Title	A study on Anomaly Detection in Surveillance videos
Author(s)	顧, 超逸
Citation	
Issue Date	2020-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/16395
Rights	
Description	Supervisor: 丁 洛榮, 先端科学技術研究科, 修士(情報科学)

Master's Thesis

A study on Anomaly Detection in Surveillance videos

1810061 GU, Chaoyi

Supervisor	Nak-Young Chong
Main Examiner	Nak-Young Chong
Examiners	Kazunori Kotani
	Atsuo Yoshitaka
	Kenta Hongo

Graduate School of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
(Information Science)

February 2020

Abstract

Anomaly detection is a task to detect abnormal and normal actions of people in terms of surveillance videos. Anomaly detection could play an important role in different areas. For example, it could release the problems of lacking the labor force in nursing/daycare facilities since it could detect abnormal actions of the elderly people and/or children to keep their securities and comfort. It could also detect abnormal actions of people in public space to keep public space safe. If abnormal actions are detected, the enforcement agencies will be informed. All of the work can be finished by the anomaly detection system, so lots of human force will be saved. Therefore, the research of anomaly detection is essential, necessary and promising.

The concept of anomaly detection has been proposed in the last century. Due to the limitations of technologies of computer science and sensor, the development of anomaly detection is slow until entering 21 century. Because of the development of science, especially the rapid development of computer science, anomaly detection develops rapidly. More and more attention has been drawn by researchers to the anomaly detection field, and the obtained achievements are remarkable. However, it is inescapable that researchers must define abnormal and normal actions regardless of methods used either traditional or based on machine learning. That is a subjective task as the boundary between abnormal and normal actions are not clearly defined, and it is difficult to define the boundary between them. Another limitation is data labeling. Labeling data is a task that requires lots of human effort, especially when supervised learning is applied to detect abnormal actions. Since the input data consists of image frames, all the frames have to be labeled before training the model. In order to overcome these two main limitations, we propose to apply the Multiple Instance Learning (MIL) for anomaly detection. There are two merits in MIL; Firstly, MIL is a category of weakly supervised learning. The input of MIL is a video-level label instead of a frame-level label. That would save lots of human labor. Second, it is unnecessary for human experts to define the boundary between abnormal and normal actions. Our input for MIL is a video. There is no need to label the start-time and end-time of abnormal actions. Thus, the computer learns to define the boundary between actions by itself.

Moreover, the main contributions of this research are proposing a new model based on a baseline model (deep MIL ranking model) and improving the performance of the baseline model. Before changing the inner structure of the baseline model, we optimized the parameter settings of the Fully Connected Neural Network (FCNN) at the end of the baseline model in order to obtain a better performance. After optimizing the parameter setting of FCNN, we apply Bi-directional Long Short-Term Memory (Bi-LSTM) model between pre-trained C3D model and FCNN in the baseline model to improve the performance further. In order to avoid overfitting, we optimized parameter settings of the Bi-directional LSTM module and provided the best performance in all of the models tested in this thesis.

In order to evaluate performance, ROC, AUC, F1-Measure, and Recall are used. Comparing to the baseline model, experimental results show that our model could improve AUC from 73% to 79%. F1-Measure increases from 9.1%. Recall is improved from 0.55 to 0.665. The main reason leading to this performance improvement is that since the temporal features between adjacent video segments provide valuable information for classification in FCNN, a Bi-LSTM could extract temporal features from adjacent video segments in more efficient than LSTM. Thus, our model performs better.

Keywords: anomaly detections, abnormal actions, Bi-LSTM, FUCNN LSTM, multiple instance learning

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Surveillance cameras are equipment that is used to monitoring behaviors, accidents, activities in order to observe an area. In general, surveillance cameras are connected to recording equipment or an IP network and watched by officials e.g., policemen, security guards, law enforcement officers and related others in order to keep the security of the area being observed. [1]. From the systematic review in [38], meta-analytic techniques were used to poll average effect of Closed Circuit Television (CCTV) on crime across 41 different studies. However, creating a big network of surveillance cameras and recording devices are costly. Therefore, it is usually used by state departments or companies in order to keep public spaces safe. As of 2016, it was reported that 360 million surveillance cameras in the world [39]. It is common to see surveillance in shopping malls, companies, roads, and many other places [1]. But, utilizing surveillance cameras to keep public space safe and secure in real-time is not possible due to lacking enough officers and/or security guards. Thus, building a system or model which could detect abnormal actions or events in terms of using real-time surveillance cameras is an essential and meaningful issue [3]. Indeed, from the 1990s, some researchers started applying machine learning to computer vision to design a model which could detect abnormal or normal actions of human being [27]. Especially in recent years, more and more methods for detecting abnormal and normal actions have been proposed [27]. Seki et al.[4] proposed a method that combining the Self-Organizing Map (SOM) with the Parametric Eigenspace Method (PEM) to detect abnormal actions of the elderly by learning typical actions of the old persons. In the learning stage, SOM was used to extract typical actions (normal actions). In detecting stage, a 2-step eigenspace method is utilized to classify actions. The most important contribution is that the 2-step eigenspace method provides a drastic advantage in compression of image data and calculation of correlation among images. However, there is a limitation of defining the boundary of abnormal actions and normal actions. Even though the Turing test was conducted before defining the boundary of abnormal actions and normal actions, the boundary still may remain subjective. In 2014, Sang-Hyun and Kang proposed a hybrid agent method to detect abnormal behaviors in a crowded scene [5]. The anomalies of humans are divided into two parts: individual anomalies and group interactive anomalies [5]. In the paper, the authors proposed a hybrid-agent method to detect abnormal actions of a group. A merit of this method is that the method includes static and dynamic agents. Static is assigned to a specific spot and analyzes motion information near that spot. The moving object receives a dynamic agent and the motion information will be analyzed by the dynamic agent in terms of following the object's movement [5]. In the final, authors integrate static and dynamic agent information to detect anomalies in a crowd behavior [5]. The disadvantage of this method is that before training the model, all of the samples in the training dataset have to be labeled by a human expert. And it is evident that labeling samples require lots of human resources, time and energy. With the rapid development of science and technology, especially computer science, it is a fact that automatic anomaly detection will play a more and more critical role in protecting the security of public space. Although several methods have been proposed, there are still three severe problems with anomaly detection to be dealt with. Firstly, applying supervised learning to detect abnormal action is not feasible due to the requirement of exhaustive manual data labeling. On the other hand, the performance of unsupervised learning to detect anomaly is rather low. Thirdly, the boundary between abnormal actions and normal actions is difficult to be defined. These three problems were firstly addressed in the deep Multiple Instance Learning (MIL) ranking model proposed by Sultani et al. in 2018 [1, 23]. In the deep MIL model, the learner obtains a series of labeled bags instead of a series of individually labeled instances. Since there is only labeled bags (video-level labels), the model has to determine that star-time and end-time of abnormal actions by itself. This removes the necessity of experts defining the boundary of abnormal actions and normal actions. The MIL model could define/learn that by itself. Besides, the MIL model could not only save lots of human efforts but also improve model performance comparing to unsupervised learning since the instance-level label is unnecessary for the MIL model [23]. However, the Receiver operating characteristic curve (ROC) and Area Under Curve (AUC) measures used for model evaluations is relatively low comparing to their counterparts. Improving performance is indispensable as the output of such models has a direct effect on human lives and security. Therefore, we use the MIL model as our baseline model and the goal of this research is to improve the performance of the baseline model.

In this thesis, a new model based on the MIL model [1, 23] is proposed. Experimental results are presented to show the performance improvement through comparative analysis. The details of the proposed model are given in Chapter 3, and the comparative performance evaluations are presented in Chapter 4. Besides, in Section 5 and 6, not only Accuracy, ROC and AUC, other metrics will also be used to evaluate my model in order to obtain a model with better performance in many aspects.

Chapter 2

Related Works

This chapter is composed of two parts detailing anomaly detection and ranking frameworks.

2.1 Anomaly Detection

Anomaly detection, which helps to keep citizens safe, is an important and challenging issue in the field of computer vision [1, 24, 25, 26, 27, 28, 29]. In 1987, Denning proposed a model to detect abnormal actions [6]. From that, more and more researchers have focused on the issue of anomaly detection. At the beginning of 21 century, with the rapid development of computer science (e.g., the Graphics Processing Units (GPUs)), detecting abnormal actions in surveillance videos has become available.

2.1.1 Anomaly Detection in Supervised Learning

In the work of Weixin et al. [7], they presented a method to detect abnormal actions by comparing the frames of input and reconstructed video. In the first stage, they used Convolutional Network (ConvNet) to extract spatial features of input-image. Then, the features extracted by ConvNet were fed into Convolutional Long Shot-Term Memory (ConvLSTM) in order to extract temporal features of input-frames without excluding spatial information. In final, Deconvolution Layers (Deconv) were employed to reconstruct frames based on spatiotemporal features extracted by ConvNet and ConvLSTM. By computing reconstruction error, abnormal actions could be detected. In the case of anomaly, the reconstruction error would be large. Wang et al. [8] proposed a method combining optical flow and Support Vector Machine (SVM) to detect the abnormal events in video streams. Firstly, the optical flow method called Horn-Schunck (HS) was adopted for combining data terms with spatial terms in order to compute the histograms of optical flow orientation (HOFO) from the original images or the foreground images. Secondly, authors let nonlinear one-class Support Vector Machine (SVM) learn period characterizing normal behaviors and do the anomaly detection in the current frame. The authors designed a detection algorithm that could detect abnormal events faster in terms of combing the optical flow computation with a background subtraction step [8].

2.1.2 Anomaly Detection in Unsupervised Learning

Hirokazu et al. [4] presented a competitive learning method using Self-Organi zing Map (SOM) to learn features of normal actions of the elderly people. Firstly, the SOM was used to extract the features of normal actions. Then, the authors used the Eigenspace Method to classify the actions that SOM has learned. Finally, abnormal actions of the elderly could be detected by using the Parametric Eigenspace Method (PEM).

These suggest that the supervised learning and unsupervised learning are fantastic method in the field of computer vision, especially in the field of anomaly detection. However, there one problems have not been solved. Firstly, in field of anomaly detection, the performance of unsupervised learning is worse than the result of supervised learning. Second, in general, although the performance of supervised is better, labelling the data set will consume lots of human labor. Therefore, using the deep Multiple Instance Learning (MIL) model may solve these two problems at the same time.

