Deep Learning Methods for Human Action Recognition

Zeqi Yu

A thesis submitted to the Auckland University of Technology
in partial fulfillment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2020

School of Engineering, Computer & Mathematical Sciences

Abstract

Human action recognition from digital videos is a hot topic in the field of computer vision. It has a pretty assortment of applications in a myriad of fields such as video surveillance, human-computer interaction, visual information retrieval, and unmanned driving. With the exponential growth of surveillance data on the Internet in recent years, how to implement effective and efficient analysis of the video data is extremely crucial. Traditional machine learning methods that only extract computable features have limitations and do not suit massive visual data, meanwhile deep learning methods, especially convolutional neural networks, have gained great attainments in this field.

The goal of human action recognition is to classify patterns so as to understand human actions from visual data and export corresponding tags. In addition to spatial correlation existing in 2D images, human actions in a video own the correlation in temporal domain. Due to complexity of human actions, changes of perspectives, background noises, and lighting conditions will affect the recognition.

In order to solve these thorny problems, three algorithms are designed and implemented in this thesis. Based on convolutional neural networks (CNN), Two-Stream CNN, CNN+LSTM, and 3D-CNN are harnessed to identify human actions. Each algorithm is explicated and analyzed on details. HMDB-51 dataset is employed to test these algorithms and gain the best results. Our experimental results demonstrate that the three methods have effectively identified human actions in given videos, the best algorithm thus is verified.

Keywords: Human action recognition, convolutional neural network, deep learning, LSTM, 3D-CNN, Two-Stream CNN

Table of Contents

Abstract I
Table of Contents II
List of Figures IV
List of Tables V
List of Algorithms VI
Attestation of Authorship VII
Acknowledgment VIII
Chapter 1 Introduction 1
1.1 Background and Motivations 2
1.2 Research Question 5
1.3 Contributions 8
1.4 Objective of This Thesis 10
1.5 Structure of This Thesis 12
Chapter 2 Literature Review 14
2.1 Introduction 15
2.2 Deep Learning 17
2.3 Deep Learning Models for Object Recognition 19
2.3.1 R-CNN 20
2.3.2 Fast R-CNN 21
2.3.3 Faster R-CNN 21
2.3.4 YOLO 22
2.3.5 SSD 27
2.3.6 Object Tracking 28
2.4 Human Action Recognition 29
2.4.1 Statement of Human Action Recognition 29
2.4.2 Methods of Online Human Action Recognition 31
Chapter 3 Methodology 34
3.1 Research Design 35
3.2 CNN+LSTM 35
3.2.1 Convolutional Neural Network 35
3.2.2 Long Short-Term Memory 37

3.3 The Two-Stream CNN Network 39
3.3.1 Basic Structure 41
3.3.2 Model Training Process 43
3.3.3 Network Test Process 45
3.4 3D Convolutional Network 45
3.5 Evaluation Methods 48
Chapter 4 Results 50
4.1 Data Collection and Experimental Environment 51
4.2 Results of Human Action Recognition 53
4.3 Demonstrations 55
4.4 Limitations of This Research Project 61
Chapter 5 Analysis and Discussions 63
5.1 Analysis 64
5.1.1 Dataset 64
5.1.2 Experimental Design 65
5.1.3 Action Recognition Based on the Two-Stream Network 66
5.1.4 Action Recognition Based on 3D CNN 66
5.1.5 Action Recognition Based on CNN+LSTM Model 67
5.2 Discussions 69
Chapter 6 Conclusion and Future Work 74
6.1 Conclusion 75
6.2 Future Work 77
6.2.1 More Datasets 77
6.2.2 Future Experiments 79
6.2.3 Evaluation Methods and Applications 80
References 81

List of Figures

Figure 2.1 The network structure for human action recognition.....28

Figure 3.1 The pipeline of human action recognition……..……35

Figure 3.2 Basic structure of CNNs……..……36

Figure 3.3 The structure of CNN+LSTM network…….…….……39

Figure 3.4 The structure of the Two-Stream CNN Network…….……40

Figure 3.5 An example of convolution operations……..…..……42

Figure 3.6 An example of pooling operations…….……...……42

Figure 3.7 The information channel for each frame…….……47

Figure 3.8 The convolution of the spatial dimensions.....48

Figure 3.9 Human action recognition based on C3D algorithm…..…….....48

Figure 4.1 The 51 categories in HMDB-51.....52

Figure 4.2 The accuracy of three algorithms for human action recognition.....54

Figure 4.3 The accuracy rates of the three methods for human action recognition.....54

Figure 4.4 Training accuracy and loss rate of the CNN+LSTM.....55

Figure 4.5 The results of human action recognition by using CNN+LSTM…….....56

Figure 4.6 Training accuracy and loss rate of the Two-Stream CNN network.....57

Figure 4.7 The results of human action recognition by using the Two Stream CNN network.....57

Figure 4.8 Training accuracy and loss rate of the 3D CNN……..…….……58

Figure 4.9 The results of human action recognition by using 3D CNN…...……...…….58

Figure 4.10 The recognition results of human action: Drinking with a water glass.....59

Figure 4.11 Recognition results in dark environment.....60

Figure 4.12 The recognition results of human action: Wearing with reflective clothing.....60

Figure 5.1 Different feature pooling architectures.....68

List of Tables

Table 3.1 Confusion matrix.....49

Table 4.1 The accuracy of these three models for human action recognition.....53

Table 5.2 The accuracy of recognition with different method.…….…….…….....66

List of Algorithms

Algorithm 3.1 CNN+LSTM…….…….....35

Algorithm 3.2 The Two-Stream CNN Network.....39

Algorithm 3.3 3D Convolutional Network……...…….….....45

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature: Date: 23 October 2020

Acknowledgment

This thesis was accomplished as the part of the Master of Computer and Information Sciences (MCIS) course at the School of Computer and Mathematical Sciences (SCMS) at the Auckland University of Technology (AUT) in New Zealand. I would like to deeply thank my parents for the financial support they have provided during my entire time of academic study in Auckland.

My deepest thanks are given to my supervisor Dr. Wei Qi Yan. I would like to thank him for his guidance on this MCIS thesis. From the thesis proposal to final fulfillment, Dr. Yan devoted a lot of time and energy to answer my questions. Meanwhile, my writing skills have been uplifted by leaps and bounds. Also, I am grateful to the classmates around me. Thank them for the zealous and ardent assistance amid working for this thesis project.

