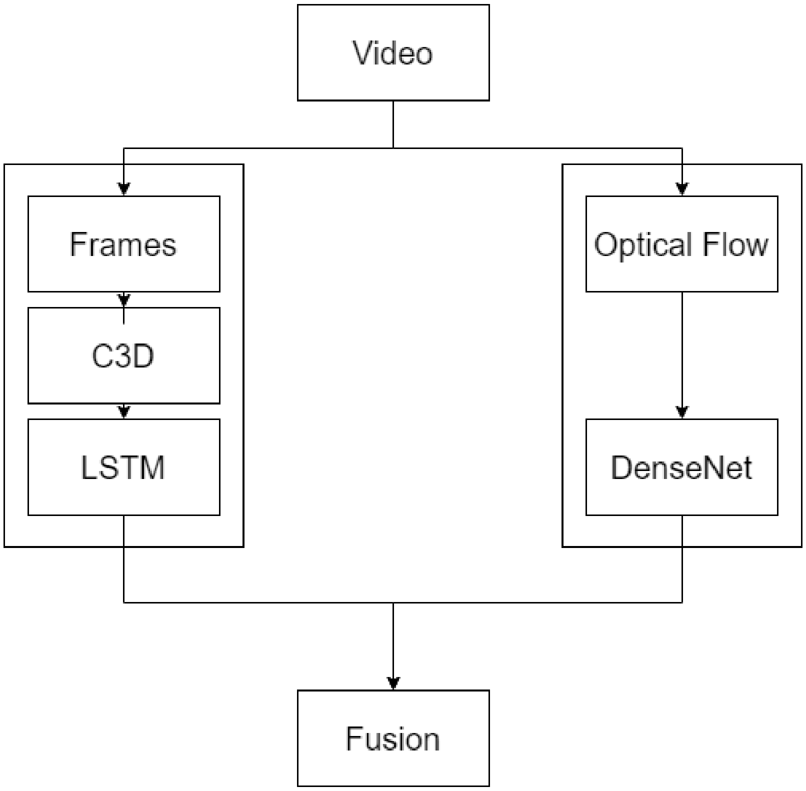


Figure 4.1: The general structure for the proposed method. It shows the two streams, which are designed for the RGB frames and the optical flow, and the fusion part

4.2.1 Clip-based Video Processing

Since video anomaly detection (UCF-Crime) dataset is much larger than the human action video dataset (UCF-101), the video processing part becomes more difficult. In addition, the usage of C3D increases the challenge of the task. Compared with traditional 2D CNN, the features of C3D are more complex. In addition, longer sequences need multiple LSTM layers or more parameters to extract the temporal information. Consequently, more layers should be added, or more parameters should be used in a single LSTM layer. All these factors will affect the time and difficulty of the training epoch. Finally, it may lead to the problem of computer power.

To solve this problem, the number of training samples should be reduced. Besides directly limiting the training samples by setting a specific parameter that would affect the detection performance, doing more pre-processing works for the input video might be a better solution. Therefore, reducing the input data becomes a serious problem. Some processing methods may cause the loss of video information. For example, the method picks one part of the video as the input or sets a skip rate to pick one frame from every several frames. The decision of picking part needs the help of an attention mechanism, which may increase the complexity of the whole model. Otherwise, it may lead to the problem that the picked part of the video is normal, but the other parts contain unusual events. Selecting one frame from several frames will lead to the loss of the input features, especially for the optical flow.

Finally, the video is divided into clips. Similar clip-level feature extraction is used in some other research activities, such as Olympic events scoring [62]. Since the video is cut into several small clips, the computation pressure for the model can be reduced. In addition, because this method does not ignore any frame in the video, the problem of losing features will not occur. In this video anomaly detection task, each video is divided into clips of 64 frames. Then the model extracts C3D features for every clip. For the final clip, the model allows its number of frames to be less than 64. For example, if a video contains 816 frames, we can get 13 clips while the final clip contains only 48 frames. Unlike the clip-level feature extraction before, we consider each clip a complete video and sample every frame in the clip. The general process is shown in Figure 4.2. C3D is used for every clip so that clip-based features can be provided.

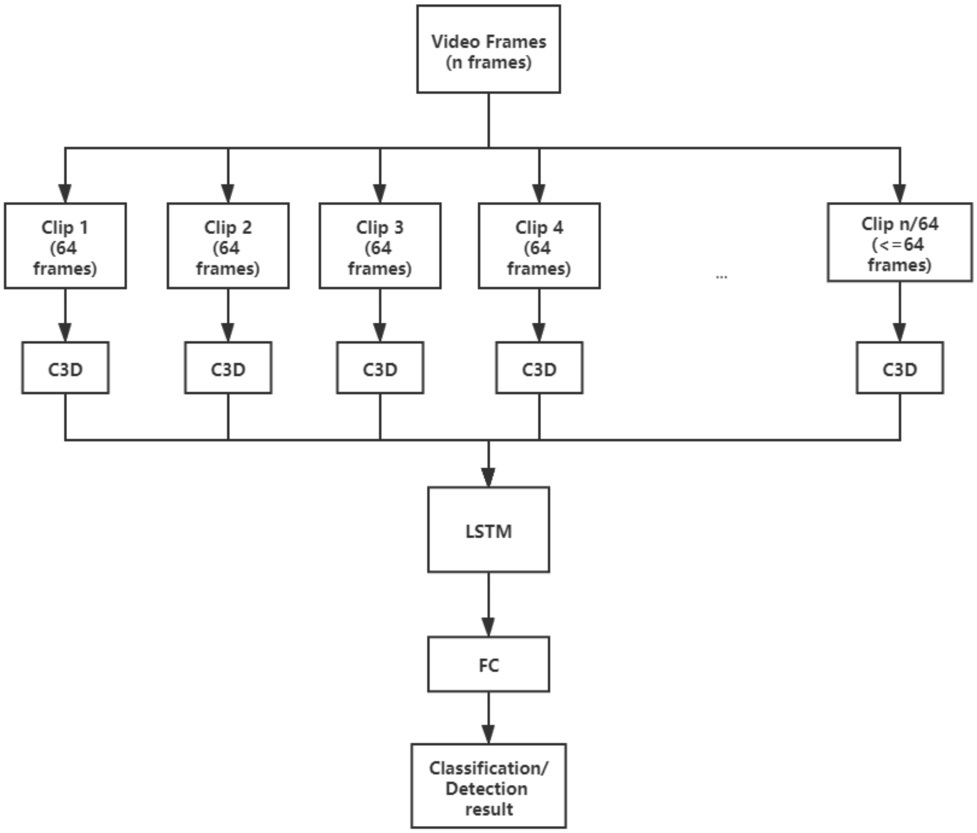


Figure 4.2: The proposed method divides the input video into several clips. except the final clip, every clip contains 64 frames. Each clip is processed by a C3D.

For the stream of optical flow, we choose Horn-Schunck method [63] for the production of the optical flow in the format of vectors. Compared with Lucas-Kanade method [64] and Lucas-Kanade derivative of Gaussian (LKDoG) [65] method, the Horn-Schunck method can keep the balance between capturing enough motion vectors and reducing useless motion vectors. From Figure 4.3, we can see that the Lucas-Kanade method is too sensitive that it records too many irrelevant and wrong features. Too many small motion vectors are recorded by this method. Usually, these vectors describe the vibrations of cameras, which may affect the further feature extractions. The LKDoG method has the opposite situation. It misses much useful information about the subject in the frame. Figure 4.4 shows the part of the optical flow. Then, the sequence of optical images containing the motion of x-channel and y-channel can be generated as the model's input according to the optical flow vectors.

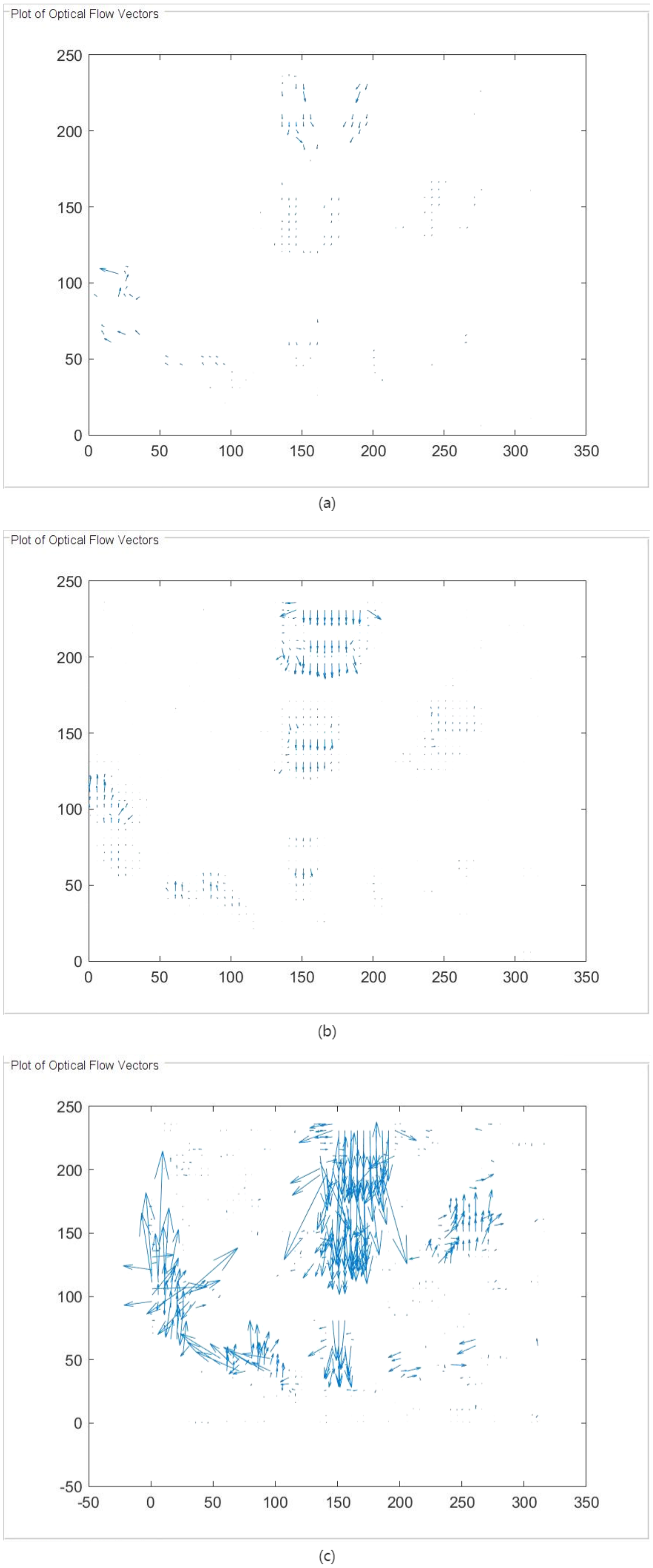
4.2.2 Feature Extraction for the RGB Frames

In this section, the feature extraction part in the first stream is discussed. The combination of a C3D and An LSTM is applied in this stream to ensure the performance of the extraction task.

4.2.2.1 C3D Layer for Clip-based Features

The kernel challenge of video-based feature extraction is to get the spatiotemporal features. 2D ConvNets have been proven to be suitable for spatial features extraction. Furthermore, several approaches have been developed for temporal features. The traditional two-stream model uses an optical flow as one of its inputs so that the temporal information in the optical flow can be captured. For the stream of RGB frames, an RNN-based structure, such as LSTM, can be used to process potential sequential information from CNN features.

To enhance the temporal features extraction, 3D ConvNet is used to the stream of RGB frames. According to previous research, [5], C3D can achieve a better performance than the traditional two-stream structure and LSTM. We do not use the whole network of C3D since it is outdated and not suitable for the video anomaly task. The 3D convolutional kernel is used for the spatiotemporal features extraction work. The basic structure of this part is also shown in Figure 4.5. Six convolutional layers work for feature extraction. Four pooling layers are set after the first, second, fourth and last convolutional layers. In general, the last two convolution layers of the original C3D are deleted to simplify the structure. Each convolutional kernel is set to be 3 x 3 x 3 to achieve the best performance. The whole structure is connected to the LSTM layer and fully connected layers for temporal feature aggression and produce the final output.