2.2 Ranking Framework

The problem of learning how to rank is also a necessary and active issue in machine learning [1]. In this section, some main methods of learning to rank will be introduced. There have been several ranking methods proposed, but most of them paid more attention to make the relative scores of the items better instead of improving the scores of individual scores [1]. A rank-SVM method is be proposed firstly to optimize the retrieval quality of search engines in terms of using clickthrough data in the work of Thorsten et al. [9].

Lately, more and more deep ranking networks are used by researchers in the field of computer vision. And the performances are good. The ranking networks are used to learn features, detect face and other tasks [1]. For example, in the work of Jiang et al. [10]. Learning fine-grained image similarity is a very challenging task since it is necessary of researchers to capture the differences of between-class and within-class image [10]. However, authors proposed a new ranking network which consists of ConvNet layer, Linear Embedding layer, pooling layer and so on in order to learn similarity metric from images directly [10]. But, for our problem, it is necessary that lots of positive samples and negative samples must be provided if we add rank network to our mode. That conflicts to the MIL method, which is a type of weakly supervised learning. Thus, we formulate anomaly detection as a regression problem in the ranking network in terms of using normal and abnormal data. And the features vectors of abnormal and normal actions will be mapped to an abnormal score between 0 and 1 [1].

Chapter 3

Model and Dataset

3.1 Proposed Anomaly Detection Model

The baseline model used in this study is summarized in Fig. 3.1 [1]. This model begins with the anomaly video and normal video which have been divided into non-overlapping 32 segments one by one. After dividing videos into non-overlapping 32 segments, we organize them as positive bag and negative bag. The positive bag consists of 32 video segments of anomaly video. On the contrary, abnormal video segments (32) compose of negative bags.

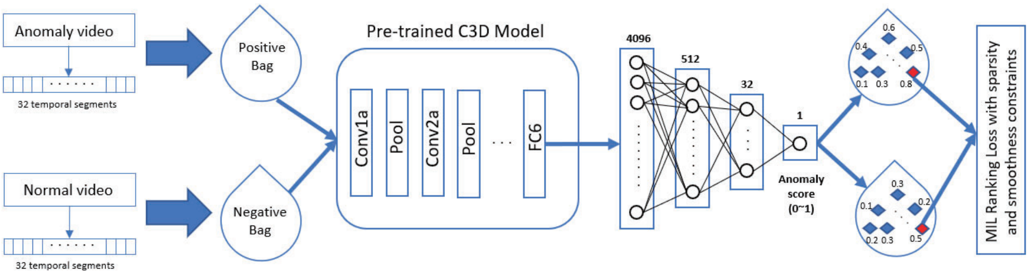


Figure 3.1: A flow diagram of the baseline model [1]

In the second stage, 30 positive bags and 30 negative bags are chosen randomly to compose a mini-batch which will be fed into a pre-trained C3D model to extract the spatiotemporal features. Due to its computational efficiency and outstanding capability of capturing appearance and motion dynamics, we opt to use this feature representation [1]. After extracting spatiotemporal features, we connect (FC) layer FC6 of pre-trained C3D model to Fully Connected Neural Network. Then a fully connected neural network is trained in terms of utilizing proposed deep MIL ranking loss [1, 19].

3.1.1 Multiple Instance Learning

In general, it is common to use the support vector machine to classify the case of supervised learning [1]. And if the positive and negative labels have been provided, the classifier could be learned in terms of using the following optimization function [1]:

$$\min_{\mathbf{w}} \quad \frac{1}{k}\sum_{i=1}^{k}\overbrace{max(0, 1 - y_i(\mathbf{w}.\phi(x) - b))}^{①} + \frac{1}{2}||\mathbf{w}||^2, \tag{3.1}$$

In this equation, b means a basic, $\mathbf{w}$ is the classifier ought to be studied. We utilize $\phi(x)$ to denote feature representations of video segments and images [1]. k is the total number of training examples. And part of ① is used to represent the hinge loss. In the normal supervised learning, accurate annotations of positive and negative samples are essential for a model to learn a robust classifier. Similarly, in a condition of abnormal action detection, if we want to make a model learn features well, the temporal annotations of every video segment are exceedingly necessary. However, it will consume lots of time and human labor that obtaining temporal annotations of every video segments [1].

It is obvious to use MIL model, because of time and human-labor consuming. This is the first reason for users to use MIL model. The second reason is the peculiarities of anomaly videos. Abnormal actions are unpredictable. That means abnormal actions could happen at any time and any place. Besides, comparing to the whole video, the time of abnormal actions is very short. For example, the action of arrest is just almost 20 seconds, arson is almost 10 seconds and so on. This is another character of abnormal video. Taking these two peculiarities of anomaly videos into consideration, the video-level labels of normal videos and abnormal videos are suitable for us [1]. Because video-level label is a type of label that label the whole video instead of labeling all of the frames of the videos. Surely, the video-level label will not consume lots of human-labor and time. Besides, the video-level label is more suitable for anomaly action detection. Because the occurrence time of abnormal action is unknowable.

In this paragraph, the basic fundamental of MIL model is going to be introduced to make it obvious how does MIL model deal with the input and output. A video without any abnormal actions will be labeled as a negative bag. On the contrary, a video containing anomaly in the whole video will be labeled as a positive bag. Then, denoting the positive bag as B_a. And the 32 temporal video segments in the positive bag will be denoted as $(p^1, p^2, p^3, ...p^m)$ and m are used to represent the number of instances in this bag. Similarly, representing a negative video as a negative bag of B_n. And the 32 temporal video segments in the negative bag will be denoted as $(n^1, n^2, n^3, ...n^m)$.

In the positive bag, there is at least one abnormal action. But in the negative bag, there is no abnormal action. This is the most important point in the MIL model. Exact information of the instance in positive and negative bag will not be known because of the video-level label in the deep MIL ranking model [1]. So we optimize the following function in terms of utilizing the instance which obtained the maximum score in positive bag [11]:

$$\min_{\mathbf{w}} \quad \frac{1}{z}\sum_{j=1}^{z} max(0, 1 - Y_{B_j}(\max_{i \in B_j}(\mathbf{w}.\phi(x_i)) - b)) + \frac{1}{2}||\mathbf{w}||^2 \tag{3.2}$$

In this function, Y_{B_j} means the bag-level label and z represents a sum total of bags. The meaning of other variables in this function is the same as in Eq. 3.1 [1].

3.1.2 Deep MIL Ranking Model

It is known that it is difficult for researchers to judge the abnormal actions accurately in the real world in terms of utilizing surveillance cameras [12]. Because there is no standard for them to judge the abnormal actions. This means the researchers have to judge that a type of action is abnormal actions or not by themselves. This kind of judgment is subjective [1]. And the judgment of abnormal actions will vary largely from person to person. This is one of the biggest problems in the field of anomaly detection. In the past, there are three solutions used by researchers, in general, to deal with this problem. Firstly, researchers label some categories of normal actions and let the model learn features from them. All of the actions which are not learned by the model will be recognized as abnormal actions. Secondly, similarly, researchers define some obvious categories of abnormal actions and let the model learn features from abnormal actions. Thirdly, utilizing the unsupervised learning method to detect abnormal actions. It is obvious that the first and the second method cannot solve the problem that avoiding judging abnormal actions in the subject well. In the third method, although labeling the abnormal actions is unnecessary, the performance of unsupervised learning is not satisfactory.

Therefore, we treat the anomaly detection as low likelihood pattern detection instead of classification problem, because of considering the unavailability of sufficient examples of abnormal and normal actions [1, 27, 13, 14, 15, 16, 17, 18].

In our proposed approach, anomaly detection problem is treated as a regression problem [1]. In general, that means the anomaly scores of abnormal action segments will be higher than anomaly scores of normal actions. So the straightforward method is using ranking loss [1]:

$$f(V_a) > f(V_n) \tag{3.3}$$

The video segments of abnormal action could obtain higher anomaly scores, comparing to normal segments. The V_a and V_n represent video segments of abnormal action and normal action [1]. Variables of $f(V_a)$ and $f(V_n)$ mean the anomaly scores of video segments of abnormal action and normal action. Besides, the range of anomaly scores is from 0 to 1. If the segment-level labels could be obtained in the training stage, the ranking function could work well [1].

However, it is impossible to obtain the segment-level labels in the training stage because there are just video-level labels in the MIL. So, using Eq. 3.3 is not possible [1]. The following multiple instance ranking objective function is proposed by us to represent Eq. 3.3.

$$\max_{i \in B_a} f(V_a^i) > \max_{i \in B_n} f(V_n^i) \tag{3.4}$$

In this ranking objective function, the meanings of variables are the same as the variables in Eq. 3.3. The difference between Eq. 3.4 and Eq. 3.3 is that Eq. 3.3 rank each instance of the bag (positive and negative), but Eq. 3.4 does not. In Eq. 3.4, we only rank on the two instances which have the highest anomaly score respectively in a positive bag and a negative bag [1]. The left part of Eq. 3.4 means the video segment which obtains the highest anomaly score in a positive bag. The right part means the video segment which obtains the highest anomaly score in a negative bag. Treating the video segment which obtains the highest anomaly score in the positive bag as true positive instance [1]. At least, the possibility that there are abnormal actions in this video segment is very high. The video segment which obtains the highest anomaly score in a negative bag is denoted as a hard instance. Because this type of segment is the one that looks most similar to an anomalous segments but is a normal instance. And the hard instance causes a false alarm easily in abnormal action detection [1].

The goal of using Eq. 3.4 is to keep positive instances away from negative instances. Our ranking loss in the hinge-loss formulation is [1]:

$$l(B_a, B_n) = max(0, 1 - \max_{i \in B_a} f(V_a^i) + \max_{i \in B_n} f(V_n^i)) \tag{3.5}$$

However, there is a demerit in this formulation that the underlying temporal structure of videos include abnormal actions does not be taken into consideration [1].

In order to improve the performance of the model, the formulation is modified. Firstly, adding ② to Eq. 3.5. ② represents a sparsity term. The occurrence time of abnormal action is short. That means the scores of instances(segments) in a positive bag should be sparse and only a few segments may include the abnormal action [1]. Secondly, adding ① to Eq. 3.5. And ① is the temporal smoothness term. Video is a type of sequential segments. That means in this case, the score of anomaly ought to change smoothly between video segments [1]. So, the score of anomaly changes smoothly between video segments could be ensured in terms of minimizing the difference of scores for adjacent video segments. And the new loss function is given as follows:

$$\begin{aligned} l(B_a, B_n) = max(0, 1 - \max_{i \in B_a} f(V_a^i) + \max_{i \in B_n} f(V_n^i)) \\ + \overbrace{\lambda_1 \sum_{i}^{(n-1)} (f(V_a^i) - f(V_a^{i+1}))^2}^{\textcircled{1}} + \overbrace{\lambda_2 \sum_{i}^{n} f(V_a^i)}^{\textcircled{2}}, \end{aligned} \tag{3.6}$$

In Eq. 3.6, error is back-propagated from the maximum scored video segments in both positive and negative bags [1]. It is possible to obtain a network that could predict high scores for video segments include abnormal actions.