Zeqi Yu

Auckland, New Zealand

October 2020

Chapter 1
Introduction

The first chapter of this thesis is split into five parts. Firstly, the background and motivation of this thesis are explicated. In this thesis, we expound the development of deep learning methods and machine vision algorithms in the field of artificial intelligence as well as a wide spectrum of applications for human action recognition. Then, major research content and relevant algorithms of this thesis are presented. The goal of this thesis is addressed in the context. At the end of this chapter, the structure of this thesis is delineated.

1.1 Background and Motivations

Human action recognition from videos is based on the analysis of a sequence of video frames by using computers, so as to automatically find human actions without manual operations (Popoola & Wang, 2012). In the era of the Internet with mobile phones, people's daily lives have been surrounded by access control of building gates, traffic sensors, security cameras, and many others. The ubiquitous cameras enable everyone's actions in public to be monitored, identification of human actions in surveillance videos has tremendous significance in the field of cybersecurity (Bruno, Mastrogiovanni, Sgorbissa, Vernazza & Zaccaria, 2013). In addition, analysis and understanding of human actions in digital videos encapsulate multiple interesting research topics such as object detection (Burgos-Artizzu, Dollár, Lin, Anderson & Perona, 2012), semantic segmentation, motion analysis, etc. Hence, human action analysis has a broad spectrum of applications including intelligent surveillance, intelligent care.

Traditional methods in machine learning for human action recognition extracted visual features primarily based on human observations (Aggarwal & Xia, 2014). It is subject to a great amount of human experience and background knowledge. Most of these algorithms only performed well based on the exact dataset for a specific experiment. At present, there are a huge number of digital video footages available on the Internet, like YouTube. It is impossible to satisfy the demand to annotate all tags and extract the features only based on our human labor. We need machines to learn how to extract the characteristics of the human action.

Fortunately, the surge of deep learning methods in recent years has provided a solution. Deep learning algorithms generate feature maps based on artificial neural networks (Xiao, Xu & Wan, 2016). Deep learning has remarkable achievements in the field of computer vision, natural language processing and robotics. However, the applications of deep learning to human motion recognition are still in its early stages, as it has only emerged in the last few years. Meanwhile, human motion is relatively complex, the relevant motion analysis is affected by various determinants such as chaotic background, lighting conditions, image

acquisition, and insufficient pattern classes, etc. (Baccouche, Mamalet, Wolf, Garcia & Baskurt, 2011). Deep learning thus has a large room for developing human action recognition. Additionally, deep learning has extremely vital value to implement self-learning and transfer learning.

In this Internet era, we are experiencing exponential growth of various types of data (Schmidhuber, 2015), a typical example is the huge volume of digital videos generated from CCTV (i.e., Closed-Circuit Television) systems. This has provided the opportunity for developing applications of deep learning methods.

Moreover, high-performance computing (HPC) platform is the engine, which drives the development of deep learning (Shuai Zhang, Yao, Sun & Tay, 2019). Deep learning models require a large number of data samples. The previous HPC hardware was not enough to be applied for training hundreds of layers of deep learning models. In 2011, Google DeepMind accommodated 1,000 machines and 16,000 Graphics Processing Units (CPU) to train a deep learning model with up to a billion neurons (Holcomb, Porter, Ault, Mao & Wang, 2018). Now, with only a few GPUs, we accomplish the same attainments of calculations and iterations. Therefore, the rapid development of GPUs, supercomputers, cloud computing, and other HPC platforms makes deep learning as real reality (Chang, Fu, Hu & Marcus, 2016). The strong computing power is conducive to the rapid implementations and validations of deep learning algorithm. The accumulated experience imposes us to amend the models, and further advance the accuracy of the model.

The structural innovation of artificial neural networks is efficient and promising. Throughout continuous optimization of deep learning methods, visual objects on an image are evidently and distinctly identified. In the field of biometrics, complex environment such as orientation, lighting, angle of view, resolution will influence the recognition of human faces and actions. This requires that deep learning models have strong generalization ability for further optimization. An appropriate number of neural network layers will improve the performance of the deep learning model (LeCun, Bengio & Hinton, 2015).

Computer vision is a machine vision in which computers are able to assist our human

beings to recognize, track, and understand visual objects (Szeliski, 2010). It has been one of the hotspots in the field of artificial intelligence. Evidences show that 80%~85% human perception, learning, cognition activities are mediated by using our visual system (Parker, 2010). In artificial intelligence, the acquisition of information is particularly important. Machine vision is the main way to obtain information. Computer acquisition and processing of visual information is an important step to realize artificial intelligence.

Video footages have been taken into account in big data. At present, 70%~80% traffic data is from videos, which are thought as the big visual data (Koresh & Deva, 2019). The massive videos bring us new challenges in storage, transmission, processing, management, and other aspects as well as great opportunities, of course. These visual data was collected from a better resource as a dataset. The continuous expansion and updating of the data set has made a valuable contribution to future experiments. Thus, letting computers analyze and understand the visual data has become a shortcut path to observe and understand our natural world (Beijbom, 2012).

Deep learning has a powerful learning representation ability, which enables machine vision to perform better than human visual system in visual object recognition. Hence, computer vision can gain more visual information from digital images than our human visual system (Grauman & Leibe, 2011). In ImageNet Large Scale Visual Recognition Challenge 2016 (ILSVRC2016), the error rate of image recognition was nearly 2.9%, far lower than the error 5.1% of our human visual system (Russakovsky et al., 2015). The rapid progress of computer vision and the swift development of deep learning not only broaden the applications of our vision, but also deepen the research work related to deep learning (Junior, Musse & Jung, 2010).

Video retrieval and analysis are based on mobile phones, digital cameras, tablets, and other portable devices, which instantly shoot and transmit video footages in a convenient and speedy way. It makes the volume of visual data on the Internet growing at an exponential rate. By the end of 2015, YouTube had more than 500 video uploadings per minute. A bulk of these videos are currently annotated with textual tags labelled by uploaders (Mühling et

al., 2017). However, manual annotation is obviously tiresome and inefficiency. The subjectivity of these labels results in bias and leads to low accuracy of pattern classification and visual information retrieval (Dong & Li, 2018).

Therefore, human action recognition is necessitated to analyze the video content and automatically adjust the annotations, which is able to effectively lessen the diversity of human subjectivity and successfully promote the accuracy of video retrieval. In addition, human action recognition is able to automatically purge inappropriate videos that are not suitable for dissemination on the Internet.