(a)

(b)

(c)

Figure 4.3: Outputs of different approaches: (a): LKDoG method, (b): Horn-Schunck method, and (c): Lukas-Kanade method

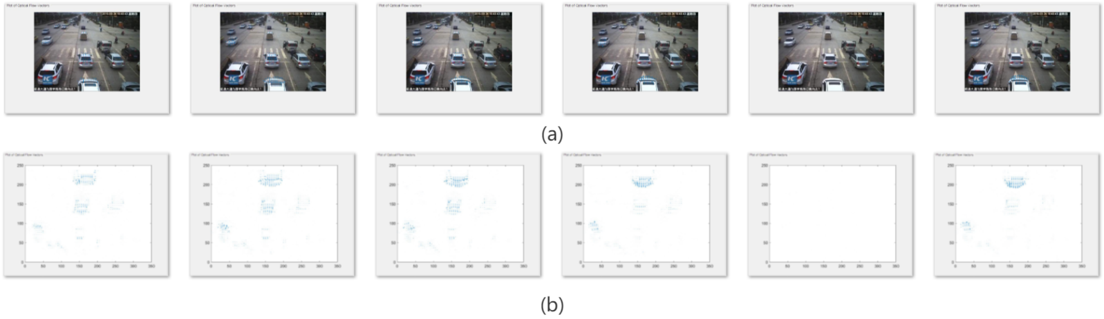


Figure 4.4: (a): Original frames in the video, (b): Corresponding optical flow

4.2.2.2 LSTM for Video-based Features

Typically, FC layers are used to do high-level analysis in the CNN. However, the FC layers cannot effectively extract all the spatial information from the feature map as it simply connects all neurons. According to the structure of C3D, the feature map of C3D is a complex and indifferent structure. Existing research shows that RNN and similar networks have the function of fusing different structures. Therefore, an LSTM is built for this model in a single directional structure [66]. It can filter the input information by three gates, which allow the layer to forget, enhance or output the features. This structure has been evaluated to be effective in video classification tasks [26]. It contains one LSTM layer and two fully connected layers. Each clip provides its clip-based features by C3D. Then the LSTM further processes them into a video-based output. Therefore, it enhances the temporal features. For the streams of the optical flow, all parameters in the C3D layers, LSTM and the FC layer should be updated after every training epoch. Since the original C3D is designed for behaviour classification, keeping its parameters may affect the performance.

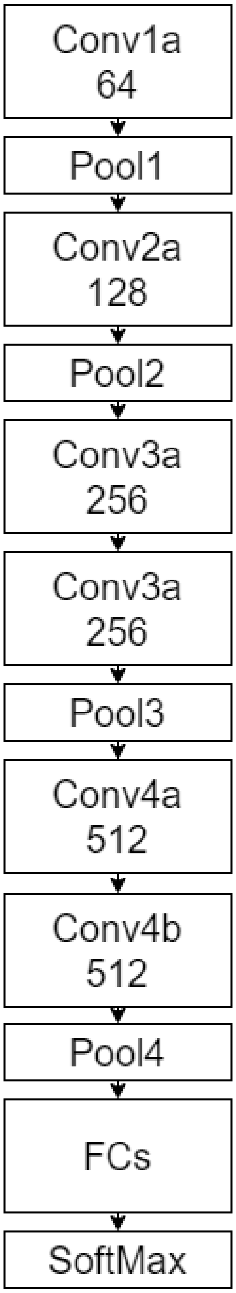


Figure 4.5: The structure of C3D in this method

4.2.3 Feature Extraction for the Optical Flow

In this section, the feature extraction part in the second stream is discussed. The DenseNet is used in the second stream [3]. The structure is this network that enhances the relationship with convolution layers by the dense block. Unlike the traditional CNN networks, layers in each block are related to each other. This structure has several advantages compared with other CNN networks:

- It has fewer parameters.
- It enhances the reuse of features.
- The network is easier to train.
- It alleviates the problems of gradient vanishing and model degradation

4.2.4 Fusion Layer and Detection Stage

Since the streams of RGB frames and optical flows can provide unique results, a fusion layer should be set after the two streams to get the final detection result. The traditional method uses an averaging method or a Supported Vector Machine (SVM) [67], which uses the softmax scores as features to do the feature-based fusion. Due to the limitation of computing power, we use the averaging method in this model to reduce the training time. A softmax layer will process both two streams. Then the fusion layer averages the scores from the softmax layers and gets the detection result according to the averaging score. In addition, the averaging method can reduce the training time due to its simple structure. The detailed procedure is shown in Figure 4.6.

4.3 Experiments

4.3.1 Experimental Setup

In this section, detailed setup of the experiment is introduced, which include the environment and datasets we use in the experiment.

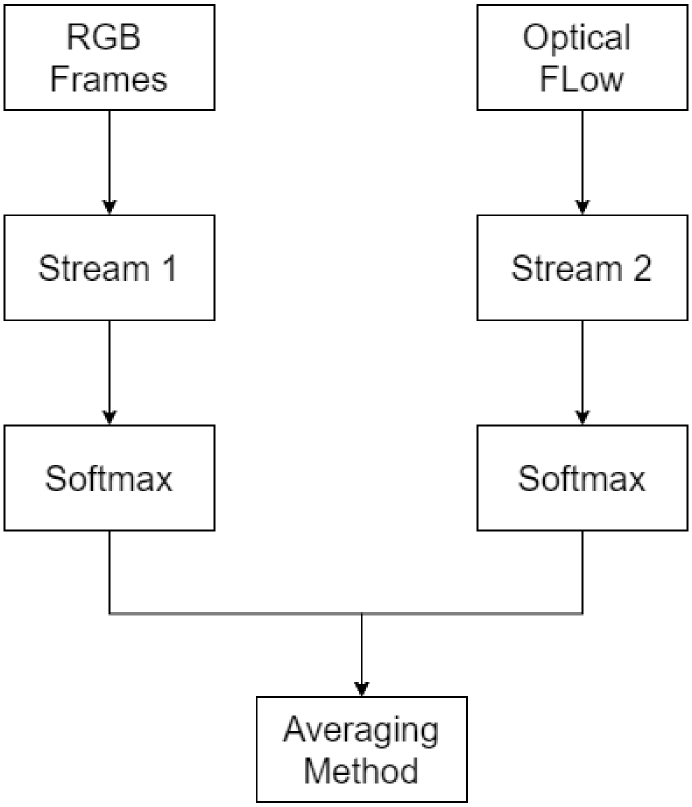


Figure 4.6: The procedure of fusion in the proposed method

4.3.1.1 Experimental Environment

This section shows both the hardware and software environment we use for the experiments. In addition, some details of the experimental settings are also described.

The components of experimental environment are shown in Table 4.1. For the hardware, we use an Nvidia RTX Titan GPU for the experiments. The Turing GPU architecture powers the GPU. It has a 24 GB GDDR6 frame buffer, which can provide the ability for video-related tasks. Python is chosen as the language of the model since it is the most popular programming language in the field of deep learning. For the software, considering the limitation of Keras, PyTorch [68] is chosen to be the platform of the development.

PyCharm completes the whole training process. Totally 30 epochs are trained in this experiment, and we use Wandb to monitor the loss of every epoch.

Table 4.1: Experimental Environment

Component	Name
GPU card	Nvidia RTX Titan
Platform	PyTorch
IDE	PyCharm
Language	python
Monitor	Wanbd

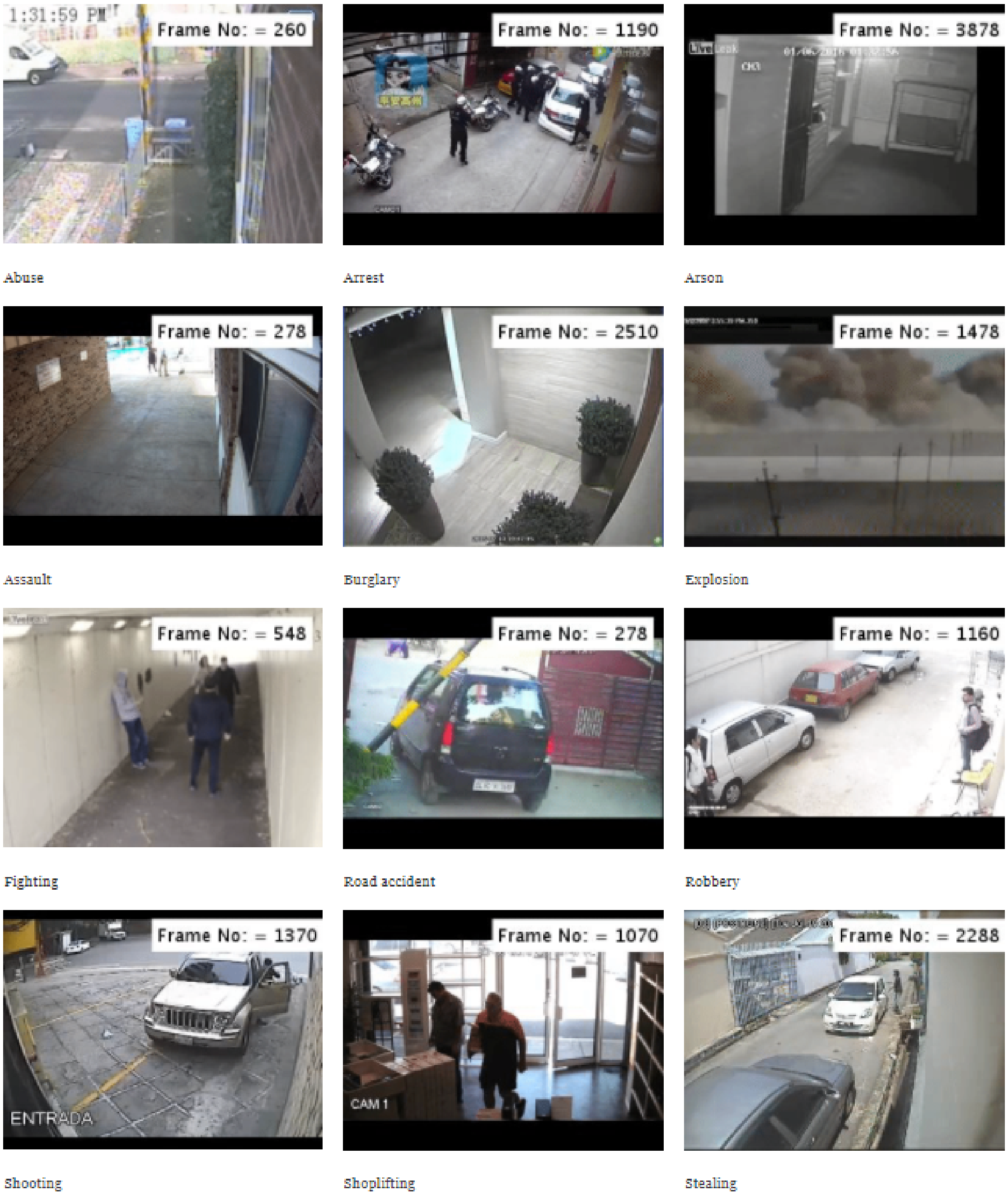
4.3.1.2 Datasets

Experiments are carried out on the UCF-Crime dataset, which is a popular video dataset in the anomaly detection field. It consists of long untrimmed surveillance videos which cover 13 real-world anomalies. Table 4.2 presents some basic information for this dataset. From the table, we can see that it is a large-scale video dataset that contains 1977 videos. UCF-crime is a benchmark dataset in the video anomaly detection field. It could be easy to compare the proposed method with other anomaly detection methods. In addition, since all videos from this dataset are captured from monitoring cameras, it is more closer to the city monitoring system.