In the final, it is necessary to add model weights to the loss function. So the final formulation is given as follows [1]:

$$L(w) = l(B_a, B_n) + \lambda_3 ||w||_F, \tag{3.7}$$

In this equation, w denotes model weights.

3.2 Model Architecture

In this section, the introduction of final model and process of changing the model form the baseline model to final model have been given. In Section 3.1, introducing lots of details of the baseline model. However, we found there are 2 limitations in the baseline model. Firstly, the parameters settings in the Fully Connected Neural Network could not supply the best performance. Secondly, the temporal features extracted by the C3D model do not be made full use of.

Therefore, our model can be divided into 2 stages. In the first stage, optimizing the parameters of the Fully Connected Neural Network (FCNN). This stage will be introduced in Section 3.2.1. In the second stage, inserting

Bi-directional LSTM module to the baseline model to obtain the final model. However, before inserting Bi-directional LSTM module to the baseline model, we tried to insert LSTM module to baseline model first. This process also makes a contribution to obtaining the finial model. So this process is also necessary to be introduced. The details of a model with Bi-directional LSTM module (final model) or LSTM module will be introduced in Section 3.2.2 and 3.2.3. Surely, the output and input will be introduced again, if they change. Besides, all of the experiment results will be given in Chapter 4 instead of in this chapter. Every model has been marked by the name which are corresponding to inner structure of itself. All of names and introductions of inner structures will be given in Section 4.2 and Fig. 4.1.

3.2.1 Parameter Optimization of Fully Connected Neural Network

To obtain the best performance without changing other parts of the baseline model, we only optimized the parameters in FCNN. In the right part of Fig. 3.2, the red numbers are the optimized parameters. After doing 6 experiments, the parameter settings in the right of Fig. 3.2 have been found which could give the best performance if the inner structure of the baseline model does not be changed. The experiment result will be given in Chapter4.

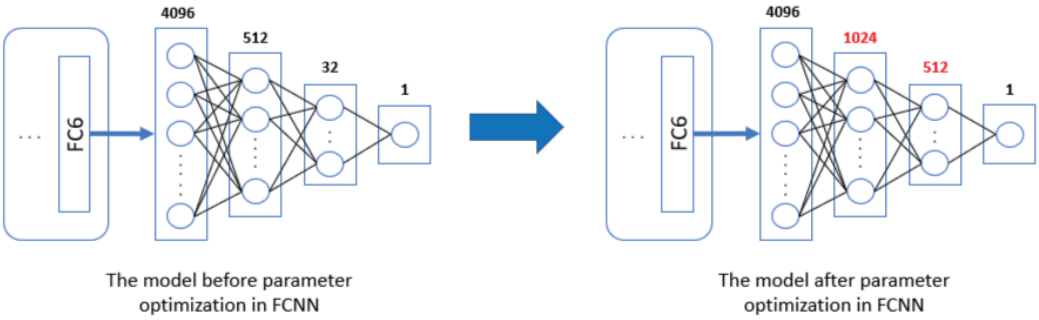

4096
512
32
1
FC6
...
The model before parameter optimization in FCNN
4096
1024
512
1
FC6
...
The model after parameter optimization in FCNN

Figure 3.2: Left part is the baseline model. The right part is the baseline model after optimizing parameters in Fully Connected Neural Network [1]

After finding the best parameter setting in Fully Connected Neural Network, we will start changing the inner structure of the model without changing the parameter settings. And we use ”ParameterA” to represent parameter settings which have been optimize. ”Baseline Model” denotes the model without optimizing parameter settings.

3.2.2 Combing Long Short-term Memory layer to Fully Connected Neural Network

In this and next section, the process of changing the model from baseline model to final model will be introduced. And changing the inner structure of the baseline model to get better performance.

From the baseline model, the 3D convolution features(spatiotemporal features) which are extracted by C3D model are fed into a fully connected neural network directly to obtain anomalous scores. However, there is a limitation that the temporal features extracted by C3D model don't be made most of.

The input of baseline model is 32 non-overlapping temporal video segments of the positive bag or negative bag [1]. In training stage, the positive bag or negative bag is fed into C3D model directly one by one in order to extract spatiotemporal features of 32 non-overlapping temporal video segments. The normal actions or abnormal actions in the video are divided into one or some video segments which are included in 32 non-overlapping video segments [1]. And it is obvious that even though normal actions or abnormal actions in video are divided into some segments, there are some temporal relationships between adjacent video segments. If abnormal or normal actions are divided into one video segment because the occurrence time is shot, there are also temporal relationships between adjacent video segments. However, the spatiotemporal features of 32 non-overlapping video segments are directly fed into a Fully Connected Neural Network to do classifications in the baseline model [1]. So it is necessary to insert LSTM module between C3D model and FCNN to extract temporal features between adjacent video segments in order to improve the performance of the whole model.

In the following article in this section, the inner structure of LSTM will be introduced firstly. Then, introducing the whole model with LSTM module. Especially, explaining how does LSTM connects to the C3D model and FCNN.

In the Fig. 3.3, the inner structure has been given. In Fig.3.3 (a), the function of red line is to send the information from C_{t-1} to C_t. There are two gate on the red line in Fig. 3.3 (a), the left and right one are denoted as ”Forget Gate” and ”Input Gate”. The whole blue area is a cell which could memory the information. These two gates are used to decide the information comes from C_{t-1} could be sent to C_t or not. The layer (red) which connects to ”Forget Gate” directly is ”Forget Gate Layer” in Fig. 3.3 (b). The inputs of ”Forget Gate Layer” are h_{t-1} and X_t. The output is a vector and the range of every element of this vector is between 0 and 1. 0 means the information of C_{t-1} is denied. 1 means the information could pass ”Forget Gate Layer” and be sent to C_t. In simple, the output of ”Forget Gate Layer” is a vector

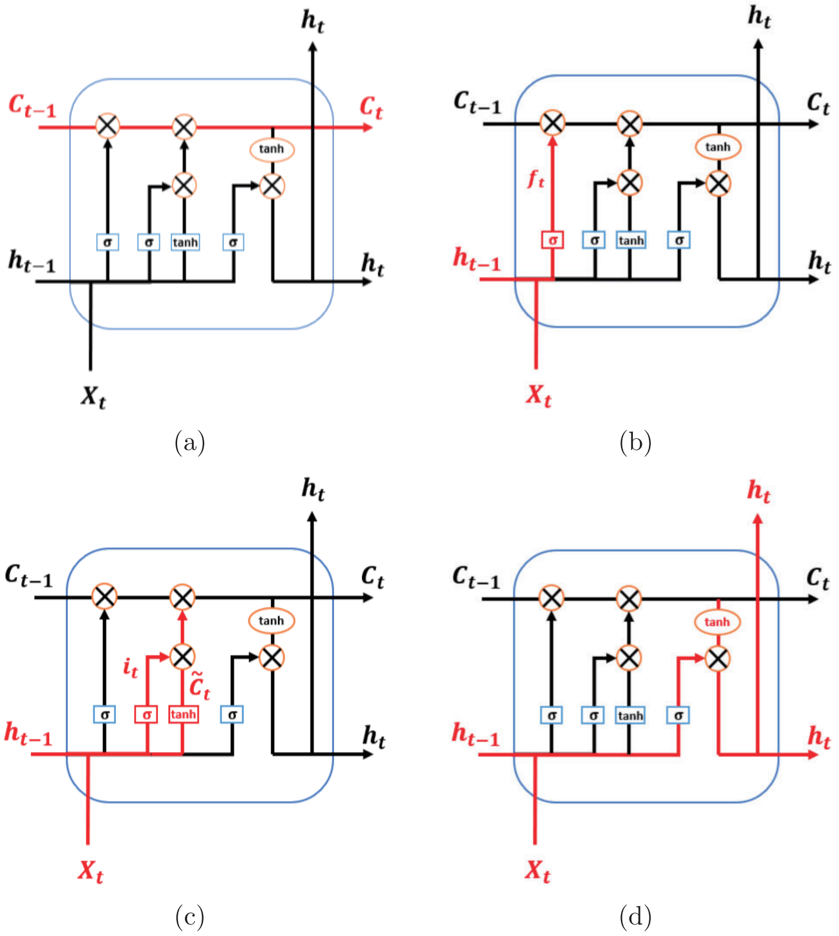

(a) (b) (c) (d)

Figure 3.3: Inner structure of LSTM [33, 35]

which could decide how much information in C_{t-1} will be remained. The layer (red) which connects to "Input Gate" directly is "Input Gate Layer" in Fig. 3.3 (c). The inputs of this layer are h_{t-1} and X_t. Output is also a vector which could decide how much information of h_{t-1} and X_t could pass "Input Gate" then, be sent into the cell. In Fig. 3.3 (d), the "Output Gate Layer" connects to "Output Gate". The inputs are h_{t-1} and X_t. Output is h_t. The function of "Output Gate Layer" is to decide how much information will be output. In general, the inputs of LSTM are h_{t-1} and X_t. Outputs is h_t.

Fig. 3.4 is a LSTM network. X_{t-1}, X_t, X_{t+1} represent inputs and h_{t-1}, h_t, h_{t+1} are the outputs of LSTM network. From Fig. 3.4, it is easy to find that LSTM network could extract the temporal features of input in efficient. Because according to Fig. 3.3, function of forward layer is to decide what

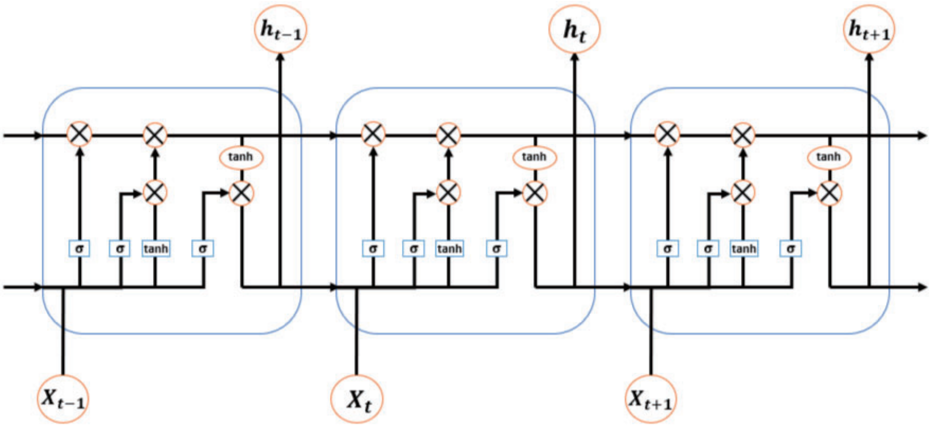


Figure 3.4: The temporal inner structure of LSTM [33, 35]

category of information will be forgotten and what kind of information will be remembered (saved) instead of saving all of the information which are input. Besides, LSTM network can avoid vanishing gradient problem and exploding gradient problem. Because of the character that LSTM network only remembers or saves the necessary or important information which is decided by forward layer.