In surveillance monitoring, indoor or outdoor surveillance cameras around the world constantly generate massive visual data (Shao, Cai & Wang, 2017). The uninterruptedly surveillance data from the networked surveillance cameras extremely relies on manually grouping. Spotting abnormal and suspicious behaviors, such as handbag or wallet abandon in real time, have become extremely tough from a mass of video footages. When surveillance staff is distracted, hazardous behaviors will not be found in time so as to take effective measures, which endangers public security and safety. Therefore, the abnormal action recognition assists security staff to fulfil the real-time monitoring work, which gains the tradeoff between costs and work efficiency (Guo, Luo & Yong, 2015).

Intelligent surveillance based on human action recognition is able to automatically monitor abnormal and hazardous events as well as risky and perilous human actions from the elderly (such as tumble, falls, etc.). Surveillance alarms will be sent out timely and accurately. This will cut down the delay of medical treatment (Chen, 2010). Similarly, intelligent surveillance is able to accomplish real-time assistance for patients, children, and the disabled through monitoring.

1.2 Research Question

The title of this thesis is deep learning methods for human action recognition. First of all, deep learning is the main way to carry through the recognition. Although traditional machine learning methods can also achieve the recognition of pictures, the bulk of the feature

selection work requires human effort. The current deep learning method has shown excellent performance in various fields such as image, speech and natural language processing. Deep learning has reached a very high level in image processing. Thus, we develop the research direction for human action recognition. Video footages, as we know, are made up of multiple frames, organized in a sequence.

With the development of human-computer interaction technology, a large number of methods for human action recognition have been proposed. It is easy for computing machines to gather the feature information of human actions. But the machines do not understand what it means and how to act like human. In other words, it is important to translate human understanding of digital images into machine-readable language. Human action understanding is able to be applied to a plethora of research fields, e.g., intelligent monitoring, unmanned driving, and so on. Therefore, in this thesis, we propose three deep learning methods to redeem the recognition of human actions.

The goal of this thesis is to recognize human actions effectively and correctly. Throughout a series of digital video processing, visual information in the videos will be recognized by using deep learning methods. In this thesis, we compare multiple deep learning algorithms to find out the best one with the highest accuracy. Therefore, the research questions of this thesis are:

How to recognize human actions in a given video?

In deep learning algorithms, deep belief network (DBN), convolutional neural network (CNN) and recursive neural network (RNN) are most widely employed. These algorithms are also the basis for implementing various recognition models. DBN is a generation model, by training the weights between its neurons, we can make the whole neural network generate training data according to the maximum probability.

As a kind of artificial neural network, CNN has become a research hotspot in the field of speech analysis and image recognition. Its network structure for weight sharing makes it resemble to biological neural network, which reduces the complexity of network model and

the number of weights. This advantage is much apparent when the input of the network is multi-dimensional image, the visual information can be directly imported as the input of the network, avoiding the complex process of feature extraction in the traditional machine learning algorithm.

RNN primarily is offered to deal with the time series problems. This is very important for natural language processing, speech recognition, handwriting recognition, and other applications. In a fully connected network or CNN, the signals of each layer of neurons only travel up one layer. Data sampling is an independent procedure. Therefore, it is also known as feedforward neural networks. In RNN, the output of the neurons is directly imported to the next step. The final result of the network is the output result of the inputs at the moment and all the past. RNN is regarded as a recurrent neural network which is to transmit visual information by using time series analysis.

Based on the reiterations of deep learning methods, in this thesis, we establish three different recognition models. They are CNN+LSTM, two-Stream Convolution and 3D CNN. The three recognition models will deal with the visual time series problems in human action recognition. Thus, our second question for human action recognition is:

Which algorithm has the best recognition accuracy?

The core idea of this thesis is human action recognition by using deep learning methods. Therefore, the methods need to be evaluated so as to achieve the best accuracy for human action recognition in this research project.

Finally, in this thesis, we work towards solving a research problem of human action recognition in a dark environment. The weak illumination has increased the difficulties of human action recognition, random noises affect the action analysis from these motion pictures. At present, most of the human motion recognitions are based on well-layout environment. Many of the datasets were also collected during daytime.

Considering the deep learning methods will be accommodated to recognize human actions. Hence, we ask the third research question:

Can the optimal algorithm accurately recognize human actions in completely dark environment?

In the field of human action recognition, much of the research has been carried out in well-lit daylight. For intelligent recognition, the dark environment at night is a big challenge. Intelligent identification technology should be able to serve people day and night, in order to bring real intelligence to people's lives.

As a prediction, we believe that Long Short-Term Memory (LSTM) has a strong ability to process the time series problems. Therefore, the experimental results of CNN+LSTM model may be higher. But accurate identification in completely dark environments depends on the results of experiments. This experiment will strive to improve the accuracy of the three algorithms so as to gain the best result.

Throughout the analysis of these three questions, we clearly understand the importance and implementation of deep learning for human action recognition. In this thesis, we will answer these three questions one by one by using the empirical way. Our conclusion is drawn by comparing the experimental outcomes. At the end of this thesis, the results of our experiments are compared and summarized, thus, deep learning in the field of human action recognition will be uplifted.

1.3 Contributions

The main contribution of this thesis is to find out the best method for human action recognition based on deep learning by analyzing and experimenting three kinds of models. At first, the appropriate dataset is selected for training the three algorithms. The evaluations are based on analyzing the optimal model. The experiments also include the identification challenge in a completely dark environment.

Human actions are split into four levels. The motions of our body parts belong to simple actions, such as waving, raising feet, drink water, etc. Individual behaviors comprise of simple actions, such as walking, running, jumping. The interactive behaviors include

combing hair, reading books, exercises, so on. Multiple human actions encompass those of multiple people, e.g., meeting, fighting, and hugging, so on. The research work at present for human action recognition is chiefly categorized into twofold: Action classification and action detection. Action classification is to assign a class label for a given video. Each video clip has only one instance of the behavior. Human action detection is to find out all actions within a given video which may have multiple actions. In this thesis, we primarily investigate the problem of human action recognition from the well-segmented videos related to human behaviors.

CNN is a core method in deep learning. In deep learning, we segment a given image into multiple regions, we extract visual features from each region. The visual features of these regions are concatenated together as a feature vector, which consists of the chief process of human object recognition. The three proposed algorithms in this thesis are all related to deep neural networks. Based on CNN, multiple methods are exploited to uplift the accuracy rates of human action recognition. In this thesis, we analyze each algorithm. Throughout comparisons, the best recognition algorithm is identified.