Table 4.2: Basic information of UCF-Crime dataset

Properties	Value
Number of videos	1977
Format	mp4
Average frames	7247
Frame rate	30 fps
Total length	128 hours
Labels	Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Road Accidents, Robbery, Shooting, Stealing, Vandalism, Normal

This dataset has some features that affect the efficacy of the training process and the performance of the algorithm. First, the sizes, lengths and frame rates of different videos



Abuse

Arrest

Arson

Assault

Burglary

Explosion

Fighting

Road accident

Robbery

Shooting

Shoplifting

Stealing

Figure 4.7: Frames of videos in the UCF-Crime dataset

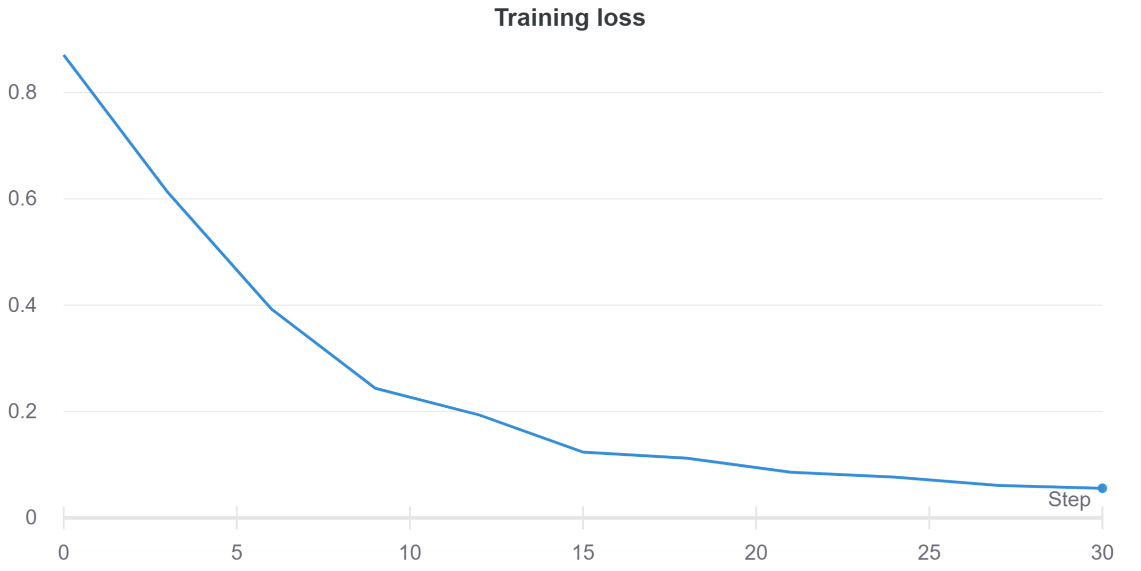


Figure 4.8: Trend of the training loss during 30 epochs. The horizontal axis presents the number of epochs. The vertical axis presents the loss of each epoch

vary a lot. Table 4.2 only shows the average data of these indexes. Second, some videos in the dataset are shot from a fixed view, while others contain clips from different views, which test the robustness of the algorithm. All these points make the detection method more challenging. For the evaluation part, we use the AUC as the standard.

4.3.2 Results and Discussion

In this section, the process of the experiment is analysed and the AUC is compared with other methods.

4.3.2.1 The ROC Curve and the Trend of the Loss

The trend of the loss is shown in Figure 4.8. The loss drops quickly from the first epoch. It is less than 0.1 on the 20th epoch. Then the rate of decline gradually becomes lower and eventually fluctuates. Up to 30 epochs, the loss drops to 0.05, which is a very low value. After the 30th epoch, the value of the loss no longer shows a clear downward trend.

Therefore, we choose the AUC in the 30th epoch as the final results of our experiment.

A Receiver Operating Characteristic curve (ROC curve) [69] is a graphic plot that always is used to evaluate the performance of a model in classification tasks. It uses True Positive Rate (TPR) and False Positive Rate (FPR) as its parameters. The ROC curve of the proposed method is shown in Figure 4.9. From the figure, we can see that the area under the curve is big, which means the performance of the model is good. In detail, according to the data in this figure, we finally get the AUC of our model for the experiments of UCF-Crime, which is 85.18%.

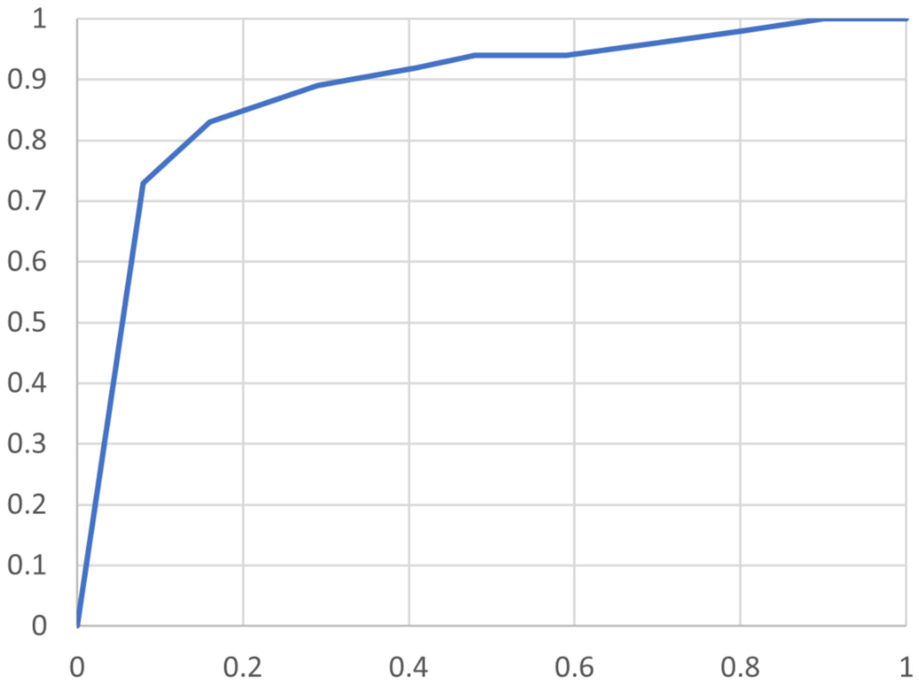


Figure 4.9: ROC curve of the proposed method. The horizontal axis presents the False Positive Rate (FPR). The vertical axis presents the True Positive Rate (TPR)

Table 4.3: Experimental Result

Methods	AUC (%)
Binary classifier	50
CNN (VGG19) + LSTM	80
C3D	81.08
C3D (with 32 frames as an input) [61]	75.14
Inception-V3 [70]	79
Weakly-supervised spatiotemporal anomaly detection [71]	63
Sparse combination learning [72]	65.51
Graph Convolutional Network [73]	82.12
Stream of C3D and LSTM	82.36
Proposed Method	85.18

4.3.2.2 Results Comparisons

We compare our methods with other existing approaches on the UCF-crime dataset. The result of AUC has been presented in Table 4.3. Firstly we choose some benchmarks and classic methods as an object for comparisons. They are C3D, combination of CNN (VGG19) [2] and LSTM, inception-V3 [70]. Our proposed method achieves an AUC of 85.18%, which is higher than above mentioned models. Compared with the traditional C3D structure, whose structure is shown in Figure 2.11, it gets a 4% improvement. In addition, it is 6% higher than the combination of VGG-19 and LSTM. Finally, compared with another classic CNN, the Inception-V3, the proposed model still has a 7% higher performance according to the AUC.

Then we compare our method with other complex methods. From Table 4.3 we can see that the AUC of the proposed method is 20% higher than the method of Sparse combination learning [72]. Compared with the second-highest AUC, which is achieved by Zhong with a Graph Convolutional Network [73], our result is still 3% higher than it. Other approaches in this table achieve AUC values of 63% [71] (Weakly-supervised spatiotemporal anomaly detection) and 75.41% [61] (C3D that uses 32 frames as an input). In conclusion, our proposed method achieves the best performance among all these methods.

4.4 Conclusion and Future Work

This chapter proposes a novel model for video anomaly detection. The model has two streams, which are used to receive RGB frames and optical flows of input videos. For the stream of frames, because the complexity of this stream is increased due to the utilisation of C3D and LSTM, we divide the frames into small clips. Then the C3D can get clip-based spatiotemporal features of the video. Finally, an LSTM is used to enhance the temporal features and produce video-based features and detection results of the whole stream. Regarding the stream of the optical flow, since the input images have temporal information, the model uses traditional 2D CNN for the feature extraction part. In addition, the structure of DenseNet is used in this stream to enhance the relationship between different convolutional layers. Finally, to fuse the outputs of two streams and avoid further increasing the training time, we use the averaging method to get the final detection result. The model is evaluated by the UCF-crime video dataset and gets the highest AUC, which is 85.18%, among different methods. This model also has a good prospect in practical application. By the cooperation of different computer vision methods, it may be applied in different fields. For example, the proposed model can be the supplement of some existing traffic surveillance methods which use the vehicular cameras [74]. For human behaviours, it can provide a primary event recognition result so that methods that use complex behaviours as judgement factors can achieve a better performance [75, 76].

In the future, this new structure can be used in the monitoring system, improving the management of smart cities. The future work will focus on simplifying the model since the current model requires a large amount of computing power. Since the Averaging pooling is only a basic solution for the fusion layer, the spatial fusion [8] can be implemented in the proposed model to improve this part. As shown in Figure 4.10, which introduces two potential examples for the spatial fusion, two streams are fused in the middle of the hidden layers. The left example shows fusion after the fourth convolution layer. Only a single network tower is used from the point of fusion. The right example shows fusion at two layers (after conv5 and after fc8) where both network towers are kept, one as a hybrid spatiotemporal network and one as a purely spatial network.

In addition, The transformer structure [10] may be added into the method since it could reduce parameters for the model and would become popular in the video processing field

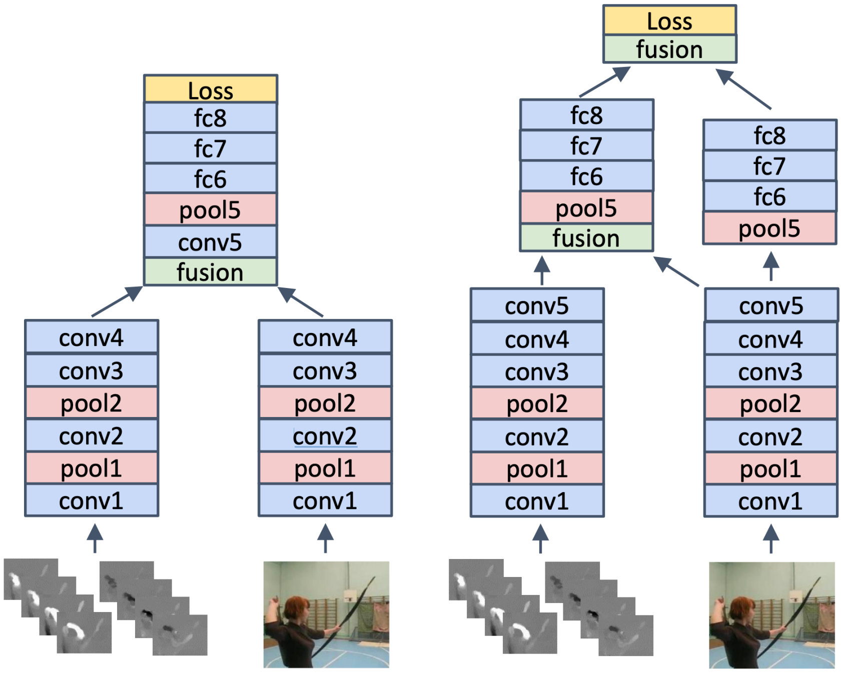


Figure 4.10: Spatial fusion [8]

from this year. The architecture of the transformer has been approved to have better performance in natural language processing field. It would help for both extraction works of spatial and temporal features.

Chapter 5

Applications: Enhanced Public Video Press (EPVP) System

The contents of this chapter have been published in the following papers:

- Han, Z., Zhao, Y. and Man, K.L., 2019. Design and Implementation of the Enhanced Public Video Press for Unforeseeable Events Management. Journal of Industrial Information Technology and Application (JIITA), vol.3 no.4, pp.294-297.

- Zhao, Y. and Man, K.L., 2017. Autonomous Alert Generation and Recommendation for Disaster Management. in Proceedings of RESKO Techinical Conference 2017, pp.100-101.

5.1 Introduction

By applying deep learning methods, it is possible to identify the unusual events in videos, such as fire detection and violent behaviours. However, due to the massive increment of social event needs after lifting the coronavirus pandemic lockdown, the pressure of handling anomalies on government departments, such as hospitals, police and fire stations, is growing with each passing day. In addition, surveillance cameras cannot record every corner of our city. There are still blind areas, such as indoor environments and remote regions, which cannot be monitored.