After introducing the inner structure of LSTM, it is necessary to concentrate on the whole model with LSTM and explain how do 32 video segments be dealt with in the whole model [33].

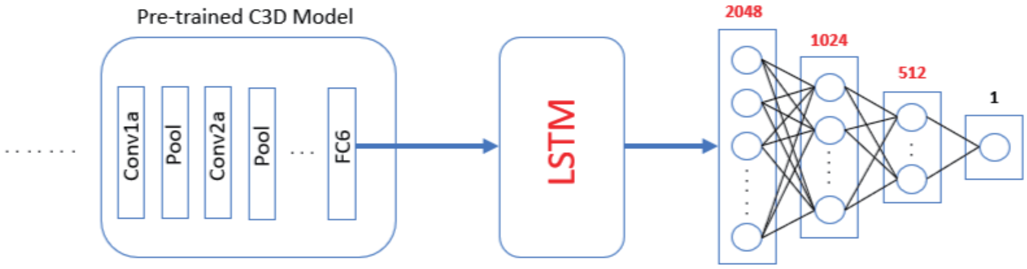


Figure 3.5: The Model with LSTM. The parameters of FCNN in this figure is "ParameterA".

In Fig. 3.5, the positive bag and negative bag consist of non-overlapping 32 video segments respectively are fed into a pre-trained C3D model to extract spatiotemporal features. The output of (FC) layer FC6 of the pre-trained C3D model [19] is a feature vector with 4096 dimension. And the feature vector will be fed into LSTM in Fig.3.4 to extract the temporal features between adjacent video segments in order to make FCNN make a better classification. After extracting temporal features between adjacent video segments by LSTM, the output is fed into FCNN directly in Fig. 3.5. In finial, we could obtain an anomalous score between 0 and 1 [33, 35].

3.2.3 Combing Bi-directional Long Short-term Memory layer to Fully Connected Neural Network

If the LSTM module is inserted between C3D model and FCNN, the performance is better. However, this only a speculation based on the theory. According to the experience of using LSTM and other models, there is a possibility that although LSTM is used to extract temporal features between adjacent video segments, the performance will be worse or cannot be better [35]. Because there is only a forward layer in LSTM module [33]. This will cause LSTM could not extract the temporal features efficiently. Besides, there is a huge number of parameters in LSTM and that will result in overfitting easily. Based on these two reasons, it is possible that performance of the model with LSTM will be affected [33, 35]. Thus, replacing LSTM with Bi-directional LSTM to obtain better performance is essential and necessary. Because Bi-directional LSTM could utilize a forward layer and backward layer to extracting the temporal features between adjacent video segments compared to the inner structure of LSTM [35]. Therefore, it is possible that Bi-directional LSTM with forward layer and backward layer is able to extract features more efficient than LSTM. The input and output are invariable comparing to the model with LSTM [33].

The inner structure of Bi-directional LSTM is given in Fig. 3.6. The input is still a feature vector with 4096 dimension from C3D model. Output is a 2048-dimension vector which could be fed into FCNN directly. The equations of Bi-LSTM are give as follows:

$$A_t = f(WA_{t-1} + UI_t) \tag{3.8}$$

$$A_t^{'} = f(W^{'}A_{t+1}^{'} + U^{'}I_t) \tag{3.9}$$

$$O_t = f(VA_t + V^{'}A_t^{'}) \tag{3.10}$$

In these equations, variate W and $W^{'}$ represent the weight of forward layer and backward layer. I_t and O_t mean input and output of bi-directional LSTM. A_t and $A_t^{'}$ represent the cells of forward layer and backward layer. In Fig. 3.6, orange layer and black layer are forward layer and backward layer. Comparing to the inner structure of LSTM, Bi-LSTM not only has forward layer but also backward layer. The LSTM only use forward layer to decide how much information can enter into cell and how much information

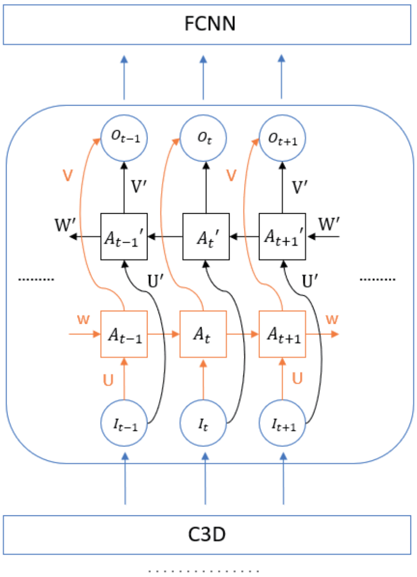


Figure 3.6: The inner structure of Bi-directional LSTM [36, 37]

could be sent into next cell or output. In Bi-LSTM, the forward layer and backward layer are used to decide how much information can enter into cell, respectively [34, 36, 37]. Besides, in the stage of output, forward layer and backward layer are used to decide how much information will be output in Bi-LSTM. This is the merit and difference between LSTM and Bi-LSTM [34, 36, 37]. Thus, it is the model with Bi-LSTM module will bring better performance.

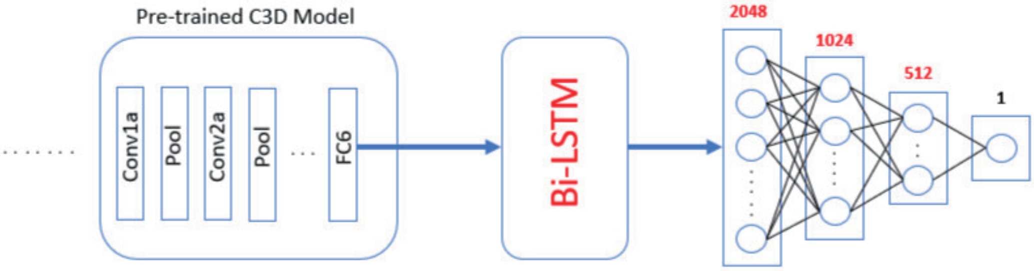


Figure 3.7: The model with Bi-directional LSTM. The parameters of FCNN in this figure is "ParameterA".

The output and input of Bi-LSTM are not changed, comparing to the model with LSTM module. Thus, the introduction of input will be skipped. However, the output and other parts of the model in Fig. 3.7 will be introduced. Concerning the parameter settings of the model with LSTM or

Bi-LSTM will be introduced in detail in Section 4 [34].

3.3 Dataset

UCF-Crime is a dataset proposed by Waqas Sultani et al., in 2018 [1]. It consists of 1900 long and untrimmed real-world surveillance videos. The 1900 long and untrimmed surveillance videos include 13 categories of abnormal actions including Abuse, Arrest, Assault, Arson, Road Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The total time of this dataset is 128 hours. And the details about this dataset will be shown in Table 3.1.

Anomaly	Number	Anomaly	Number
Abuse	50 (48)	Road Accidents	150 (127)
Arrest	50 (45)	Robbery	150 (145)
Arson	50 (41)	Shooting	50 (27)
Assault	50 (47)	Shoplifting	50 (29)
Burglary	100 (87)	Stealing	100 (95)
Explosion	50 (29)	Vandalism	50 (45)
Fighting	50 (45)	Normal events	950 (800)

Table 3.1: Total number of videos of every anomaly in UCF-Crime Dataset. The numbers in brackets represent the number of videos in training dataset.

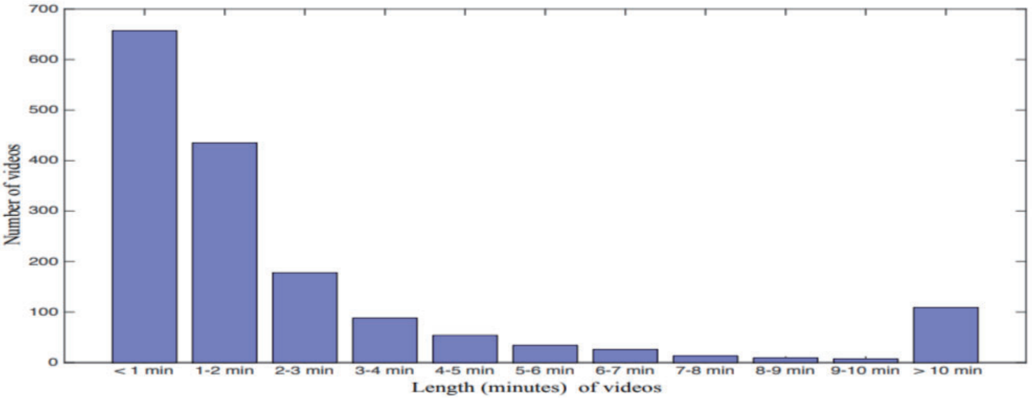


Figure 3.8: The time distribution of videos in training dataset [1]

.

The UCF-Dataset is divided into two parts: the training dataset and the testing dataset [1]. And there are 800 videos with normal actions and

810 with abnormal actions. In the testing dataset, there are 150 videos with normal actions and 140 videos with abnormal actions [1]. Besides, the training dataset and testing dataset include all 13 anomalies at various temporal locations in the videos [1].

In Fig. 3.8, it is obvious that most of the videos are short-time videos. There are two merits of this type of video. Firstly, the abnormal actions happen in a very short time, in general. Secondly, the short-time videos in the training dataset will not cost lots of training time. Therefore, all of the experiments of this research will be conducted in UCF-Crime Dataset.

Chapter 4

Experimentation and Evaluation

In this chapter, details about implementing the experiment will be introduced, evaluation metrics and the results of the experiment. Because there are lots of experiments in this research, in order to let readers understand this chapter easily, we will give a short analysis after giving the result corresponding to the model, respectively. And in finial, we analyze results of the baseline model and the model we proposed (final model) to give a conclusion.