The development of human action recognition in the field of deep learning is overviewed through a large number of surveys. From the literature review, the experimental procedures for human action recognition are generally akin to each other. More and more deep learning recognition models were proposed. This will help us to have a deeper understanding of deep learning. The current research goal is aimed at finding the optimal algorithm. Human action recognition of deep learning method is applied to various environments and fields. In addition, a training dataset of human action recognition is also particularly important. We need to find the dataset which suits for the experiments to improve the accuracy.

Furthermore, multiple action recognition algorithms are implemented. The deep learning methods, i.e., Two-Stream CNN, 3D-CNN, and LSTM+CNN are proposed, these three algorithms are expounded on details. The recognition process includes six phases: Input original video, preprocess the video, extract visual features, establish and construct

deep neural networks, classify human actions by using the deep learning methods from digital videos, and output class labels.

Each algorithm handles the video timing problem differently. Sequence processing is particularly important in human action recognition. The same action may be performed in another order. Therefore, in this thesis, we propose the processing problems such as LSTM network and optical flow information between two adjacent frames. The recognition of human action is realized by adjusting the parameters.

In this thesis, the process of human action recognition includes the following steps: Video preprocessing, feature extraction, neural network construction, human action classification and result evaluation. Three approaches based on deep learning will be justified. The advantages and disadvantages of these algorithms are analyzed. By analyzing the experimental results, the best method will be determined.

The results will be presented by using tables and figures. The recognition accuracy of the three models and the experimental results of each algorithm are also presented separately. In addition, we analyze the recognition of each action in the conclusion part. We also show the accuracy of multiclass actions. The experimental results are unveiled by using the accuracy of these three methods as classifiers so as to gain the best one. The advantages and disadvantages of each method are probed. Conclusions will also be drawn on the challenges of dark recognition environments. This asserts that the proposed methods could be applied to a wide spectrum of research areas. At last, we summarize the thesis project, envision our work in the near future.

1.4 Objective of This Thesis

In the field of deep learning, an efficient and automatic method is urgently needed to classify and identify human actions in a large amount of video data. Although deep learning method for image recognition has made a wealth of research results, the time series analysis is still in the process of continuous improvement in the aspect of video recognition. There is still a room to improve the accuracy of human motion recognition. Hence, the human action

recognition methods based on deep learning have theoretical and practical values.

Typical examples of deep learning algorithms are R-CNN series, YOLO series, and SSD. In the aspect of human motion recognition, R-CNN series algorithm is the most broadly applied to object detection so far. As a kind of artificial neural network, CNN has become a research hotspot in the field of speech analysis and image recognition. The network structure with shared weights reduces the complexity of network model and the number of weights. This advantage is much evident when the input of the network is multi-dimensional image, so that the images are directly imported as the input of the network, avoiding the complex process of feature extraction in the traditional machine learning algorithms.

Based on CNN networks, three human motion recognition models are proposed, including CNN+LSTM, two-stream Convolution and 3D CNN. All these three models take CNN networks as the solidly core. Throughout time sequence analysis, the recognition of human actions in the videos can be achieved. By comparing the experimental results, the advantages and disadvantages of the three models are clearly identified.

CNN+LSTM algorithm consists of two main parts, namely, CNN network and LSTM network. Firstly, the experiment inputted the processed video data to CNN network to generate characteristic data. The feature dataset is then entered into the LSTM network in chronological order. Finally, the network is employed to motion-based spatiotemporal analysis.

Two-Stream CNN networks are based on dense and optical streams that deal with RGB images related to current and adjacent frames respectively by using Two CNN networks. Spatially, image analysis is carried out for video frames to obtain the corresponding feature set. In terms of temporal analysis, the feature is obtained by using optical flow changes of two adjacent frames. Thus, the features are fused together to obtain the detection network.

3D-CNN network is composed of a cube by stacking several consecutive frames, and 3D convolution kernel is applied to the cube. Through this structure, the feature map in the convolutional layer will be connected to multiple adjacent frames in the upper layer, so as to

capture motion information. This model obtains the information of multiple channels from the input frame, the final feature is the visual information of all channels combined.

The objective of this project is towards human action recognition from videos. The first step is to input video information. The second step is to preprocess the input video data and parse video frames. In this project, we embark on comparing three different algorithms and extracting the action features in the video. The next step is to establish the labelled dataset. By using the extracted features, we train our models, test the given samples, and recognize the human action. Finally, we export our classification results.

1.5 Structure of This Thesis

The thesis is structured as follows, in this section, we will introduce the structure in the order of chapters. In Chapter 1, we introduce the background and significance of this thesis, set forth the main research questions to be probed and the challenges to overcome the difficulties of human action recognition in a dark environment. The objectives and contributions of this thesis will also be clearly addressed.

In Chapter 2, the literatures will be reviewed, in-depth surveys based on the development of human action recognition will be investigated. Firstly, the deep learning algorithms are proposed, the advantages and challenges are analyzed. Secondly, various algorithms that have been practiced in intelligent recognition will be described. We analyzed the algorithms of R-CNN series, YOLO series and Single Shot MultiBox Detector (SSD), respectively. Finally, we summarize the state-of-the-art solutions of human motion recognition.

In Chapter 3, we bring in the research method. Firstly, the overall steps of this project are proposed. According to our research plan, three kinds of human action recognition algorithms based on deep learning have been proposed, including CNN+LSTM, Two-Stream Convolution and 3D CNN. The principle process of these three algorithms will be identified in detail. By evaluating the accuracy of the results, we get a motion recognition model with high accuracy and confidence.

In Chapter 4, computable methods and algorithms will be implemented. Moreover, the experimental results will be fully presented. The outcomes of each algorithm will be presented in the form of tables and figures. The resultant comparisons for these methods are offered.

In Chapter 5, based on the experimental results, the advantages and disadvantages of each method are justified. The computing settings, computational methods, and computable determinants of the experiments are detailed, the results are analyzed in depth. The important role of deep learning in action recognition will be discussed.

In Chapter 6, we summarize our research achievements and conclude this project, meanwhile, our future work will be envisioned. There are more challenges in experimentation, datasets and evaluation methods.

Chapter 2
Literature Review

Machine vision is a hot topic at present. In this chapter, we introduce the state-of-the-art methods in deep learning and machine vision. Similarly, this chapter summaries the past work in the field of human behavior analysis. The new methods for human action recognition are put forward.