With the enormous development in smart devices, telecommunication and video processing in the past few years, it is possible to use these techniques to help police, fire stations, and other related practitioners reduce the pressure on dealing with anomalies.

To improve the ability of handling unusual events, this chapter introduces and develops an extended system to enlarge the coverage of the current surveillance system and apply a deep learning method to achieve video anomaly detection. The basic idea is to use mobile phones to record videos. In addition, the deep learning part in the proposed method has been moved to the cloud to reduce the computing press of surveillance systems and smartphones, which both have a limited processing ability.

5.2 Methods

The Enhanced Public Video Press (EPVP) is proposed as a system that allows authorised users to capture unforeseeable events and provides a web portal for related organisations or government departments monitoring the unforeseeable events. It aims to receive videos taken by users' phones and process them by online servers to recognise anomalies using computer vision and big data analytic techniques. The EPVP uses a mobile App to receive and transfer data to the online server via cloud computing.

The overall working process of this system is shown in Figure 5.1. Firstly, the mobile App captures two kinds of data: the video of the events and location information. The video then will be sent to the Video Server. This Video Server uses a deep learning model to identify the category of the event. Then the video with its detection result will be combined with the location information and be shown in the web-based portal.

It consists of three main parts as follows:

- Smart Phone App: to be used by authorised users to send videos.

  To enlarge the coverage of the current city monitoring system, we developed a mobile phone app, which allows users to capture abnormal videos from their phone cameras. The working procedure of the App is presented in Figure 5.2. Firstly, as

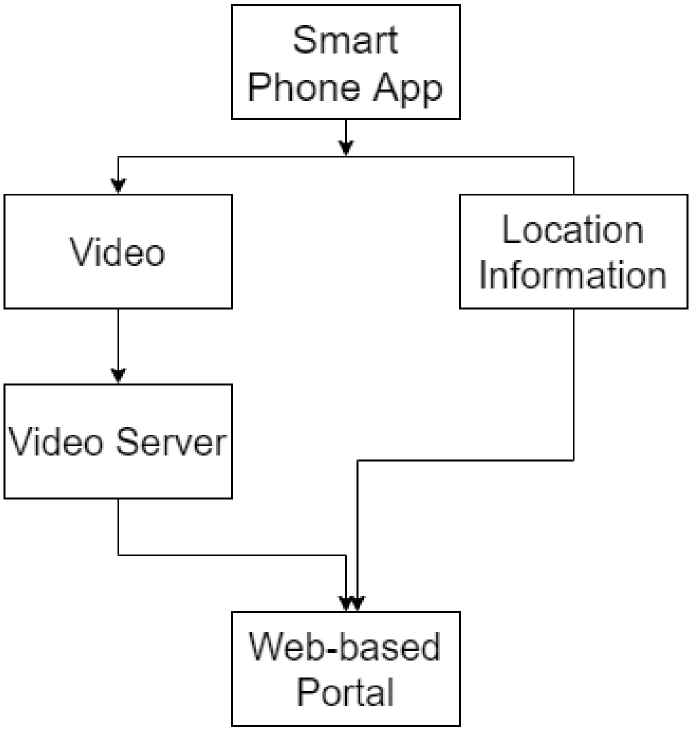


Figure 5.1: The working process of the EPVP system

the world-leading mobile operating system, Android is set as the smart recording platform in this research project [77]. Moreover, based on the requirements from Google, each Android application should define the minimum Android SDK, which is used to identify the lowest Android API level requirement. Moreover, since the Android application in this system has some fundamental functionalities such as getting location, capturing and sending videos, regarding the comprehensive compatibility purpose of this Android application and under the concern of Android fragmentation problems, the minimum Android SDK is set to 21 (Android Lollipop) in this App. Therefore, 96.6% Android devices are capable of running the App [78]. Additionally, because Android is a universal platform with open-source support, smartphones are not limited to Android phones. Any Android devices with Android API 21 and above, including cameras, phones, tablets, watches and televisions, should be able to run this application after some minor compatibility changes.

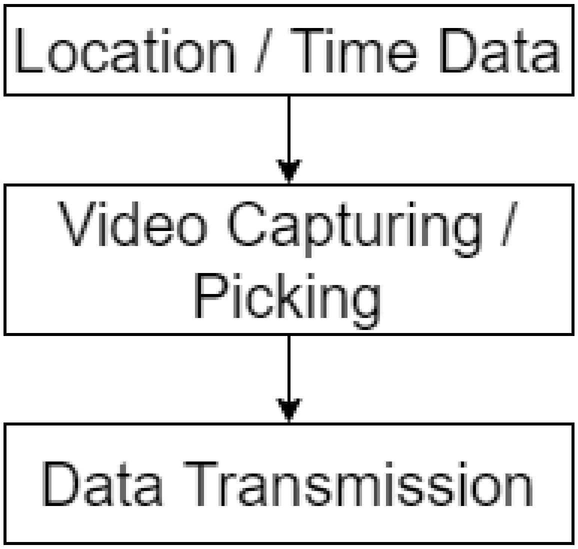


Figure 5.2: General working procedure of the web server

In Figure 5.1, before capturing videos, this application would firstly get the current location from Baidu Map Android SDK, which can provide access to Baidu Map services and data. This system decides to generate a detailed location description from Baidu Map Android SDK rather than GPS coordinates because of the higher usability and readability of the location description.

After the location description is generated, users can choose to record videos or pick videos in this Android application, which corresponds to recording a video via the system default camera and picking an existing video from the Android file picker. The video recorder and the file picker are both implemented by using the built-in system solutions to ensure the compatibility of this App on different devices. Finally, once the video is captured, this application would upload this video and the corresponding information of location and time to the server via HTTP with the pre-specified URL of the server address.

- Video Server: to receive, process and manage uploaded videos.

In this server, a deep learning model is applied to identify the event category and extract useful information from the video, which is the kernel of the project. As the significant component of this system, the server is responsible for multiple functionalities, including video receiving, video sampling and dangers classification, which can be seen in the working procedure of the server in Figure 5.3.

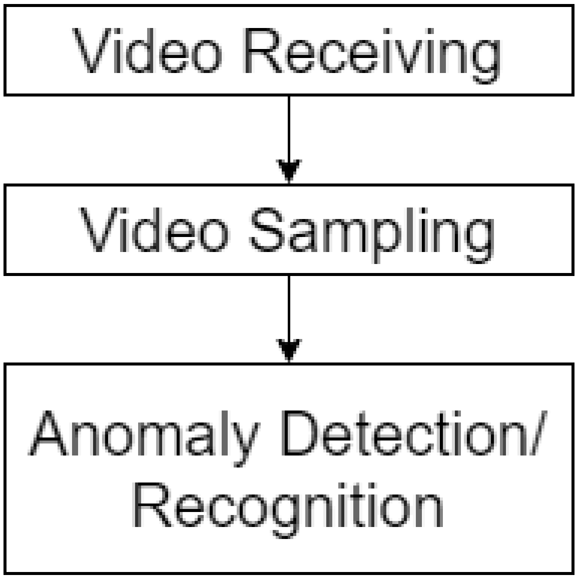


Figure 5.3: General working procedure of the web server

Firstly, the video receiver is designed to receive the uploaded videos from the Android applications and save these videos to the specified location. Moreover, the video receiver is based on PHP, which is a wide-use back-end scripting language [79]. The PHP server part is designed for receiving and saving videos. The detailed information of the uploaded videos such as file name, location and time are updated to the JSON file, which would be multiply accessed by the other components of the system.

After the videos are received and stored correctly, these videos would be processed by the video sampling method, which is based on Python with OpenCV. With the help of the OpenCV package, each video is sampled to images. These sampled images are saved to the specified location and used as input images in the anomaly detection stage.

Due to the limitation of the processing ability, the anomaly detection problem is simplified to a video classification task in this application. There are four kinds of unusual events, including traffic accidents, fire, knife and rifle, which can be seen in Figure 5.4. Videos of the knife and rifle are from the ILSVRC-2014 datasets [2] and others are from datasets in Chapters 3 and 4. 80% of videos are defined as the training set.

In addition, due to the limitation of the computing power, currently models in Chapter 3 and Chapter 4 cannot be implemented in this chapter. As the replacement, a VGG16 is implemented as the classification model. To fit the input requirement of VGG16, all sampled frames are resized to 224*224. After that, they are normalised to the range of [-1,1] so that all frames can have similar data distribution, which could accelerate the convergence during the training process. Additionally, with the help of CUDA acceleration, the number of training epochs is set to be 50, which means the training data will be passed through 50 times. In each training epoch, the verification dataset is used to calculate the current accuracy of this model. Finally, the model is saved on the server so that it can be accessed easily in each round of detection and classification in the future.

To implement the deep learning model of this system, a separate classification script is developed, which is activated after the video sampling procedure when the server receives new videos. The classification script loads the VGG16 model, the sampled frames and the JSON file with the information of each video. Then each sampled image is processed by the model to return the classification result with the highest possibility in the four defined categories: traffic accident, fire, knife and rifle. Moreover, it is also necessary to implement a threshold value checking procedure to avoid the impacts of images classification with no potentially dangerous information. Otherwise, the classification method would always return one of the four defined anomaly types whether there is a potentially unusual event in the sampled images or not. After a series of testing, the threshold value is set to 0.95, which means the classified result would be safe if the highest possibility of the initial classification result is still smaller than 0.95. Finally, the classification result including traffic accident, fire, knife, rifle and safe would be written to the video information JSON file as Boolean parameters.

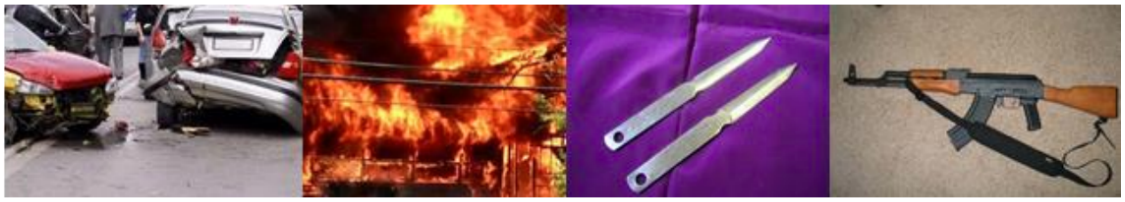


Figure 5.4: Screenshots for different events

- Web-Based Portal: to be used by related organisations and government departments, and enforcement agencies to watch videos and get the result of the video server.

A web portal is developed in this system to provide the uploaded videos with vital information to different authorised audiences on different platforms. Moreover, this web portal is implemented with NAT traverse so that any devices on the public network can access this web portal with a public HTTPS address rather than only the devices in the private network of the bench environment. Target users would be able to watch and download all the uploaded videos with the corresponding important information, including time, location and potentially dangerous classification result.

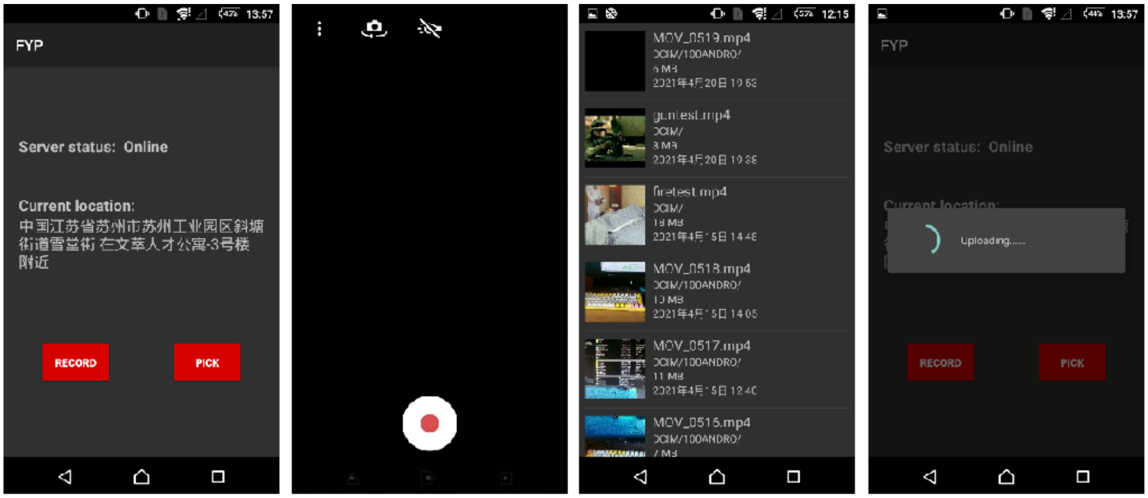
The web portal is based on HTML, JavaScript and CSS with Bootstrap, a free front-end framework, and it runs on a Node.js server. The web portal loads the video files in the specified location and reads the video information JSON file, which is created and modified in video receiving and deep learning classification. The information of each video in the JSON file, including file name, time, location and potentially classification result, would be displayed under the video player.