4.1 Process of Model changing

Due to lots of experiments in this thesis, it is necessary to narrate the whole process of model changing in order to ensure readers could understand this research easily.

Because the parameter settings in Fully Connected Neural Network of the baseline model could be optimized further to make performance better, we change parameter settings in FCNN for finding a better performance of FCNN. And we denote this parameter setting which is mentioned in Section 3.2.1 as ”ParameterA”. However, the parameters of ParameterA are too many and we doubt that a huge number of parameters is an important reason to cause overfitting. Thus, except parameter settings of parameterA, we also found another parameter settings which bring a better performance of FCNN than baseline model and the parameters are less than ParameterA. We denote this parameter settings of FCNN as ”ParameterB”. Surely, the performance of ParameterA is better than ParameterB. After obtaining new parameter settings (ParameterA and ParameterB), we insert LSTM module between pre-tainded C3D model and FCNN to extract the temporal features between adjacent video segments. We doubt temporal features between adjacent video segments could not be extracted efficiently because there is only a forward layer in LSTM module. Thus, we tried to use Bi-directional LSTM includes the forward layer and backward layer to extract temporal features. In finial, we found Bi-directional LSTM could extract temporal features between adjacent video segments more efficiently and bring the best performance.

4.2 Implementation Details

Every video will be cut into 32 non-overlapping video segments and one video treated as a bag (positive or negative) [1]. Then, 30 positive bags and 30 negative bags will be chosen randomly to be fed into a pre-trained C3D model to extract spatiotemporal features [1]. The number of 32 and 30 is empirically set [1].

Before feeding positive and negative bags to C3D model, it is necessary to resize the video frames to 240 × 320 pixel and set the frame rate as 30 fps [1]. Then, extracting spatiotemporal features of 32 video segments in terms of computing C3D features for each 16-frames video clips of each video segment followed by l_2 normalization [1]. 16-frame clip features are taken the average of, in order to achieve features for every video segments [1]. Then, layer FC6 of pre-trained C3D model is connected to LSTM or Bi-directional LSTM model to extract temporal features between adjacent video segment [19]. In finial, training the FCNN in terms of feeding features of LSTM or Bi-directional LSTM to FCNN.

In this research, Drop regularization between every layer in FCNN is set as 60% [20]. Besides, ReLU activation and Sigmoid activation are used for the last layer. [1, 21]. Besides, we use Adagrad optimizer with an initial learning rate of 0.001 [22, 1]. The parameters in Eq. 3.6 and Eq. 3.7 are set as follows: $\lambda_1 = 8 \times 10^{-5}$, $\lambda_2 = 8 \times 10^{-5}$ and $\lambda_3 = 0.01$.

After computing gradients and loss, we back-propagate the loss for the whole batch by utilizing Eq. 3.6 and Eq. 3.7 [1].

There are only baseline model, baseline model with LSTM module and baseline model with Bi-LSTM (final model) in this research. But, there are two sets of parameters of FCNN and the Bi-LSTM module will be optimized once. So, some models are derived. And these derived models could not be ignored. Because if results of all of models will not be compared with each other, it is impossible to find a model with the best performance. For narrating and comparing the results of each model in convenient, the introduction of every model are given in Fig. 4.1.

Model	LSTM		Bi-LSTM		FCNN				
	Input	Output	Input	Output	Layer1	Layer2	Layer3	Layer4	Layer5
Baseline model	—	—	—	—	4096	512	32	1	—
FCNN-ParameterB	—	—	—	—	4096	512	64	1	—
FCNN-ParameterA	—	—	—	—	4096	1024	512	1	—
FCNN-ParameterB-LSTM	4096D	2048D	—	—	2048	1024	512	64	1
FCNN-ParameterA-LSTM	4096D	2048D	—	—	2048	1024	512	1	—
FCNN-ParameterB-BiLSTM’	—	—	4096D	2048D	4096	1024	512	64	1
FCNN-ParameterA-BiLSTM’	—	—	4096D	2048D	4096	1024	512	1	—
FCNN-ParameterB-BiLSTM	—	—	4096D	1024D	2048	1024	512	64	1
FCNN-ParameterA-BiLSTM	—	—	4096D	1024D	2048	1024	512	1	—

Figure 4.1: The structure introductions of each model. The ”D” in figure means ”Dimension” of the vector of output and input. ”—” means there is no that part in the model.

All of the model in Fig. 4.1 are based on the baseline model. ”FCNN-ParameterB” and ”FCNN-ParameterA” are the baseline model with optimized parameter settings (ParameterB and ParameterA) in FCNN. The ”FCNN-ParameterB” is a 4-layer FC neural network and it has 4096 units in the first layer. The first layer followed by 512 units, 64units and 1 unit FC layers [1]. In ”FCNN-ParameterA”, the first FC layer has 4096 units followed by 1024 units, 512 units and 1 unit FC layers [1].

The differences between ”FCNN-ParameterB-LSTM”, ”FCNN-Parameter A-LSTM” and baseline model is that LSTM module is inserted into baseline model. And the parameter settings of FCNN are changed. The input of LSTM module is a vector with 4096 Dimension. Output of LSTM is a 2048 dimensional vector. Besides, in Fully Connected Neural Network of FCNN-ParameterB-LSTM, there are 2048 units in the first FC layer. The first layer followed by 1024 units, 512units, 64 units and 1 unit FC layers. In FCNN-ParameterA-LSTM, the first FC layer has 2048 units followed by 1024 units, 512 units and 1 unit FC layers [1].

In final, the structure of ”FCNN-ParameterA-BiLSTM’” and ”FCNN-ParameterB-BiLSTM’” are almost the same with ”FCNN-ParameterA-BiLST M” and ”FCNN-ParameterB-BiLSTM”. There are two differences. The structures of Bi-LSTM and FCNN. Surely, these models are also based on baseline model. In ”FCNN-ParameterB-BiLSTM’” and ”FCNN-ParameterA-BiLSTM’”, input of Bi-LSTM module is a vector with 4096 Dimension. Output of Bi-LSTM is a 2048 deimensional vector. The first layer in FCNN of ”FCNN-ParameterB-BiLSTM’” is FC layer with 4096 units. And it followed by 1024 units, 512units, 64 units and 1 unit FC layers. In FCNN of ”FCNN-ParameterA-BiLSTM’”, the first layer has 4096 units followed by 1024 units, 512 units and 1 unit FC layers [1]. In ”FCNN-ParameterB-BiLSTM” and ”FCNN-ParameterA-BiLSTM”, the output of Bi-LSTM is a vector with 1024 demension. The inner structure of FCNN of ”FCNN-ParameterB-BiLSTM” is the same with ”FCNN-ParameterB-BiLSTM’”. Similarly, inner structure of FCNN of ”FCNN-ParameterA-BiLSTM” is the same with ”FCNN-ParameterA-BiLSTM’”.

4.3 Evaluation Metrics

Although, ROC and AUC are important and comprehensive evaluation metrics [30, 32]. But, it is not enough to only use ROC and AUC to evaluate our model. Thus, except for ROC and AUC, F-measure, Recall and Loss Convergence Rate are also used to evaluate our model [32].

4.3.1 Loss Convergence Rate

Loss convergence Rate is a relatively weak metric to evaluate the model. It reveals when does loss function finishes convergence. Loss function converges faster without changing the learning rate, iterations and so on, the parameter settings of the model are optimized better.

4.3.2 ROC and AUC

Receiver Operating Characteristic is a contingency table that consists of True Positive Rate (TPR) and False Positive Rate (FPR). The TPR and FPR are the horizontal coordinate and vertical coordinate. From Eq. 4.1, it reveals TPR means how many correct positive results occur among all positive samples available. From Eq. 4.2, it reveals FPR defines how many incorrect positive results occur among all negative samples available. All of the parameters in Eq. 4.1 and Eq. 4.2 are given in Tab. 4.1.

		True condition	
	Total population	Condition positive	Condition negative
Predicted condition	Predicted condition positive	True positive (TP)	False positive (FP)
	Predicted condition negative	False negative (FN)	True negative (TN)

Table 4.1: The relationship between True condation and Predicted condation

$$TPR = \frac{TP}{TP + FN} \tag{4.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{4.2}$$

The ROC Space is given in Fig. 4.2. All of the points in dashed in the figure represent TRP = FPR. If ROC of a model matches with dashed in Fig. 4.2, that means a prediction of this model is 50%. If ROC of a model is under the dashed, the model should not be used because the prediction is under 50%. The perfect point of ROC Space is (1, 0), that represents the model could predict all of the positive samples correctly. Because TPR = 1 and FPR = 0. Therefore, ROC is closer to (1, 0), the classification of the model is better. ROC is a full-scale evaluation metric that is used to evaluate a model is an advantage or not.

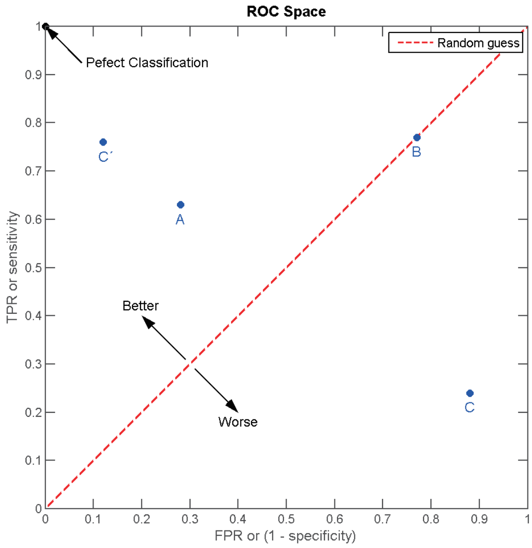


Figure 4.2: ROC Space (https://en.wikipedia.org/wiki/Receiver_operating_characteristic)

.

However, in reality, it is difficult for to distinguish the advantage model or disadvantage model in terms of only comparing the ROC of model respectively. Because there is some model with similar ROC. At this time, AUC is a good auxiliary evaluation metric to evaluate a model. Area Under Curve means the area under the ROC curve. Sometimes, At this time, AUC is worked. A advantage model could be distinguished with a disadvantage model easily by AUC which is expressed as a percentage. So, AUC and ROC are used together by researchers. If the AUC of one model is 1, the model is a perfect model in the world. But, that is not reality. Thus, if a model with $AUC > 0.5$, that is an acceptable model.