2.1 Introduction

In recent years, a great deal of breakthroughs have been attained in the field of machine vision and deep learning. A plenty of methods for human action recognition based on deep learning have been explored and exploited (Pleshkova, Bekyarski & Zahariev, 2019). Compared with machine learning methods for human action recognition, deep learn approaches do not require a specific type of human experience and domain knowledge. Instead, human actions in a video are identified directly in the end-to-end way (Zhang, Quan & Ren, 2016). According to feature extraction methods, the approaches are grouped into two categories, e.g., human action recognition based on skeletons, human action recognition based on feature maps. Among the deep learning methods, spatiotemporal networks and the Two-Stream networks are the prominent ones in human action recognition. In these methods, CNN and RNN are most popular (Baisware, Sayankar & Hood, 2019).

A multimodal learning approach was proposed for the recognition and classification of human actions. Multiple deep neural networks are used for different modal information to dig out the multimodal characteristics of human action (Khan, Rahmani, Shah & Bennamoun, 2018). This is beneficial to the in-depth study of deep neural network. In 2017, 3D convolutional neural network (3D-CNN) and two-way long short-term memory network (ConvLSTM) were proposed which fulfil human action recognition by using Support vector Machine (SVM) classifier from multimodal and spatiotemporal information (Jing, Ye, Yang & Tian, 2017). A deep dynamic neural network (DDNN) was designed to implement action recognition from input data under multimodal framework, which extracts spatiotemporal features from RGB (color map) and RGB-D images (Wu et al., 2016). A scene-flow dynamic model was deployed to extract visual features from RGB and depth images, which were imported for training by using CNN networks (Wang et al., 2018). A 3D CNN network was taken into account to learn high-level features from the original images and calculated the position and angle of bone joint information. The two features were fused by using SVM for human action classification (Li, 2017). In 2018, CNN and RNN were integrated together to cope with the spatiotemporal information of human actions and achieved promising results.

The effect of different frame sequence correctly classified test data up to 96.66% (Russo, Filonenko & Jo, 2018).

Training deep learning models needs a huge number of datasets. In this digital era, everybody including adults or children is allowed to upload videos and photos to the Internet for sharing. This favorite in social networks thus has led to an explosion of video and image dataset. A vast number of images and videos provide the resources for resolving the problems in computer vision and deep learning (Morota, Ventura, Silva, Koyama & Fernando, 2018). For example, the HMDB-51 dataset and UCF101 dataset, both were collected from the Internet. The HMDB-51 is collected by Brown university in 2011. In computer vision, human action recognition plays an important role in our daily life, which is applied to video surveillance, robotics and unmanned vehicles, medical service, anomaly analysis, human-computer interaction, and other aspects (Baisware, Sayankar and Hood, 2019). These applications make us have extensive and important significance in the study of human motion recognition.

Specifically, the results of human action recognition provide evidence for video surveillance, but also accommodate clues for lessening the criminals by using deep learning methods at present (Räty, 2010). In terms of medical services, abnormal action of patients can be identified through motions and then reported to doctors and nurses. Otherwise, In addition, the human action recognition also has the ability to analyze the patient's movement to assist the diagnosis and rehabilitation (Li et al., 2017). In the aspect of analysis of abnormal human actions, the actions in the given videos are labelled, archived, classified, and predicted. If there is a large deviation between the actual action and the predicted action, the possibility of abnormal action is predicted so as to prevent the occurrences of incidents. This has taken great effects in monitoring the elderly and children as well as the disabled and patients in hospitals (Roh, Heo & Whang, 2019).

In terms of human-computer interaction, due to the demand for continuously ameliorating our life quality, people are enjoying more and more convenient services. The demand for life services has spurred the rise of the robotics (Rautaray & Agrawal, 2015). In

order to improve robots serves, human action recognition can help robots understand human behavior. The deep learning algorithms can effectively identify the motion information and translate it into machine-readable language. This has a great help in uplifting the understanding ability of robots. Thus, computers can also plan and respond interactively to human behavior.

Compared with manual operation, deep learning has the advantages of high efficiency and unlimited working time in human action recognition. It plays a pivotal role in intelligent identification which saves a lot of manpower and other resources. In robotics, if a deep learning model has been trained very well, the communications between human and computers (HCI) will be accurate, facile, and fluency (Rodríguez-Moreno et al., 2020).

2.2 Deep Learning

Deep learning is fundamentally based on Multilayer Perceptron (MLP) or deep neural networks. The most effective multilayer neural network is CNN. At present, CNN performs well in text, image, audio, and video signal processing (Strong, 2016). In terms of fundamentals, deep learning mimics the working mechanism of our human brains, especially CNN models. From a statistical point of view, deep learning as a neural work is to predict the class label of a given sample with a probability. Training a model from the given dataset and testing the trained model to predict class labels of new samples require that the test samples and the training samples have the similar pattern classes which have been labelled in the homogeneous datasets with the predefined tags (Sadoughi et al., 2018).

AI is defined as a computer to emulate our human intelligence. Machine learning enables computers to execute programs of various algorithms. Deep learning is a subfield of machine learning which enables computers to continuously learn from the training dataset as well as human experience. In the past decades, deep learning and machine vision have made a real reality (Wang, Zhang & Wei, 2019). The deep neural networks and graphics processing unit (GPU) or neural processing unit (NPU) have powerful capability for intensive computing. This provides a powerful hardware infrastructure for the development

of deep learning.

Compared to traditional machine learning methods, there are a vast number of advantages of deep learning approaches. Firstly, noisy information is able to be filtered out, deep learning algorithms classify patterns through training dataset, export class labels for solving the classification problems in our real world. Moreover, deep learning algorithms detect and recognize computable events based on valuable patterns, especially time series analysis. The unstructured or structured data is conveniently classified by using deep belief methods (DBM) and CNN. Although artificial neural networks emulate the mechanism of our human brains, deep learning models need time to gradually learn how to solve problems in the end-to-end way (LeCun, Bengio & Hinton, 2015) which means the imported data and its features will be independent on the design of neural networks.

Deep learning also encounters challenges, e.g., a large amount of data as a prerequisite is imported to deep learning models for model training, successful pattern classification by using deep neural networks is very tough to be collected. The overfitting in pattern classifications will dilute the model effectiveness. A well-trained deep neural network should not only carry out pattern classification, but also outperform others for a newly assigned task, namely, transfer learning (Voulodimos, Doulamis, Doulamis & Protopapadakis, 2018). With the rise of deep learning methods in machine vision, deep neural networks are extremely suitable for empirical applications, eventually serve the community in practical ways (Liu et al., 2017).