5.3 Results

A practical App, a simple video server with a database and a website have already been developed. The system can be successfully simulated on the local area network.

5.3.1 App

A screenshot of the App is shown in Figure 5.5. Before starting the video recording, it asks users to confirm the location information. To increase the usability level, this Android application initially tries to Ping the server address and present the current server status to target users. Then the App calls the camera of the phone to capture video. It also provides a button for users to use the file picker to pick an existing video in the album on the phone. Finally, the video is saved in the local album. Due to the limitation of the video processing ability of the phone, the video is also transferred to the video server for further processing. The video is saved in an online file system. In addition, the location data are saved online, which is shown in Figure 5.6. Since currently the system is developed on a computer, the file route is the route in the computer, but not the real route in the online file system.



Server status: Online

Current location:
中国江苏省苏州市苏州工业园区斜塘街道雪堂街 在文萃人才公寓-3号楼附近

MOV_0519.mp4
DCIM/100ANDRO/
6 MB
2021年4月20日 19:53

guntest.mp4
DCIM/
8 MB
2021年4月20日 19:38

firetest.mp4
DCIM/
18 MB
2021年4月15日 14:48

MOV_0518.mp4
DCIM/100ANDRO/
10 MB
2021年4月15日 14:05

MOV_0517.mp4
DCIM/100ANDRO/
11 MB
2021年4月15日 12:40

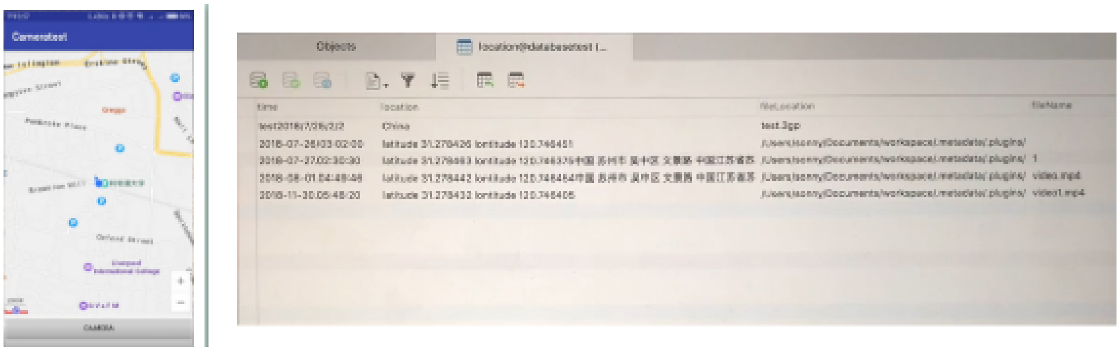
MOV_0516.mp4
DCIM/100ANDRO/
7 MB

Server status: Online

Current location:

Uploading......

Figure 5.5: Screenshots of the App: process of getting videos and location information

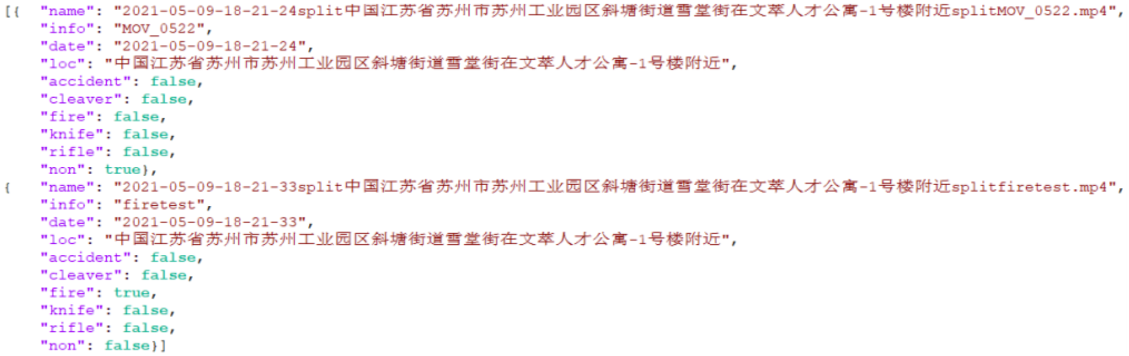


time	location	fileLocation	fileName
test2018/7/26/2/2	China	test.3gp	
2018-07-26/03:02:00	latitude 31.278426 longitude 120.746451	/Users/sonny/Documents/workspace/.metadata/.plugins/	
2018-07-27.02:30:30	latitude 31.278883 longitude 120.746375中国 苏州市 吴中区 文景路 中国江苏省苏	/Users/sonny/Documents/workspace/.metadata/.plugins/	1
2018-08-01.04:48:48	latitude 31.278442 longitude 120.746464中国 苏州市 吴中区 文景路 中国江苏省苏	/Users/sonny/Documents/workspace/.metadata/.plugins/	video.mp4
2018-11-30.05:48:20	latitude 31.278432 longitude 120.746405	/Users/sonny/Documents/workspace/.metadata/.plugins/	video1.mp4

Figure 5.6: The screenshot of the database for location information

5.3.2 Web Server

The web server detects if there is an anomaly in the captured video of the phone. In addition, it provides a detailed category of unusual events. As mentioned in the methodology part, the server saves the uploaded video to the specified location and generate a JSON file to store the information of each video, including filename, location and time, which can be seen in Figure 5.7 below.



```
[{   "name": "2021-05-09-18-21-24split中国江苏省苏州市苏州工业园区斜塘街道雪堂街在文萃人才公寓-1号楼附近splitMOV_0522.mp4",
     "info": "MOV_0522",
     "date": "2021-05-09-18-21-24",
     "loc": "中国江苏省苏州市苏州工业园区斜塘街道雪堂街在文萃人才公寓-1号楼附近",
     "accident": false,
     "cleaver": false,
     "fire": false,
     "knife": false,
     "rifle": false,
     "non": true},
{    "name": "2021-05-09-18-21-33split中国江苏省苏州市苏州工业园区斜塘街道雪堂街在文萃人才公寓-1号楼附近splitfiretest.mp4",
     "info": "firetest",
     "date": "2021-05-09-18-21-33",
     "loc": "中国江苏省苏州市苏州工业园区斜塘街道雪堂街在文萃人才公寓-1号楼附近",
     "accident": false,
     "cleaver": false,
     "fire": true,
     "knife": false,
     "rifle": false,
     "non": false}]
```

Figure 5.7: The screenshot of the json file

Then the video sampling method based on Python with OpenCV package samples the uploaded video into a stack of images, which can be seen in Figure 5.8. However, the sampled image number is limited by the time costs of the potential danger classification method, which would reduce the possibility of correct classification because the target abnormal objects might be missed during the video sampling method.

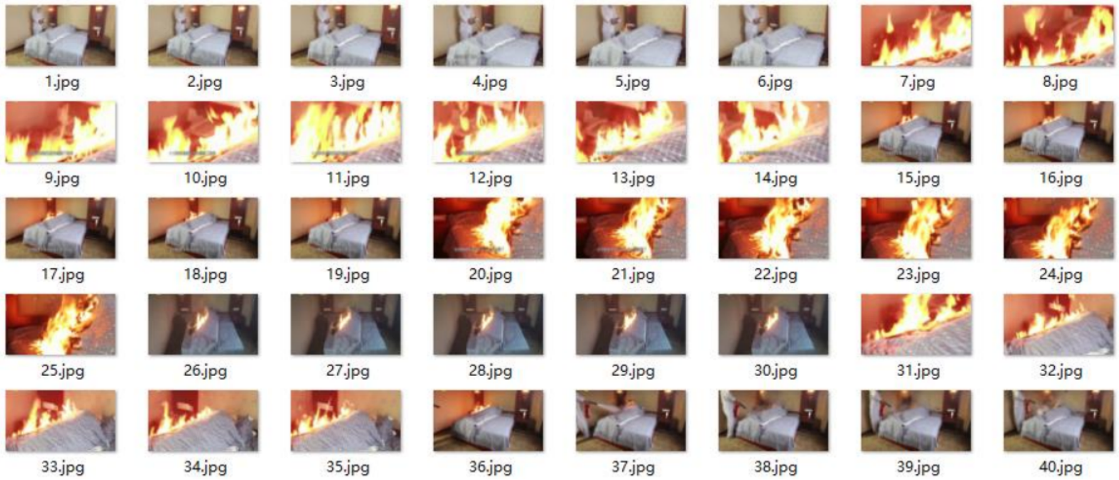


Figure 5.8: The screenshot of the sampled frames

After 50 epochs of training, the accuracy of the model reaches around 0.8. Moreover, as mentioned in the method section, the classification method sets the corresponding boolean parameter to TRUE if the classified possibility is larger than 0.95. Otherwise, the classified result will be safe. Parts of the output of the model is shown in Figure 5.10.

5.3.3 Web Portal

Then the detection result and the video are sent to the website so that authorised users can monitor them from the website. Meanwhile, the location information and the time of the video are also shown on the website. The left side of Figure 5.11 shows the video that does not contain an unusual event on the website, while the right side shows the video that contains an anomaly. The developed web portal in this system can be generally seen as three parts: video player, video information and video list. Target users can use the video list to select videos and watch/download the selected video in the video player. And the file

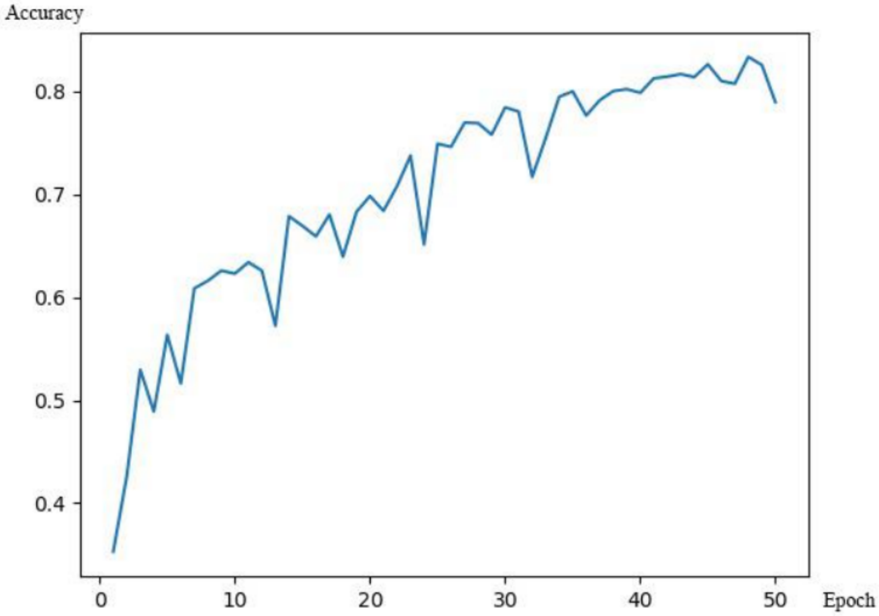
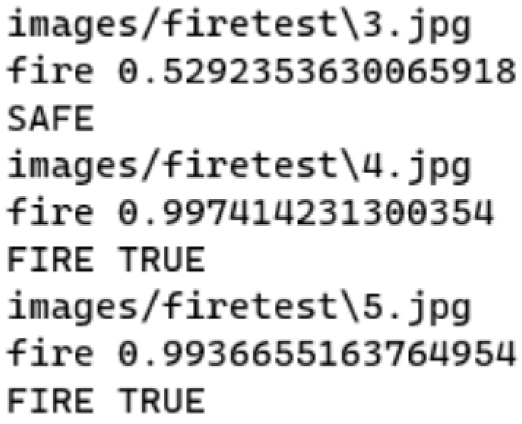