4.3.3 Recall

Recall represents how many positive samples are predicted right [32]. This could represent the capability of detecting positive samples. And the equation is given as follows:

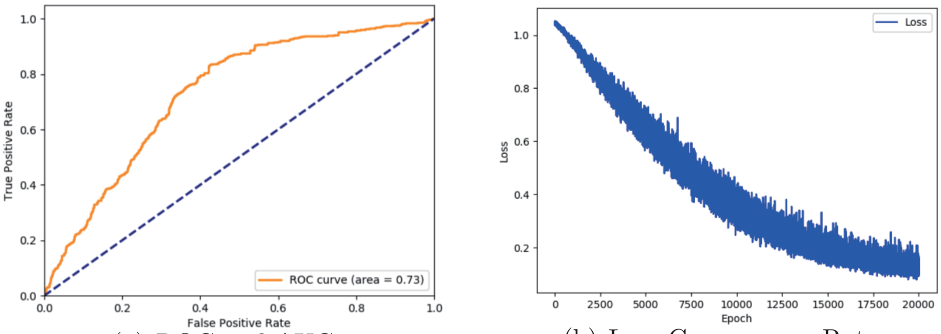
$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

4.3.4 F1-Measure

F1-Measure is a measure of a test's accuracy [31, 32]. It considers both the precision and the recall of the test. F1-Measure is the harmonic mean of the precision and recall and the score of F1-Measure is between 0 and 1 [31, 32]. If F1 = 1, that means the model had the best precision and recall. If F1 = 0, the performance of the model is the worst. In general, if $F1 > 0.3$, a conclusion could be given that the model is good and reliable.

$$F1 = \frac{2TP}{2TP + FN + FP} \tag{4.4}$$

4.4 Results



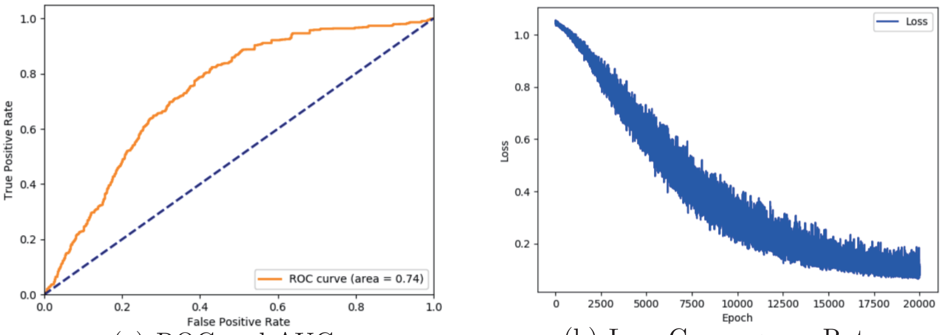
(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.3: ROC and Loss Convergence Rate of Baseline Model

This is the ROC-AUC and Loss Convergence Rate of the baseline model. The AUC of baseline model is 0.73 in Fig. 4.3 [32]. The loss function starts converging from 0 epoch and finishes on 20000th epoch. The convergence time of loss is very long.

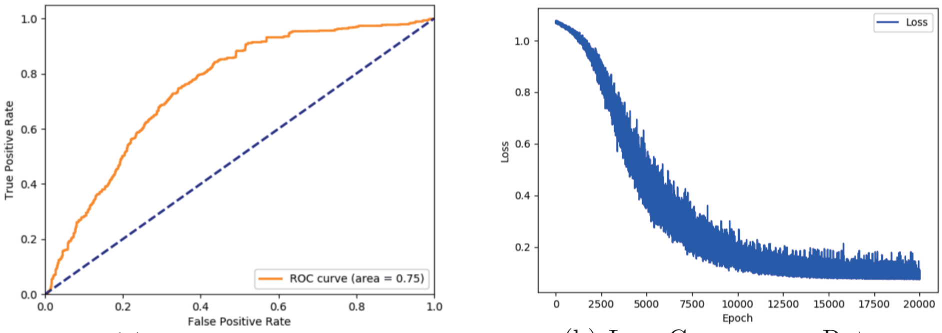
After doing 6 experiments, we found FCNN-ParameterA and FCNN-ParameterB could bring better performance than the baseline model. And parameters of FCNN-ParameterB is less than FCNN-ParameterA. The Fully Connected Neural Network of FCNN-ParameterA consists of 4 layers. There are 4096 units in the first layer which are followed by 1024 units, 512 units and 1 unit FC layers [1]. FC Neural Network of FCNN-ParameterB consists of 4 layers. The first FC layer has 4096 units followed by 512 units, 64 units and 1 unit FC layer. In terms of comparing Fig. 4.4 and Fig. 4.5 with Fig. 4.3, we can find that the performance of FCNN-ParameterA is the best. And



(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.4: ROC and Loss Convergence Rate of FCNN-ParameterB Model



(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.5: ROC and Loss Convergence Rate of FCNN-ParameterA Model

loss convergence rate is the fastest. The loss function finishes converging almost on the 10000th epoch. The loss function of FCNN-ParameterB finishes converging almost on the 17500th epoch.

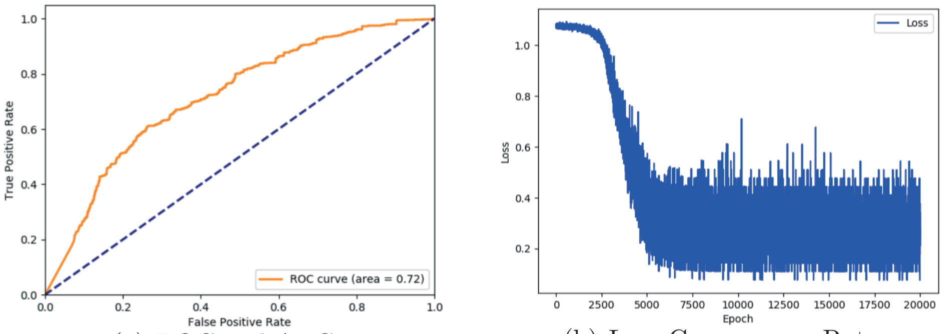
Besides, according to Tab. 4.2, we found Recall, ROC, AUC are the best in FCNN-ParameterA but F1-Measure is worse than FCNN-ParameterB. The reason is that the Recall of FCNN-ParameterA is higher too much than FCNN-ParameterB. Because F1-Measure is an evaluation metric consists of Precision and Recall. If Recall higher than Precision too much, F1-Measure will decline. Vice versa. The higher recall means the capability of detecting abnormal actions of FCNN-ParameterA is better.

In order to extract temporal features between adjacent video segments, LSTM module is inerted between C3D model and FCNN. The consequences are given in Fig. 4.6 and Fig. 4.7. Because of LSTM module, parameter settings of FCNN have been optimized further. The input of LSTM is a spatiotemporal feature(4096D) from FC6 of the C3D model [1] and output is a vector (2048D). Then we feed the vector (2048D) to Fully Connected Neural

Model	F1-Measure	Recall
Baseline model	0.260	0.55
FCNN-ParameterB	0.272	0.687
FCNN-ParameterA	0.269	0.781

Table 4.2: F1-Measure and Recall

Network to do classification. The structure of a Fully Connected Neural Network in FCNN-ParameterA-LSTM is a 3-layer FC neural network[1]. There are 1024 units in the first layer which fellowed by 512 units and 1 unit layers [1]. Fully Connected Neural Network in FCNN-ParameterB-LSTM has 1024, 512, 64 and 1 unit in the first, second, third and fourth layers [1].

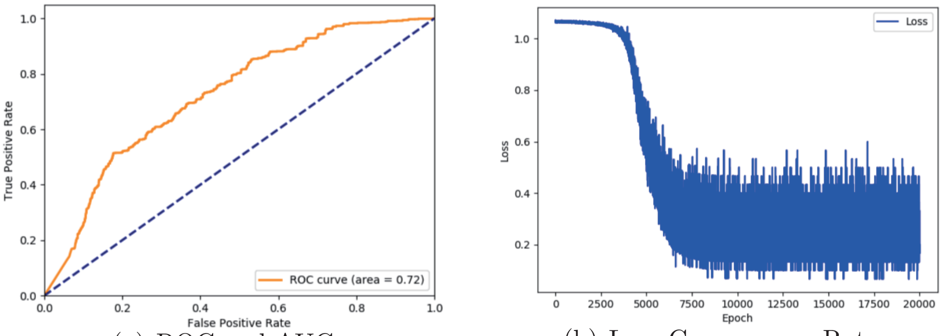


(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.6: ROC and Loss Convergence Rate of FCNN-ParameterB-LSTM Model

Comparing Fig. 4.6 and Fig. 4.7 with Fig. 4.3, we found that although the loss convergence rate is faster, AUC becomes worse after inserting LSTM module between C3D model and FCNN. According to Tab. 4.3, Recall becomes worse too. Because LSTM module could extract temporal features between adjacent video segments, the performance could be better, at least AUC should be higher if LSTM module is inserted between C3D model and FCNN. However, the performances are not very satisfactory. Thus, overfitting is the most reasonable reason for resulting in bad performance. Because the capability of extracting temporal features between adjacent videos is not very remarkable, in the meantime, a huge number of parameters are brought to the model. These two elements lead to overfitting and make performance worse than the model without the LSTM module. According to Tab. 4.3, comparing FCNN-ParameterA-LSTM and FCNN-ParameterB-LSTM than FCNN-ParameterA, FCNN-ParameterB, it shows that F1-Measures are better but Recall declined. This is not satisfactory because the decline of recall



(a) ROC and AUC

(b) Loss Convergence Rate

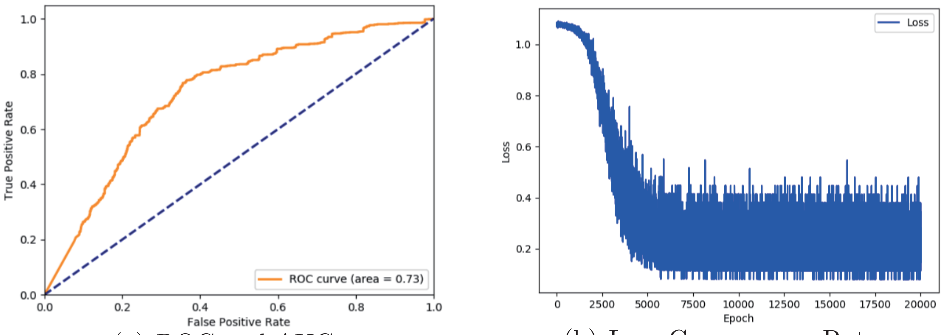
Figure 4.7: ROC and Loss Convergence Rate of FCNN-ParameterA-LSTM Model

Model	F1-Measure	Recall
Baseline model	0.260	0.55
FCNN-ParameterB	0.272	0.687
FCNN-ParameterA	0.269	0.781
FCNN-ParameterB-LSTM	0.276	0.377
FCNN-ParameterA-LSTM	0.278	0.397

Table 4.3: F1-Measure and Recall

reveals the capability of recognizing positive samples (abnormal actions) decline. This is cannot be ignored in the field of anomaly detection. Therefore, it is necessary to optimize the model further. Considering overfitting results in the worse performance, LSTM module is replaced with Bi-directional LSTM module which could extract temporal features more efficiently. The input of Bi-LSTM module is a spatiotemporal features(4096D) from FC6 of the C3D model [1] and output is vector (4096D). Then we feed the vector (4096D) to Fully Connected Neural Network to do classification. The structure of a Fully Connected Neural Network is unchanged. That means LSTM module is replaced with Bi-LSTM module only without changing any parameters.