In the near future, more deep learning methods will be exploited in the areas such as remote diagnosis, intelligent surveillance, autonomous vehicles, robotics, and smart agriculture (Freeman, 2012). Because computing power is soaring at present, deep learning methods will have a long-standing impact on the field of computational vision. In addition, deep learning will greatly boost the methods in machine vision and digital image processing. Undoubtedly, intelligent learning will benefit from deep learning.

2.3 Deep Learning Models for Object Recognition

The successful applications of deep learning encapsulate face recognition, image-based question and answer problem.

In face recognition, we match the registered faces from a given database. If we are given two faces at the same time to determine whether they are from the same person, the proposed DeepID algorithm (Wong, Yap, Zhai & Li, 2019) based on the Linear Workflow (LWF) performed well for this problem, which utilizes a CNN network in deep learning. From the comparisons, the two faces are employed to extract visual features, respectively. The features are compared with each other to get the final decisive result. The latest DeepID-3 algorithm has achieved 99.53% accuracy in LWF, which is akin to the results of face recognition by using our naked eyes (Zhao, Tian & Sun, 2019).

In the problem of image-based question and answer, the topic has been developed since 2014, which is to give an image and need to ask a question, then a computer will answer the question by itself (Ren, Kiros & Zemel, 2015). For example, there is a picture of an office near the sea, then a question “What's behind the desk?” is asked, the deep neural network thus has the ability to answer the question as “chair and window”.

An example brought in the long short-term memory (LSTM) network, the recurrent neural network (RNN) was especially designed to have memory and forget units. The characteristic is to treat the output as the input at the next layer. It is considered to be much suitable for natural language processing and time series analysis. Because when we read an article or a newspaper, our understanding is based on the words we have read. The solution of image-based question-and-answer problem is based on the combination of CNN and LSTM together. The LSTM output should be the desired answer, and the input of next layer is the output of the proceed layer (Ma, Lu & Li, 2015).

Pertaining to visual object detection, the models from CNN family include R-CNN, Fast R-CNN, Faster R-CNN, YOLO (v2-v5), and Single Shot MultiBox Detector (SSD) which are very popular at present (Xie, Li & Sun, 2019).

2.3.1 R-CNN

Deep learning has achieved very remarkable progress in visual object detection. Region-based CNN (R-CNN) algorithm was given in 2016, the basic idea is to start with a non-depth approach. First, the detection object is a region selected from a given image. Then, R-CNN net determines the properties and location of the visual object from the input image, and filters and merges these regions (Chen, Liu, Tuzel & Xiao, 2016). The training is a multi-stage task, adjusting the convolutional neural network of the object region, making the support vector machine adapt to the function of the convolutional network, and finally learning the boundary box regression.

The non-depth method is used to segment the given image into regions firstly. Whilst conducting object detection and recognition, if sliding window method is employed, we expect the scanning windows have the same size and aspect ratio. Thus, a method selective search has been accommodated. In target detection and recognition, if the sliding window method is used, the scanning window will have the same size and aspect ratio. In this way, a selective search method is adapted. This method first removes areas that are not at all likely to be part of the target object. This process is very slow. For example, R-CNN is very slow, which requires 10 to 45 seconds to process an image (Shaoming Zhang, Wu, Xu, Wang & Sun, 2019).

R-CNN has made great achievements in object detection. However, it also has disadvantages, such as low efficiency, long time and a series of problems. As a result, the applications of R-CNN have not achieved a wide range of applications. In addition, R-CNN needs to extract visual features corresponding to multiple candidate regions in advance. This operation can take up a lot of disk space. For traditional CNN, the input map needs a fixed size, the deformation of images in the normalization process will lead to the size change of the image, which is fatal to the feature extraction of CNN. Each region proposal needs to enter CNN network for calculation. This leads to multiple iterations of the same feature extraction.

2.3.2 Fast R-CNN

Fast R-CNN has been meliorated in three aspects. Firstly, the speed of the test was improved. R-CNN algorithm overlaps with a large number of candidate frames in the image, resulting in a large amount of redundancy in feature extraction. Fast R-CNN solves this problem very well, which input the whole picture into the CNN network for feature extraction. Secondly, the training speed has been improved. The training requires more space. A large number of features are needed as training samples for classifiers and regressors in R-CNN. Fast R-CNN doesn't require extra storage (Hsu, Huang & Chuang, 2018).

In R-CNN, digital images are generally segmented and deformed to a fixed size before convolution operations. This has a huge impact on the subsequent feature selection (Li et al., 2017). The difference between Fast R-CNN and R-CNN is that Fast R-CNN does not have any restrictions on the data input, the key to implement the unrestricted pooling layer is the ROI pooling layer. The role of this layer is to extract a fixed feature representation for each input ROI area based on a feature map of any size, then ensure that subsequent classifications for each area perform properly.

Even though Fast R-CNN has been improved a lot compared with R-CNN, Fast R-CNN still has shortcomings. Because Fast R-CNN takes use of selective search, it is a time-consuming process, which takes about 2-3 seconds to extract candidate areas, while 0.32 seconds to extract feature classification, which makes it impossible to meet real-time application requirements (Qian et al., 2016).

2.3.3 Faster R-CNN

In Faster R-CNN model, object detection requires four steps, namely candidate region generating, feature extraction, classifier classification, and regression (Li et al., 2019). In R-CNN and Fast R-CNN, all these four steps are accomplished by using deep neural networks. This greatly upgrades the efficiency of traditional deep nets.

Faster R-CNN mainly consists of two modules. One is a deep convolutional network to

extract candidate regions. The other is a Fast R-CNN detector that takes use of these regions. Region Proposal Network (RPN) takes the image as input and generates output of the rectangular candidate region, and each rectangle has a detection score. RPN networks are different from ordinary CNN. RPN is a full CNN, whose inner part turns the full connection layer in CNN to a convolutional layer. Faster R-CNN is to detect and identify visual object in the proposal based on the extraction of RPN. The specific process is summarized into five steps. The first step is to enter the image. The second part is to generate candidate regions through region generation network RPN. The third one is to extract the features (Jiang & Miller, 2017). The fourth step is a classifier for pattern classification. The final step is for the returner to return and adjust its position.

This is a superfast version of R-CNN network which has seven frames per second for object detection and recognition. In R-CNN, the deep neural network decides where the visual objects are and what class the object is classified. The speed of R-CNN has been greatly accelerated (Girshick, 2015).