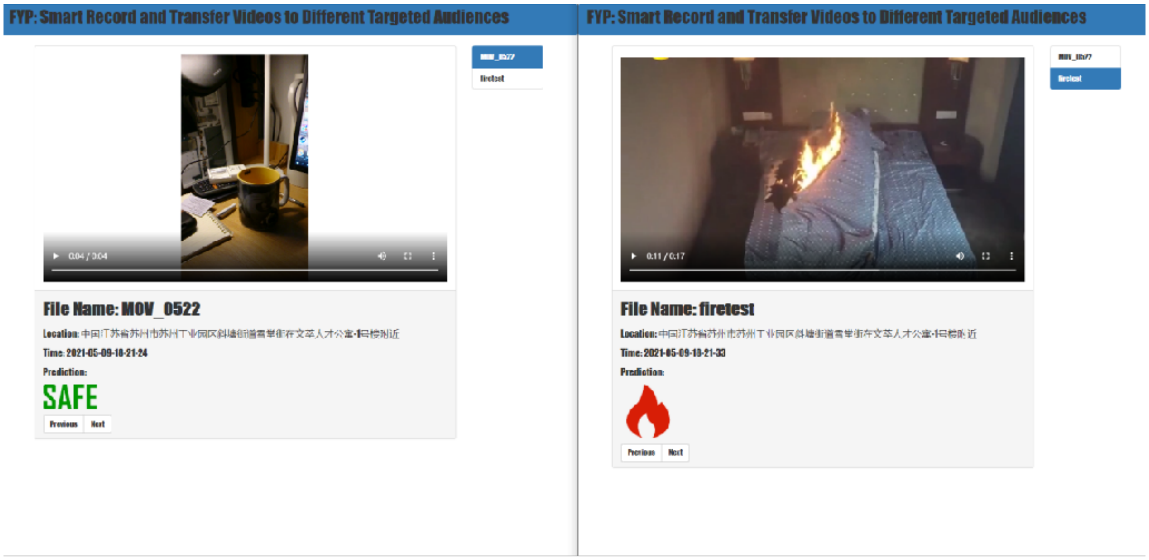
Figure 5.9: Top 1 accuracy of the model after 50 epochs



```
images/firetest\3.jpg
fire 0.5292353630065918
SAFE
images/firetest\4.jpg
fire 0.997414231300354
FIRE TRUE
images/firetest\5.jpg
fire 0.9936655163764954
FIRE TRUE
```

Figure 5.10: Partly output of the model

name, upload location, upload time and danger classification result are listed in the video information part. Moreover, the danger classification result would be represented in a range of icons so that different target users can find the videos they want visibly. Additionally, as Figure 5.11 presents, this system successfully classified non-dangerous information and fire accident information in the safe test video and fire test video.


FYP: Smart Record and Transfer Videos to Different Targeted Audiences

File Name: MOV_0522

Location: 中国江苏省苏州市苏州工业园区斜塘街道雪堂街东文萃人才公寓-1号楼附近

Time: 2021-05-09-10-21-24

Prediction:

SAFE

FYP: Smart Record and Transfer Videos to Different Targeted Audiences

File Name: firetest

Location: 中国江苏省苏州市苏州工业园区斜塘街道雪堂街东文萃人才公寓-1号楼附近

Time: 2021-05-09-10-21-33

Prediction:

Figure 5.11: The screenshot of the website

5.4 Summary

An Enhanced Public Video Press System is introduced in this chapter. The system aims to solve the problem that the monitoring system cannot cover every corner in the city and apply deep learning methods for video anomaly detection. So far, a smartphone App has been developed for users to capture unusual events by their phones. A video server is built to detect and recognise if there is an anomaly in the video. In addition, authorised users can monitor videos and their detection results by a website. Besides the video and its result, some related information such as location and time are also added to the video page.

5.5 Future work

Although the research system is generally developed, it is believed that there are still a lot of room for improvement. The following works are planned to be implemented to this project shortly:

- More kinds of anomalies should be added into the system.

- The current model is too simple and cannot achieve the best detection result. Models in Chapters 3 and 4 should be implemented in the server to improve the better performance.

- The website only contains one main page for all unusual events, more sub-pages which be created for diff rent kind of events.

5.6 Additional Materials

Please find following materials according to links

- The video of fire accident test video:
  https://drive.google.com/file/d/1YiZunKx6k74kBNpc6RehBkVPcLdTsI1g/view?usp=sharing

- The video of the whole experimental procedure:
  https://drive.google.com/file/d/1J1k9CJOJDLxH14EgixDyDimWTnCsZMzq/view?usp=sharing

Chapter 6

Conclusions and future works

6.1 Conclusions

In this thesis, the task of the video anomaly detection and its application have been discussed. Several deep learning methods for videos and video transmission system have been proposed as follows:

- Chapter 3 proposes an innovative deep learning model, which is used for video classification. The basic structure of this model is the two-stream structure. The LSTM-based model is used to replace the traditional convolutional neural network in the spatial stream to extract both spatial and temporal features from the RGB video frames. Furthermore, a DenseNet is applied in the stream of optical flow to improve the performance of the temporal stream. According to the experimental results, with respect to the traditional two-stream model and other neural networks, the key achievement of our proposed method is to obtain the highest top 1 accuracy among the human action recognition tasks of UCF-101 dataset. The model also gets good performance on FIRESENCE dataset and a merged dataset.

- Chapter 4 proposes a novel model for video anomaly detection. The model has two streams, which are used to receive RGB frames and optical flows of input videos. Because the complexity of this stream is increased due to the utilisation of C3D and LSTM, frames are divided into small clips. Then the C3D can get clip-based spatiotemporal features of the video. Finally, an LSTM is used to enhance the temporal features and produce video-based features and detection results of the whole stream. Regarding the stream of the optical flow, since the input images have temporal information, the model uses traditional 2D CNN for the feature extraction part. In addition, the structure of DenseNet is used in this stream to enhance the relationship of different convolutional layers. Finally, to fuse the outputs of two streams and to avoid further increase of the training time, we use the averaging method to get the final detection results. The model is evaluated by the UCF-crime video dataset and achieves the highest AUC, which is 85.18%, among different methods.

- In Chapter 5, an Enhanced Public Video Press (EPVP) is proposed. It contains a smartphone app, a video server and a Web-Based Portal. The EPVP system allows authorized users to capture videos of unusual events by mobile phones and provides a web portal for related organizations or government departments to monitor these anomalies. Furthermore, they can take corresponding actions immediately when there is emergency events happen according to the system.

As a summary, Chapter 3 provides some novel modifications to the traditional two-stream CNN model. For the experiment part, UCF-101 is used for the evaluation of general video classification tasks. FIRESENCE dataset and the merged dataset are used for video anomaly detection tasks. Chapter 4 improves the model in Chapter 3 a lot to enhance the temporal features extraction. In addition, in the experiment section, a larger dataset, UCF-crime, is used to test the performance of anomaly detection. Chapter 5 simulates a potential application of applying the anomaly detection models into applications.

In general, the main contents of this thesis contain the following impacts:

- It proposes two deep-learning models for video anomaly detection tasks. Both are based on the two-stream architecture and can outperform traditional two-stream CNN models and other states of the art methods. These models also have a good prospect in practical application in the future. It may be applied in specific fields such as traffic control and human behaviours monitoring.

- Some innovations in the video processing part have been achieved, which improve the efficiency of the video sampling and training process. These video processing methods can also be used for other methods that need to pre-process or feature extraction of videos.

- An EPVP system is developed to enhance the ability of video recording. In addition, it provides the possibility that different departments and organizations can work together to monitor and handle unusual events in intelligent cities according to the EPVP system in the future, which is represented in Figure 6.1. For example, if there is a traffic accident happened in our city, the city monitoring system or the phones by people around the accident can record videos of this abnormal event. Then videos will be sent to the video server with location and time information. The server can use the anomaly detection models to judge and recognise the anomaly and send all related materials to the web portal. Finally, associated departments and agencies such as hospitals and police stations can take action according to the web portal. All these procedures can be completed automatically and quickly.

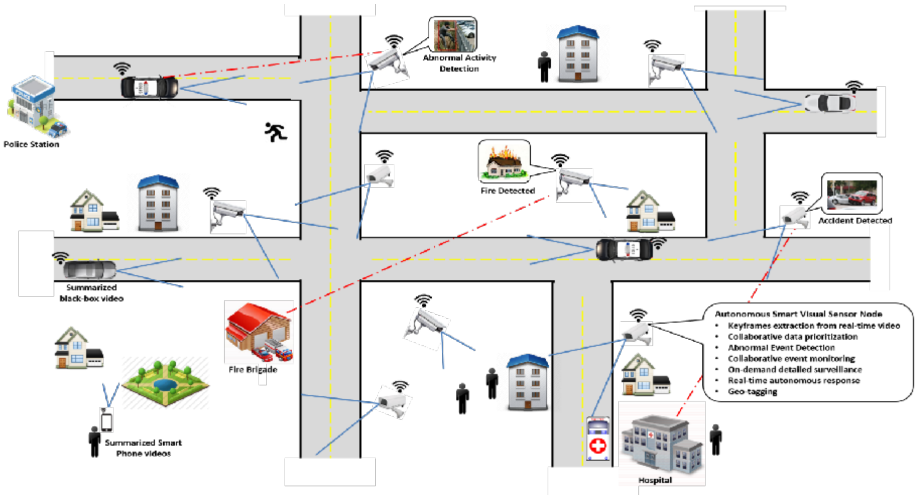

Abnormal Activity Detection
Police Station
Fire Detected
Accident Detected
Summarized black-box video
Fire Brigade
Autonomous Smart Visual Sensor Node
• Keyframes extraction from real-time video
• Collaborative data prioritization
• Abnormal Event Detection
• Collaborative event monitoring
• On-demand detailed surveillance
• Real-time autonomous response
• Geo-tagging
Summarized Smart Phone videos
Hospital

Figure 6.1: A potential use for the future EPVP system

6.2 Future works

Although the proposed model have achieved promising performance with respect to the video anomaly detection problems, improvements can still be done for the future work.

6.2.1 More Datasets

More Datasets are needed to evaluate the performance of proposed models. UCF-Crime is classic and popular for video anomaly detection. There are some other datasets that could be useful for experiments, such as UCSD Pedestrian Dataset [80] and CUHK Avenue dataset [72]. These methods may further evaluate the performance of our proposed model.

6.2.1.1 UCSD

The UCSD Pedestrian dataset was created for the purpose of video anomaly detection in campus. The dataset mainly focuses on crowded scenes. All videos in this dataset are black and white and taken from a single view. In this dataset, anomalous events are either due to non-pedestrian entities in walkways or anomalous pedestrian motion. Some anomalous examples include bicycle riders that cross pedestrian walkways, skaters, cats, and the like. The dataset is split into two subsets, where each subset corresponds to a particular scene. The first scene includes people walking to and from the camera's angle, while the second has people walking parallel to the camera plane. An example of the anomalies can be seen in Figure 6.2.

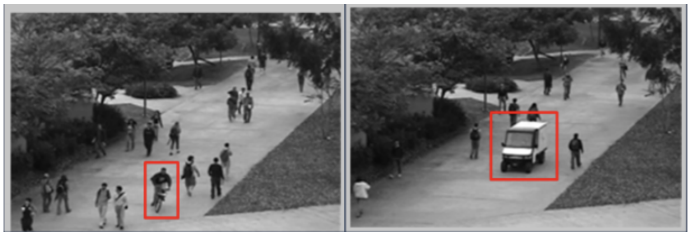


Figure 6.2: USCD Dataset

The detailed information of the subset is shown in Table 6.1. The first subset contains 34 training clips and 36 testing clips, having a resolution of 234 × 159. Meanwhile, the second subset contains 16 clips for training and 12 clips for testing with a resolution of 360 × 240. In addition, a subset of 10 clips for subset 1 and 12 clips for subset 2 are provided with manually generated pixel-level binary masks, which identify the regions containing anomalies. These masks could be helpful for the region recognition research.