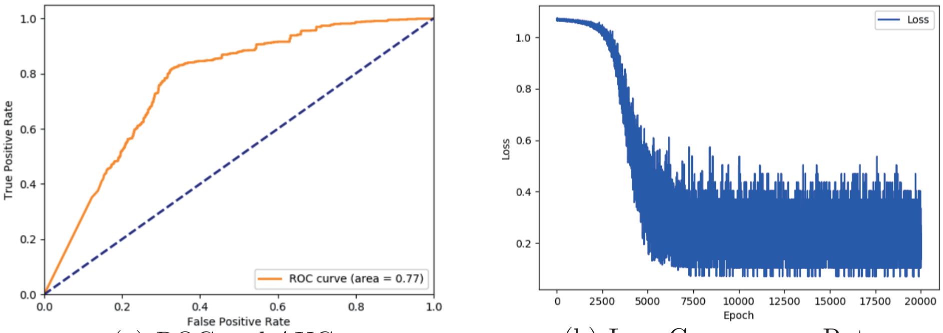
According to Fig. 4.8 and Fig. 4.9, it shows the performances improve obviously in terms of replacing LSTM module with Bi-directional LSTM module. Comparing Fig. 4.8 with Fig. 4.6, it is obvious that AUC increases 1%. The changing of the loss convergence rate is not conspicuous. And AUC increases 5%, if we compare Fig. 4.9 to Fig. 4.7. The loss convergence rate does not change a lot. According to Tab. 4.4, comparing FCNN-ParameterB-BiLSTM' and FCNN-ParameterA-BiLSTM' with



(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.8: ROC and Loss Convergence Rate of FCNN-ParameterB-BiLSTM' Model



(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.9: ROC and Loss Convergence Rate of FCNN-ParameterA-BiLSTM' Model

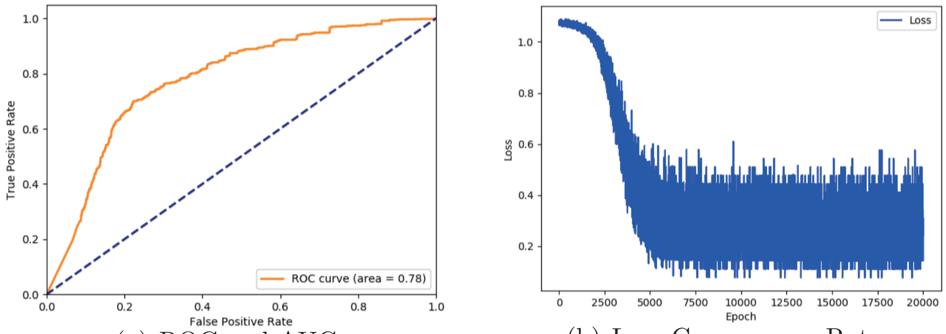
FCNN-ParameterB-LSTM and FCNN-ParameterA-LSTM respectively, F1-Measure and Recall increased.

It is obvious that the Recall and F1-Measure of FCNN-ParameterA-BiLSTM' are the best. Thus, two conclusions could be obtained. Firstly, Bi-directional LSTM is worked and that can make the performance of the model better. Secondly, because there is one more FC layer in FCNN-ParameterB-BiLSTM' than FCNN-ParameterA-BiLSTM', the overfitting has existed already in FCNN-ParameterB-BiLSTM'. And the overfitting affects the AUC performance of FCNN-ParameterB-BiLSTM'.

Although replacing the Bi-directional LSTM module with LSTM module and the performance of the model has been better, it is necessary to optimize the parameter settings of Bi-directional LSTM module further in order to obtain the better performance. Because it is possible that there is an overfitting phenomenon in FCNN-ParameterB-BiLSTM'. Therefore, we

Model	F1-Measure	Recall
FCNN-ParameterB-LSTM	0.276	0.377
FCNN-ParameterA-LSTM	0.278	0.397
FCNN-ParameterB-BiLSTM'	0.242	0.352
FCNN-ParameterA-BiLSTM'	0.295	0.651

Table 4.4: F1-Measure and Recall



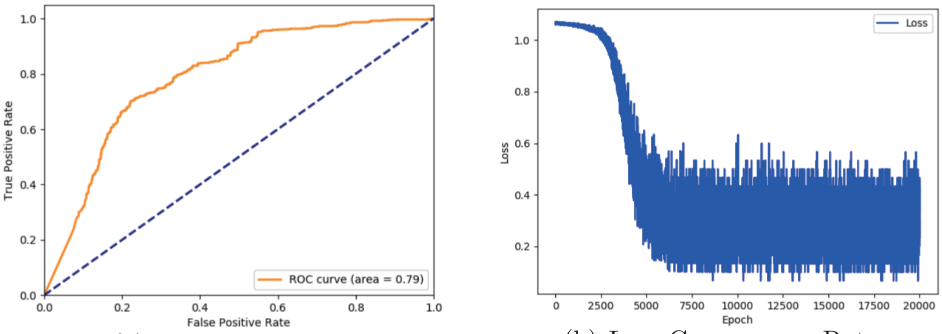
(a) ROC and AUC

(b) Loss Convergence Rate

Figure 4.10: ROC and Loss Convergence Rate of FCNN-ParameterB-BiLSTM Model

optimize the parameters and make the output of Bi-directional LSTM is a vector (2048D). Input is still a spatiotemporal features(4096D) from FC6 of the C3D model [1]. The structure of a Fully Connected Neural Network is unchanged. After adjusting the parameters of Bi-directional LSTM, we obtain new results which are given in Fig. 4.10 and Fig. 4.11. Comparing Fig. 4.3, Fig. 4.4, Fig. 4.6, Fig. 4.8 and Fig. 4.10, we could find the AUC performance of FCNN-ParameterB-BiLSTM is the best in the model which are based on Fully Connected Neural Network with parameterB. The AUC performance of FCNN-ParameterA-BiLSTM is the best in the model which are based on Fully Connected Neural Network with parameterA. Besides, AUC performance of FCNN-ParameterB-BiLSTM is better than FCNN-ParameterB-BiLSTM.

From the Fig. 4.14, F1-Measure is the highest of all of the model. Besides, Recall is also the highest of all the model. That reveals that the capability of detecting abnormal actions of FCNN-ParameterA-BiLSTM is the best. And F1-Measure shows FCNN-ParameterA-BiLSTM is the most stable model of all of the model. Thus, the conclusion can be given that the model of FCNN-ParameterB-BiLSTM has the best performance. And the model is given in Fig. 3.7.



(a) ROC and AUC

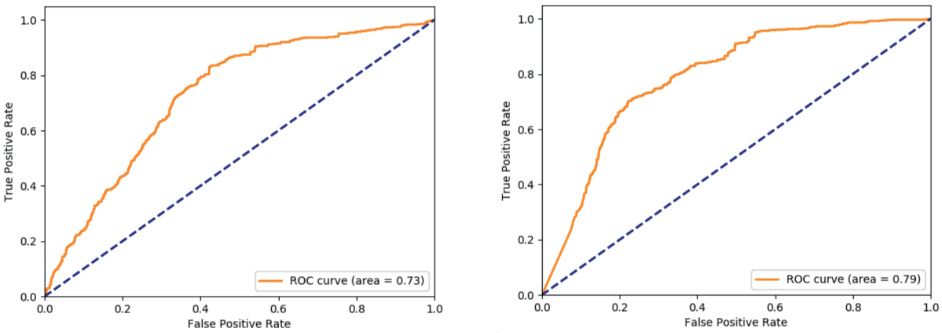
(b) Loss Convergence Rate

Figure 4.11: ROC and Loss Convergence Rate of FCNN-ParameterA-BiLSTM

4.5 Comparison and Analysis

In Section 4.4, the results and process of optimization of the model have been given. For the convenience of analyzing baseline model and final model (FCNN-ParameterA-BiLSTM), the evaluation metrics of these two models are compared only.

According to Fig. 4.12, it shows the ROC of baseline and finial model are smooth. It represents there is no overfitting in these two models. Besides, from Fig. 4.13, ROC of the final model covers the ROC of baseline model fully and AUC in Fig. 4.12(b) is 6% than baseline model. Thus, a conclusion can be given that from ROC's point of vies, the performance of final model is better than the baseline model.



(a) Baseline Model

(b) FCNN-ParameterA-BiLSTM Model

Figure 4.12: Comparison of Baseline and FCNN-ParameterA-BiLSTM Model (ROC, AUC)

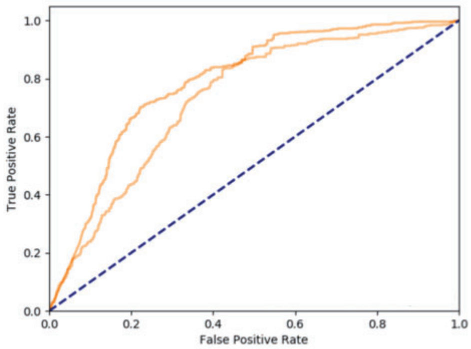


Figure 4.13: ROC of Baseline and FCNN-ParameterA-BiLSTM model

Model	AUC	F1-Measure	Recall
Baseline model	0.73	0.260	0.554
FCNN-ParameterA-BiLSTM	0.79	0.351	0.665
Performance Improvement	0.06	0.091	0.111

Table 4.5: AUC, F1-Measure and Recall of baseline and FCNN-ParameterA-BiLSTM model

By comparing the final model with baseline model, it reveals Recall increases 11.1%. That means the possibility of detecting abnormal actions successfully has been increased by 11.1%. The abnormal actions which are could not be detected successfully by the baseline model could be detected successfully by our model. Furthermore, recall means how many the positive samples are detected in all of positive samples. It is shown clearly in Eq. 4.1. In the field of anomaly detection, the positive sample is abnormal action. Our goal is to detect all of the abnormal actions in terms of surveillance videos. In another word, high recall means the high possibility of detecting abnormal actions. But not the higher the better. Sometimes we had better to decline recall to improve prediction. Because if the model detects every video as abnormal video, all of the positive samples absolutely will be detected and recall surely is 100%. Obviously, this model is a bad. Because the prediction will be very very low.