2.3.4 YOLO

YOLO addresses visual object detection as a regression problem. Based on a separate end-to-end network, the output is completed from the input of the original image to the object position and category (Shafiee, Chywl, Li & Wong, 2017). In terms of network design, YOLO is very different from R-CNN, Fast R-CNN, and Faster R-CNN.

Firstly, YOLO training and testing take place in a separate network. YOLO did not show the process of seeking a regional proposal. However, R-CNN and Fast R-CNN use separate modules to get the candidate box. The training process is therefore split into following modules. Faster R-CNN uses RPN convolution network to replace the selective search module of R-CNN and Fast R-CNN. It integrates RPN into Fast R-CNN network to obtain a unified detection network. Although RPN and Fast R-CNN share the convolutional layer, the RPN network and Fast R-CNN network need to be trained repeatedly during the model training (Shinde, Kothari & Gupta, 2018).

In the YOLO network, the input image is tackled with an inference that enables the position, class, associated labels, and corresponding confidence probability of all objects in the image to be detected. The R-CNN series chiefly reflect in the twofold of results: Object class label and location of bounding box.

YOLO has many advantages. First of all, it runs fast. YOLO takes object detection as a regression problem; the whole detection network is simple. Secondly, its background false detection rate is low. YOLO detects the overall information in an image during training and reasoning. Tests show that false detection rate of YOLO models for background images was less than half of Fast R-CNN. Finally, YOLO network has strong versatility. The successful rate of non-natural objects detection is much higher than that of R-CNN series detection methods (Lan, Dang, Wang & Wang, 2018).

YOLO is an object recognition and location algorithm based on deep neural network. It stands for You Only Look Once, which means, the class labels and locations of visual objects in the paradigm only have one shot, the relationship between speed and accuracy of object detection is perfectly balanced. Its biggest characteristic is that it runs fast and can be used for real-time system. YOLO has now been developed to version 5, but the new version is an evolution of the original one.

In YOLO framework, object recognition and detection are treated as two tasks. Firstly, we find an area of the image where visual objects are located, then we identify which objects are in that area. Although various methods based on CNN networks have achieved the results, the main problem is to find the location of the target object. The simplest idea is to iterate through all the possible positions in the image. For each region in different positions, the existence of an object is detected one by one, and the result with the highest probability is selected as the output. Clearly, this approach is inefficient.

YOLO creatively combines the two stages of candidate selection and object recognition into one, if we look at an image, we know which objects are there and where they are. In fact, YOLOv1 does not really eliminate candidates, instead, uses predefined candidates. That is, the images are divided into $7\times7 = 49$ grids. Each grid allows two bounding boxes to be

predicted. In total, we have $49\times2=98$ bounding boxes. It is thought of as 98 candidate areas, which roughly cover the entire area of the image.

The structure of YOLOv1 is very simple, convolution, pooling, and finally adding two layers of full connection. The greatest difference is that the final output layer uses linear functions as the activation function, because the position of a bounding box needs to be predicted, not only the probability of confidence, but also the size of bounding box. The structure of YOLOv1 network is composed of 24 convolutional layers and two full connection layers. The network entry is 448×448.

The main reason for scaling the original image is that in YOLO network, the convolutional layer is finally connected with two fully connected layers, and the fully connected layer requires a fixed size vector as the input. Therefore, the original image is required to have a fixed size, the size of YOLOv1 is 448×448.

When the input images are segmented into 7×7, the blocks of the output correspond to 7×7 grid of the input image. If we think $7\times7\times30$ vectors as 49×30-dimensional vectors, that means, the grids are attached with the input image, each of which will produce a 30-dimensional vector, which contains the classification probabilities of 20 visual objects, positions and confidence of two bounding boxes.

YOLOv1 supports the identification of 20 different objects (people, birds, cats, cars, chairs, etc.). There are 20 values that represent the probability of any these objects being presented at the grid location. Each bounding box requires four bounding boxes to represent its position (*Center_x*, *Center_y*, *width*, *height*), the two bounding boxes require a total of eight values to represent their positions. The coordinates *x* and *y* are normalized within [0,1] by using the offset of the corresponding grid, where *w* and *h* are normalized within [0,1] by using the width and height of the given image. The confidence of a bounding box refers to which it contains objects and their exact locations. A high degree of confidence indicates that there is an object that the location is accurate. The low confidence indicates that there may be no object or there is an object that may have a large position deviation. The advantages of YOLO models are:

- YOLO models have a fast detection speed, up to 45fps. This thanks to a network design that combines recognition and positioning, this unified design makes training and prediction easy to do end-to-end.

- The model has strong generalization ability and can be widely applied to other test sets.

- The background prediction error rate is low because the whole image is put into the network for prediction.

The disadvantages of YOLO models are:

- YOLO model has low accuracy.

- The detection accuracy of small targets and nearby targets is low.

- The detection results of small visual objects is not very good.

- The accuracy of border prediction is not very high.

The overall prediction accuracy of YOLO is slightly lower than Fast R-CNN. The reason is that the grid settings are sparse, each grid predicts only two borders. In addition, pooling layer will lose much details, which will have an impact on positioning.

In order to promote the accuracy of object positioning and recall rate, YOLOv2 improves the resolution of the training image and introduces the idea of anchor box in Faster R-CNN. The design of the network structure is improved, which makes the model easier to be trained. In YOLOv1, the full connection layer is used directly after the convolutional layer to predict the coordinates of bounding box. YOLOv2 takes use of the idea of Faster R-CNN to predict the deviation of bounding box. Thus, anchor boxes are introduced to predict bounding boxes.

YOLOv2 proposes a new classification model DarkNet-19, which contains 19 convolutional plus 5 max pooling. This model replaces YOLOv1 with 1×1 convolutional layer. The 1×1 convolution layer mainly takes use of 3×3 convolution and doubles the

number of channels after pooling. Average pooling is used instead of full connection for prediction classification, 1×1 convolution compression feature is utilized between 3×3 convolution. The batch normalization is used to improve stability, accelerate convergence, and regularize models.

At present, most detection models are pretrained as the main part of the model based on the ImageNet dataset. The classification model basically takes an image of size 224×224 as input. The relatively low resolution is not conducive to the detection of the model. Therefore, YOLOv2 uses 448×448 as input based on the ImageNet dataset to the fine-tuned classification network, which makes the model suitable for high-resolution input before finetune detection.