Table 6.1: Detailed information of two subsets in UCSD

	UCSD Sub1	UCSD Sub2
Number of videos	70	28
Training samples	34	16
Testing samples	36	12
Total frames	14000	4560
Resolution	238 * 158	360 * 240

6.2.1.2 CUHK Avenue Dataset

CUHK Avenue dataset contains 16 videos for training and 21 videos for testing, including 15,328 training frames and 15,324 testing frames with a resolution of 640 × 360. Furthermore, the dataset contains 47 different anomalies, which include loitering, running, and throwing objects. The detailed information is shown in Table 6.2. Some videos in this dataset are taken with slight camera shakes. In the training set, some normal patterns seldom appear, and a few of outliers are included. All these challenges increase the difficulty of the training process. Screenshots of videos in this dataset are shown in Figure 6.3.

Table 6.2: Detailed information of CUHK Avenue Dataset

	CUHK Avenue
Number of videos	37
Training samples	16
Testing samples	21
Total frames	30652
Resolution	640 * 360

Figure 6.3: CUHK Dataset

6.2.1.3 RWF-2000

RWF-2000 is a video-based database used for scientific research in violence detection [81]. It is mainly collected from common multimedia platforms (YouTube and Youku), and contains 2000 video clips selected from 1000 unique video materials. All videos in this dataset are captured by surveillance cameras. Table 6.3 shows the details of this dataset.

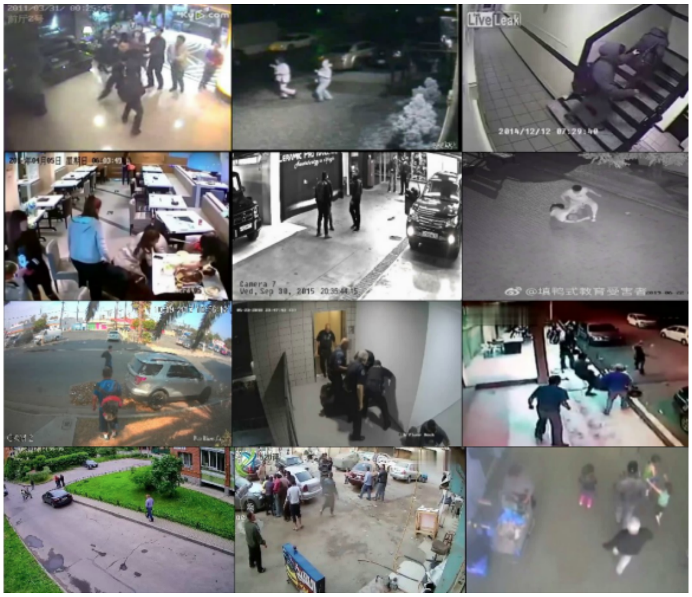


Figure 6.4: RWF-2000 Dataset

Table 6.3: Detailed information of RWF-200 Dataset

	RWF-2000
Number of videos	2000
Training samples	1600
Testing samples	400
Total frames	more than 300000
Resolution	Variable

6.2.2 Transformer Structure

The currently proposed models mainly use RNN-related models for part of temporal feature extractions. However, in Natural Language Processing (NLP), RNN has been proven that cannot have an equal performance compared with the Transformer model. Inspired by the success in the NLP field, it becomes another solution for computer vision tasks. Transformer has strong representation capabilities. Researchers are looking at ways to apply Transformer to computer vision. In a variety of visual benchmarks, transformer-based models perform similar to or better than other types of networks such as convolutional and recurrent networks. Given its high performance and no need for human-defined inductive bias, Transformer is receiving more and more attention from the computer vision community.

6.2.2.1 Transformer for NLP

Transformer is a very classic deep neural network structure in natural language processing. This structure is mainly based on the self-attention mechanism. The basic structure of Transformer is mainly consisted of an encoder module and a decoder module. Several encoders and decoders can be added to these modules according to the design of the model. Each encoder and decoder is composed of several attention layers and a feed-forward neural network, while each decoder also contains an encoder-decoder attention layer. The structure is shown in Figure 6.5. Attention layers are multi-head self-attention mechanisms.

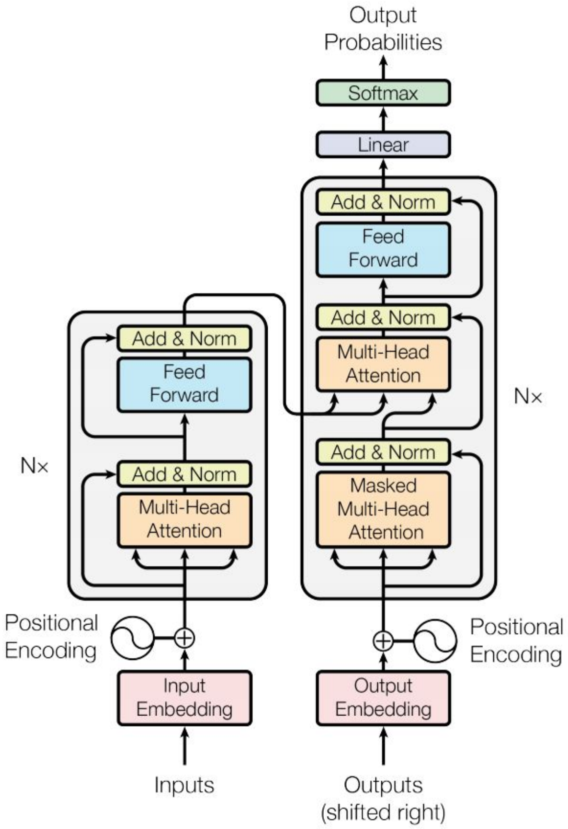


Figure 6.5: The structure of Transformer [9]

6.2.2.2 Transformer for Image Classification

To transfer the model from NLP to computer vision field, image related tasks would be the beginning work since the temporal features of videos can be ignored in image problems. Vision Transformer (ViT) is an excellent example for applying Transformer structure in image classification tasks [10].

The author aims to directly use the most standard Transformer to make minimal changes, divides the image into patches and forms a linear embedding sequence to replace the tokens in the original NLP tasks as inputs for supervised image classification experiments. The basic structure of this model is shown in Figure 6.6. The image is divided into nine patches.

By linearly embedding each patches, the structure simulates the token inputs. Since the image classification task has different target with the NLP problem, after the Transformer architecture, ViT introduces a class token, whose output features can be classified by adding a linear classifier.

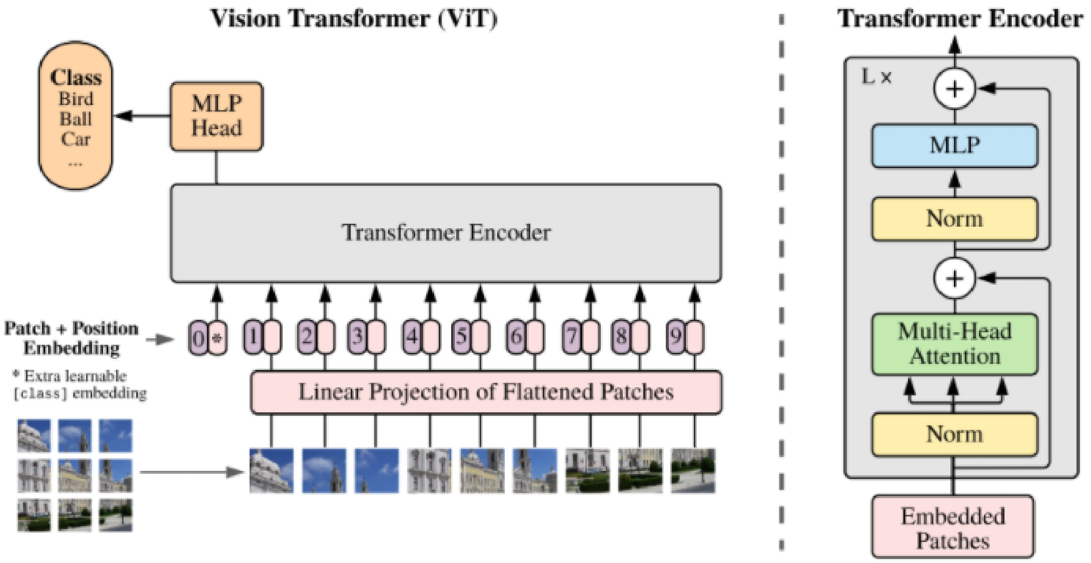


Figure 6.6: The basic structure of ViT model. [10]

The experimental results show that some lacks of inductive biases, such as translation equivalence and locality, makes the ViT cannot have a good performance on datasets with small sizes. However, if the training samples are large enough, ViT can have a better performance than traditional CNNs such as ResNet.

6.2.2.3 Transformer for Video Analytic

Some researchers have already applied the transformer to the video related tasks. TimeSformer [11] is one of the most popular models among them. Researchers compare five different kinds of attention mechanisms to extent the self-attention in Transformer structure from 2D image space to 3D spatiotemporal space. Figure 6.7 presents the working processes of these five methods.

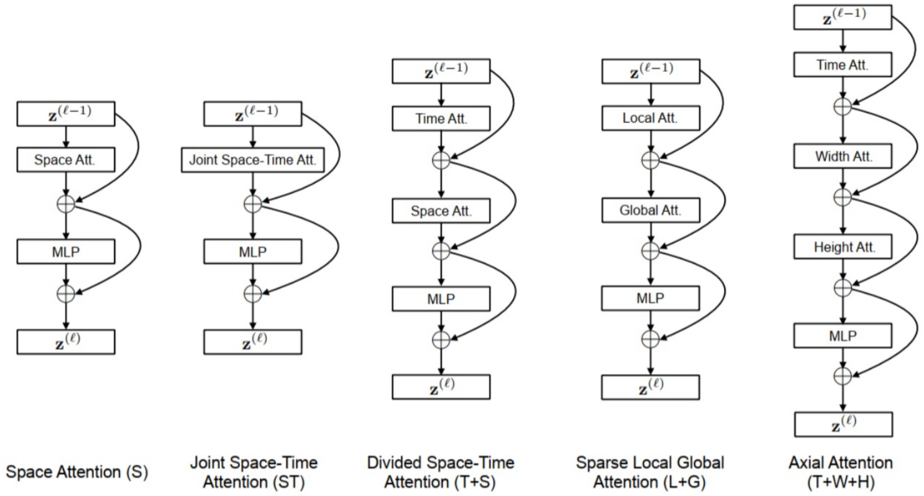


Figure 6.7: Five space-time self-attention mechanisms [11]

The Space Attention (S) only take patches in the same frame for self-attention. In general, it cannot handle temporal information in videos. The Joint Space-Time Attention (ST) take patches in all frames to do the self-attention. It is just like how the 2D CNN is extended to C3D. Thus, this method increases the amount of calculation. The Divided Space-Time Attention (T+S) uses two self-attention mechanisms to handle temporal and spatial information. In temporal attention, each patch is only attached to the patches extracted from the corresponding positions of other frames. In spatial attention, this patch is only attached to other patches of the same frame. The Sparse Local Global Attention only takes half patches of every frame to reduce the computation. Finally, the Axial Attention (T+W+H) uses three self-attention mechanisms. Firstly, the self attention mechanism is carried out in the time dimension, then the self attention mechanism is carried out on the patches with the same ordinate, and it is carried out on the patch with the same abscissa. The detailed comparisons is shown in Figure 6.8. According to the experimental results, the Divided Space-Time Attention achieves the best performance.