Thus, it is essential to use another evaluation metric —F1-Measure to evaluate our model. Because of Eq. 4.4, it shows F1-Measure is an evaluation metric that is affected by recall and prediction. It means a model could not obtain a good F1-Measure score unless the model could obtain a balance between good recall and prediction. Therefore, F1-Measure is a full-scale evaluation metric. In general, if the score of F1-Measure is more than 0.3, the model could be considered as a good model. According to Tab. 4.5, the F1-Measure score changed from 0.260 to 0.351. The growing rate is 9.1%. It is obvious that F1-Measure improves because of the improvement of Recall.

Concerning the Loss Convergence Rate, that is a weak evaluation metric, so we don't compare the loss convergence rate of baseline and FCNN-ParameterA-BiLSMT model in terms of giving images of Loss Convergence Rate, respectively. From Tab. 4.14, it is obvious that baseline model converges slower than the FCNN-ParameterA-BiLSMT model. Thus, if the loss convergence rate of a model becomes faster without changing the learning rate and loss function, the model is optimized better.

After comparing every evaluation metric of baseline and final model, we could conclude that the finial model (FCNN-ParameterA-BiLSTM) obtained better performance than the baseline model.

Model	AUC	F1-Measure	Recall	Loss Convergence Rate (Approximately)
Baseline model	0.73	0.260	0.55	20000th epoch
FCNN-ParameterB	0.74	0.272	0.687	17500th epoch
FCNN-ParameterA	0.75	0.269	0.781	12500th epoch
FCNN-ParameterB-LSTM	0.72	0.276	0.377	6000th epoch
FCNN-ParameterA-LSTM	0.72	0.278	0.397	8000th epoch
FCNN-ParameterB-BiLSTM’	0.73	0.242	0.352	6000th epoch
FCNN-ParameterA-BiLSTM’	0.77	0.295	0.651	6000th epoch
FCNN-ParameterB-BiLSTM	0.78	0.345	0.620	5500th epoch
FCNN-ParameterA-BiLSTM	0.79	0.351	0.665	5500th epoch

Figure 4.14: AUC, F1-Measure, Recall and Loss Convergence Rate of each model

Chapter 5

Conclusion and Further Work

In this thesis, our research ”A Study on Anomaly Detection in Surveillance Videos” is presented. In this research, the new proposed deep learning approach are used to detect abnormal actions in terms of utilizing surveillance videos [1]. In the past, lots of researchers use only abnormal actions or abnormal actions to train a model in order to obtain a good performance in detecting abnormal actions by using surveillance videos [1]. However, the results are not very satisfactory. Thus, in this research, we use normal actions and abnormal actions to train the model to detect abnormal actions [1]. Our research focuses on improving the performance of the anomaly detection model which is proposed by Sultani et al. in 2018 [1]. Especially, improving the performance of ROC, AUC, F1-Measure, and Recall. We take a model proposed by Sultani et al. in 2018 [1] as our baseline mode. And analyzing why our model could obtain better performance than the baseline model. The most important contributions of my research as follows:

- After doing 6 experiments in terms of only optimizing parameter settings in Fully Connected Neural Network, a set of parameters which could improve the performance of FCNN has been found. And the structure of FCNN with best performance is a 3-layer FC neural network (FCNN-ParameterA) [1]. There are 1024 units in the first FC layer which are followed by 512 units and 1 unit FC layer [1]. Surely, the structure of FCNN changed, after LSTM or Bi-directional LSTM module is inserted between the pre-trained C3D model and FCNN [1]. However, that is another story. Because based on FCNN-ParameterA, all of the models in this research are designed. Even though, the structure of FCNN changed, the function of FCNN-ParameterA is still important and it is indispensable.

- In order to extract temporal features between adjacent video segments, LSTM and Bi-directional LSTM module are inserted between C3D model and FCNN. Besides, we also give the analysis on every experiment and reveal why the performance become worse or better after inserting LSTM or Bi-directional LSTM module between C3D and FCNN. Especially, after inserting LSMT module into the model, the performance become worse. That is not a positive case. However, the analysis and explanation are given. And it is obvious that the analysis will be meaningful to others' researches in future. According to Fig. 4.14, it reveals the performance of FCNN-ParameterA-BiLSTM is the best. Comparing to baseline model, AUC and ROC increase 6% and F1-Measure increase almost 9.1%. It means the strategy that in terms of extracting temporal features between adjacent video segments, improving performance of the whole model is successful. Toward to extract temporal features from adjacent video segments, Bi-directional LSTM module could extract features in more efficient. And will contribute to someone's research in future.

Although comparing to baseline model, the better performance has been obtained, there are still some limitations in our model. If these limitations are solved, the performances will become better.

- The Recall of our model is 0.665. And that means the possibility that detecting normal action as abnormal action is high. And if the system developed based on our model go live, the mis-detection will cause lost of problems and waste of resource of enforcement agencies [1]. Thus, finding some way to decline the recall and improve the F1-Measure. This is a important theme in future.

- The input of C3D model is only 16 frames whatever how long does the video segments. And the performance could be improved further if a new model that could decide the number of frames of input based on the time of video segment could be found. If the time of the video segment is short, the model will decrease the number of frames of input to save the resource of computation. If the time of video segment is very long, the model will decrease the number of frames of input to extract more spatiotemporal features.

- Last but not least, it is also possible to combine the visualization algorithm to this model. We could know which part of the frames is used to extract spatiotemporal features by the model. And the performance could be improved easily because we could know how does the model learns or extract the spatiotemporal features from inputs.

We think other researchers who are interested in anomaly detection could spare no effort to solving the limitations of our model which had been introduced above. And if the three problems are solved well, detecting abnormal actions to keep the safety of public space is not just a dream and we think that could be realized in the future.

Bibliography

[1] Waqas Sultani, Chen Chen, and Mubark Shah. Real-world anomaly detection in surveillance videos. In CVPR, 2018. 2

[2] X. Cui, Q. Liu, M. Gao, and D.N. Metaxas, “Abnormal Detection Using Interaction Energy Potentials,” Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2011.

[3] S. Mohammadi, A. Perina, H. Kiani, and M. Vittorio. Angry crowds: Detecting violent events in videos. In ECCV, 2016.

[4] H. Seki and Y. Hori: ”Detection of abnormal human action using image sequences”, PTOC. of International Power Electronics Conference (IPECZOOO), Tokyo, pp.1272-1277 (2000)

[5] S.-H. Cho and H.-B. Kang, “Abnormal behavior detection using hybrid agents in crowded scenes,” Pattern Recognit. Lett., vol. 44, pp. 64–70, Jul. 2014.

[6] D. Denning. An Intrusion-Detection Model. IEEE Transactions on Software Engineering, February 1987.

[7] W. Luo, W. Liu, and S. Gao. Remembering history with convolutional LSTM for anomaly detection. In 2017 IEEE International Conference on Multimedia and Expo, ICME 2017, Hong Kong, China, July 10-14, 2017, pages 439–444, 2017.

[8] Wang T, Snoussi H (2014) Detection of abnormal visual events via global optical flow orientation histogram. IEEE Trans Inf Forensic Secur 9(6):988–998

[9] T. Joachims. Optimizing search engines using clickthrough data. In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, 2002.

[10] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In CVPR, 2014.

[11] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In NIPS, pages 577–584, Cambridge, MA, USA, 2002. MIT Press.

[12] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. ACM Comput. Surv., 2009.

[13] B. Anti and B. Ommer. Video parsing for abnormality detection. In ICCV, 2011.

[14] T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behaviour in video. In ICCV, 2009.

[15] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. In CVPR, 2009.

[16] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. In ICCV, 2013.

[17] B. Zhao, L. Fei-Fei, and E. P. Xing. Online detection of unusual events in videos via dynamic sparse coding. In CVPR, 2011.

[18] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. In CVPR, June 2016.

[19] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 2014.

[21] G. E. Hinton. Rectified linear units improve restricted boltzmann machines vinod nair. In ICML, 2010.

[22] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res., 2011.

[23] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence, 89(1):31–71, 1997.

[24] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe. Learning deep representations of appearance and motion for anomalous event detection. In BMVC, 2015.

[25] S. Wu, B. Moore, and M. Shah. Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes. In CVPR, 2010.

[26] A. Basharat, A. Gritai, and M. Shah. Learning object motion patterns for anomaly detection and improved object detection. In CVPR, 2008.

[27] X. Cui, Q. Liu, M. Gao, and D. N. Metaxas. Abnormal detection using interaction energy potentials. In CVPR, 2011.

[28] Y. Zhu, I. M. Nayak, and A. K. Roy-Chowdhury. Contextaware activity recognition and anomaly detection in video. In IEEE Journal of Selected Topics in Signal Processing, 2013.

[29] W. Li, V. Mahadevan, and N. Vasconcelos. Anomaly detection and localization in crowded scenes. TPAMI, 2014.

[30] Fawcett T (2006) An introduction to ROC analysis. Pattern Recogn Lett 27:861–874

[31] Derczynski, L. Complementarity, F-score, and NLP Evaluation. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, 23–28 May 2016.

[32] D. M. Powers, “Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation,” 2011.

[33] Cheng Jianpeng, Dong Li, Lapata Mirella. Long short-term memory-networks for machine reading. CoRR. 2016 abs/1601.06733.

[34] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In IEEE Workshop on ASRU, 2013.

[35] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. PAMI, 31(5):855–868, 2009.

[36] M. Schuster and K.K. Paliwal, “Bidirectional Recurrent Neural Networks,” IEEE Trans. Signal Processing, vol. 45, pp. 2673-2681, Nov. 1997.

[37] Kiperwasser, E. and Goldberg, Y. Simple and accurate dependency parsing using bidirectional LSTM feature representations. TACL, 4:313–327, 2016.

[38] "Public Area CCTV and Crime Prevention: An Updated Systematic Review and Meta-Analysis". Journalist's Resource.org. 11 February 2014.

[39] "Rise of Surveillance Camera Installed Base Slows" .May 5, 2016. Retrieved January 5, 2017.