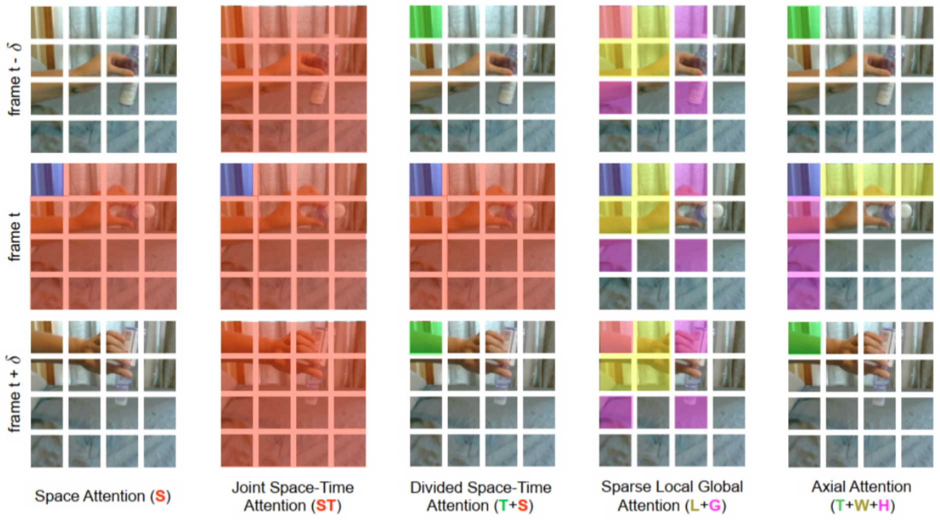

frame t - δ
frame t
frame t + δ
Space Attention (S)
Joint Space-Time Attention (ST)
Divided Space-Time Attention (T+S)
Sparse Local Global Attention (L+G)
Axial Attention (T+W+H)

Figure 6.8: five space-time self-attention mechanisms [11]

6.2.2.4 Summary

Transformer structure may have a good performance if it can be applied in the video anomaly detection tasks. Compared with previous methods such as CNNs and RNNs, the Transformer structure mainly has the following differences:

- The C3D uses 3D convolution kernels to extract features. Since the limitation of the size of kernels, it cannot extract features directly from a long video. However, the self attention mechanism in Transformer architecture allows it to directly capture the spatiotemporal dependence of the whole model's whole video.

- When applied to long videos, the training process of the depth CNN network is very computationally expensive. At present, it is found that Transformer training and derivation are faster than CNN in the field of images. This makes it possible to use the same computing resources to train networks with more vital fitting ability.

- Compared with RNNs, Transformer has a better efficiency on parallel computing. However, both Transformer and RNN are sequenced to sequence models. Modifications are needed when they are used for a sequence to one task, such as detection and classification problems.

6.2.3 Other potential future works

Some other works can be done in the future. Firstly, the method of generating optical flow is the same as the one in the original two-stream model. The process of getting optical flow costs a lot of time, especially when the input video is long. In Chapter 4, the method of integrating C3D into LSTM is directly stacking C3D and LSTM units in a feed-forward way and using the feature maps of C3D as the sequence of inputs of LSTM. According to existing research, these straightforward extensions may not achieve the best performance. C3D and LSTM are two very different structures [82]. A deeper connection between these two methods should be necessary. In addition, although many modifications have been done to simplify the complexity of the model, both models need powerful GPU cards for the training process. Future work can focus on this part to see if any optimization can be achieved. For the fusion layer, proposed models use SVM and Averaging method. These two methods are basic solutions for the fusion layer and may not achieve the best performance. In detail, both SVM and Average pooling can only fuse the classification scores of two streams. They cannot learn the pixel-wise correspondences between spatial and temporal features. It may not be the best way to use these two methods together. The method of spatial fusion may be applied in the future [8]. Finally, the EPVP system still needs further development for real application.

References

[1] AC Bahnsen. Building ai applications using deep learning. *https:,//blog. easysol. net/building-ai-applications*, 2016.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[4] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *arXiv preprint arXiv:1406.2199*, 2014.

[5] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[7] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[11] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[12] A Alfred Raja Melvin, G Jaspher W Kathrine, S Sudhakar Ilango, S Vimal, Seungmin Rho, Neal N Xiong, and Yunyoung Nam. Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud. *Transactions on Emerging Telecommunications Technologies*, page e4287, 2021.

[13] Wei Zhang et al. Shift-invariant pattern recognition neural network and its optical architecture. In *Proceedings of annual conference of the Japan Society of Applied Physics*, 1988.

[14] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[15] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962.

[16] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982.

[17] A Waibel, T Hanazawa, G Hinton, K Shikano, and K Lang. Phoneme recognition using time-delay neural networks (technical report tr-i-0006). *Japan: Advanced Telecommunications Research Institute*, 1987.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Rhys Andrews and George A Boyne. Capacity, leadership, and organizational performance: Testing the black box model of public management. *Public administration review*, 70(3):443–454, 2010.

[21] Jessie James P Suarez and Prospero C Naval Jr. A survey on deep learning techniques for video anomaly detection. *arXiv preprint arXiv:2009.14146*, 2020.

[22] Fan Jiang, Junsong Yuan, Sotirios A Tsaftaris, and Aggelos K Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):323–333, 2011.

[23] Frederick Tung, John S Zelek, and David A Clausi. Goal-based trajectory analysis for unusual behaviour detection in intelligent surveillance. *Image and Vision Computing*, 29(4):230–240, 2011.

[24] Simone Calderara, Uri Heinemann, Andrea Prati, Rita Cucchiara, and Naftali Tishby. Detecting anomalies in people's trajectories using spectral graph analysis. *Computer Vision and Image Understanding*, 115(8):1099–1111, 2011.

[25] Muazzam Maqsood, Maryam Bukhari, Zeeshan Ali, Saira Gillani, Irfan Mehmood, Seungmin Rho, and Young Jung. A residual-learning-based multi-scale parallel-convolutions-assisted efficient cad system for liver tumor detection. *Mathematics*, 9(10):1133, 2021.

[26] Yuxuan Zhao, Ka Lok Man, Jeremy Smith, Kamran Siddique, and Sheng-Uei Guan. Improved two-stream model for human action recognition. *EURASIP Journal on Image and Video Processing*, 2020(1):1–9, 2020.

[27] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

[28] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and Daviv Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *IEEE transactions on pattern analysis and machine intelligence*, 30(3):555–560, 2008.

[29] Siqi Wang, En Zhu, Jianping Yin, and Fatih Porikli. Video anomaly detection and localization by local motion based joint video representation and ocelm. *Neurocomputing*, 277:161–175, 2018.

[30] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1705–1714, 2019.

[31] Davide Abati, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Latent space autoregression for novelty detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 481–490, 2019.

[32] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. In *International symposium on neural networks*, pages 189–196. Springer, 2017.

[33] Joey Tianyi Zhou, Jiawei Du, Hongyuan Zhu, Xi Peng, Yong Liu, and Rick Siow Mong Goh. Anomalynet: An anomaly detection network for video surveillance. *IEEE Transactions on Information Forensics and Security*, 14(10):2537–2550, 2019.

[34] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6536–6545, 2018.

[35] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[36] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[37] Medhini G Narasimhan and Sowmya Kamath. Dynamic video anomaly detection and localization using sparse denoising autoencoders. *Multimedia Tools and Applications*, 77(11):13173–13195, 2018.

[38] Guansong Pang, Cheng Yan, Chunhua Shen, Anton van den Hengel, and Xiao Bai. Self-trained deep ordinal regression for end-to-end video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12173–12182, 2020.

[39] Allison Del Giorno, J Andrew Bagnell, and Martial Hebert. A discriminative framework for anomaly detection in large videos. In *European Conference on Computer Vision*, pages 334–349. Springer, 2016.

[40] Somboon Hongeng, Ram Nevatia, and Francois Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[41] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19(5):1005, 2019.

[42] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso Poggio. A biologically inspired system for action recognition. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. Ieee, 2007.

[43] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[44] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[46] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(3):436–450, 2012.

[47] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[48] Carl Gold and Peter Sollich. Model selection for support vector machine classification. *Neurocomputing*, 55(1-2):221–249, 2003.

[49] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[50] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[51] K Kose, F Tsalakanidou, H Besbes, F Tlili, B Governeur, E Pauwels, et al. Firesense: fire detection and managment through a multi-sensor network for protection of cultural heritage areas from the risk of fire and extreme weather conditions. *Proceedings of the 7th Framework Programme for Research and Technological Development*, 2010.

[52] Xiaoling Xia, Cui Xu, and Bing Nan. Inception-v3 for flower classification. In *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, pages 783–787. IEEE, 2017.

[53] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[54] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[55] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[56] ZeYuan Hu and Eung-Joo Lee. Human motion recognition based on improved 3-dimensional convolutional neural network. In *2019 IEEE International Conference on Computation, Communication and Engineering (ICCCE)*, pages 154–156. IEEE, 2019.

[57] Aniqa Dilawari, Muhammad Usman Ghani Khan, Ammarah Farooq, Zahoor-Ur Rehman, Seungmin Rho, and Irfan Mehmood. Natural language description of video streams using task-specific feature encoding. *IEEE Access*, 6:16639–16645, 2018.

[58] Shaojie Kang, Wen Ji, Seungmin Rho, Varshinee Anu Padigala, and Yiqiang Chen. Cooperative mobile video transmission for traffic surveillance in smart cities. *Computers & Electrical Engineering*, 54:16–25, 2016.

[59] Feng Jiang, ZhiYuan Chen, Amril Nazir, WuZhen Shi, WeiXiang Lim, ShaoHui Liu, and SeungMin Rho. Combining fields of experts (foe) and k-svd methods in pursuing natural image priors. *Journal of Visual Communication and Image Representation*, 78:103142, 2021.

[60] Yuxuan Zhao, Jie Zhang, and Ka Lok Man. Lstm-based model for unforeseeable event detection from video data. In *CICET 2020.*, page 41, 2020.

[61] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018.

[62] Paritosh Parmar and Brendan Tran Morris. Learning to score olympic events. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–28, 2017.

[63] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

[64] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981.

[65] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004.

[66] Sue Han Lee, Chee Seng Chan, and Paolo Remagnino. Multi-organ plant classification based on convolutional and recurrent neural networks. *IEEE Transactions on Image Processing*, 27(9):4287–4301, 2018.

[67] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[68] Nikhil Ketkar. Introduction to pytorch. In *Deep learning with python*, pages 195–208. Springer, 2017.

[69] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[70] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[71] Urvi Gianchandani, Praveen Tirupattur, and Mubarak Shah. Weakly-supervised spatiotemporal anomaly detection. *University of Central Florida Center for Research in Computer Vision REU*, 2019.

[72] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE international conference on computer vision*, pages 2720–2727, 2013.

[73] Jia-Xing Zhong, Nannan Li, Weijie Kong, Shan Liu, Thomas H Li, and Ge Li. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1237–1246, 2019.

[74] M Mazhar Rathore, Anand Paul, Seungmin Rho, Murad Khan, S Vimal, and Syed Attique Shah. Smart traffic control: Identifying driving-violations using fog devices with vehicular cameras in smart cities. *Sustainable Cities and Society*, 71:102986, 2021.

[75] Muhammad Bilal, Muazzam Maqsood, Sadaf Yasmin, Najam Ul Hasan, and Seungmin Rho. A transfer learning-based efficient spatiotemporal human action recognition framework for long and overlapping action classes. *The Journal of Supercomputing*, pages 1–36, 2021.

[76] Maryam Bukhari, Khalid Bashir Bajwa, Saira Gillani, Muazzam Maqsood, Mehr Yahya Durrani, Irfan Mehmood, Hassan Ugail, and Seungmin Rho. An efficient gait recognition method for known and unknown covariate conditions. *IEEE Access*, 9:6465–6477, 2020.

[77] StatCounter Global Stats. Mobile operating system market share worldwide. *Dostopno prek https://gs. statcounter. com/os-market-share/mobile/worldwide*, 2019.

[78] Je-Ho Park, Young Bom Park, and Hyung Kil Ham. Fragmentation problem in android. In *2013 International Conference on Information Science and Applications (ICISA)*, pages 1–2. IEEE, 2013.

[79] Rasmus Lerdorf, Kevin Tatroe, Bob Kaehms, and Ric McGredy. *Programming Php*. " O'Reilly Media, Inc.", 2002.

[80] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1975–1981. IEEE, 2010.

[81] Ming Cheng, Kunjing Cai, and Ming Li. Rwf-2000: An open large scale video database for violence detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4183–4190. IEEE, 2021.

[82] Yunbo Wang, Lu Jiang, Ming-Hsuan Yang, Li-Jia Li, Mingsheng Long, and Li Fei-Fei. Eidetic 3d lstm: A model for video prediction and beyond. In *International conference on learning representations*, 2018.