YOLOv2 used fine-tuning as the input size of the network after several iterations. During the training, the new input image size will be randomly selected after every 10 iterations. Since the downsampling ratio used by YOLOv2 network is 32, a multiple 32 downsampling is applied to adjust the size of the input image. The minimum image size for training is 320×320, the maximum image size is 608×608. This allows the network to be adaptive to multiple scales for the input.

YOLO models have made a series of improvements from YOLOv1 to YOLOv5. Throughout maintaining classification accuracy, YOLOv2 improves target positioning accuracy and recall rate. YOLOv2 is adaptive to various sizes of input images, the accuracy and speed of object detection are dependent on its needs. A joint training method was applied simultaneously to object detection and classification.

YOLOv2 still could not solve the problem of overlapping objects. YOLOv3 continues to make minor improvements to YOLOv2. By using a residual model, a new network DarkNet-53 with 53 layers was proposed. The structure of FPN (i.e., feature pyramid network) was used to obtain the feature map of three sizes.

According to DarkNet-53 network structure, the depth of YOLOv3 is pushed to 106 compared to YOLOv2 network. YOLOv3 refers to ResNet and FPN network structure. At

the same time, this model takes use of multiscale prediction to compensate for the lack of fineness of the initial partition of 13×13 grid. In addition, YOLOv3 network still adopts data enhancement, batch normalization, and other operations in YOLOv2.

YOLOv2 network structure has a special passthrough layer. Suppose the size of the extracted feature graph is 13×13. The function of the transformation layer is to stack the previous 26×26 feature map and the 13×13 feature map of this layer, then merge them. The fusion feature map was then used for object detection. The purpose of this method is to enhance the accuracy of the algorithm in detecting small targets.

YOLOv3 takes advantage of upsampling and fusion, which mingles 3 scales (13×13, 26×26 and 52×52) and conducts independent detection based on the fusion feature map of multiple scales. Finally, the detection results of small targets are improved significantly.

In 2020, YOLOv4 has an average accuracy 43.5% based on Microsoft COCO dataset with the frame rate at 65FPS, an improvement of more than 10% over YOLOv3. YOLOv4 splits the target detection framework as backbone, neck, and head. The backbone refers to subnet for feature extraction over a pretrained network, such as VGG16, ResNet-50, DarkNet53, or a lightweight network like MobileNet and ShuffleNet. The neck refers to feature enhancement module which enhances the backbones, such as SPP, SAM, FPN, PAN, ASFF, SFAM. The head means object detector, which exports the desired result of object detection.

In general, it is also a challenging subject to apply YOLO methods to human action recognition. It is also very helpful for deep learning algorithms.

2.3.5 SSD

In Single Shot MultiBox Detector (SSD) network, an entire image is required as input, along with a label of bounding box as the ground truth for each visual object. Amid the convolutions, a small number of default bounding boxes for each object in the feature maps with multiple sizes are verified. Regarding each default box, shape offset and confidence

probability are predicted for the visual object (Liu et al., 2016). During model training, the default boxes are compared with the ground-truth bounding boxes. SSD is a deep neural network based on feedforward convolutional neural network, which generates a series of fixed-size boundary boxes and scores for these boxes. Thus, a non-maximum suppression algorithm is followed to obtain the final prediction (Li & Zhou, 2017).

In SSD, each layer in the neural network generates a fixed set of predictions by using a convolutional kernel. The size of a kernel is $m \times n \times p$, the basic element is employed to predict potential parameters which is a small convolution kernel $3 \times 3 \times p$. SSD produces a score for the object detection and a shape offset related to the default bounding box. The offset output is measured by using the default box position and the eigen graph position (Ning, Zhou, Song & Tang, 2017).

SSD is an updated version of YOLO nets which learned from the decline of YOLO accuracy. SSD has attained 58 frames per second with an accuracy 72.1%. The speed is 8 times faster than that of Faster R-CNN but reserves the similar accuracy.

2.3.6 Object Tracking

The focus of object tracking is on visual objects of interest in the frames of a given video. No matter how the object is spined and shaken, it will be tracked even if occlusions happen (Bertinetto, Valmadre, Henriques, Vedaldi & Torr, 2016).

Deep learning has a significant impact on object tracking. Deep-track algorithm (Hanxi Li, Li & Porikli, 2015) is regarded as the first online work that took use of deep learning for object tracking. The performance of this algorithm was superior to all others. Nowadays, more and more deep learning tracking algorithms are put forward. For example, the hierarchical convolutional feature algorithm achieves the best performance based on the given training data, however, it is not online-based deep learning network, alternatively, a large network is used for model pretraining. The big network was trained by using the location of a given visual object before object tracking. This takes the advantages of the merits of deep learning. As a result, visual objects are tracked in the rate of 10 frames per

second.

The latest object tracking algorithms are based on hierarchical convolutional feature, multi-domain convolutional neural network (MDNet) was set forth (Jung, Son, Baek, & Han, 2018). It is a collection of the previous two depth algorithms. Firstly, there is a model training while offline, the training is applied for object tracking in videos (Lu, Chen & Li, 2017).

2.4 Human Action Recognition

Human action recognition is an important direction of video content understanding. At present, visual object recognition in computer vision has become mellow, but the action recognition still has not achieved the ideal outcomes. Compared with image recognition, human action recognition has temporal correlation, such as “open the door” and “close the door”, “throw the ball”, and “catch the ball” Human action recognition not only needs to model spatial semantic information but also necessitates to model temporal information. This is where object recognition is much intricate. In addition, human action recognition often needs to be computed based on multiple frames of images, its computational capability is higher than that of image recognition.

2.4.1 Statement of Human Action Recognition

In recent years, a great assortment of breakthroughs has been made in the field of machine vision. Human action recognition methods were developed based on deep learning (Popoola & Wang, 2012). Compared with the traditional human behavior recognition methods, the deep learn-based approach does not require a specific type of segmentations to distinguish between different behaviors. Instead, human actions are encoded directly and then adjusted.

According to the feature extraction methods, human actions are generally grouped into skeleton feature-based, depth image-based, and mixed feature-based human behavior recognition methods. Among the deep learning methods, spatiotemporal network and dual-stream network are widely used (Nguyen, Fookes, Ross & Sridharan, 2017). The method of

this thesis is also based on deep learning. In these methods, CNN and RNN are employed, the relevant theoretical basis will be expounded in detail in Chapter 3 of this thesis.

A method based on skeleton characteristics (Si, Chen, Wang, Wang, & Tan, 2019) was proposed to project 3D bone joints onto three orthogonal 2D planes and combine 3D depth information. The linear function is used to construct images with distance information to recognize and classify human behavior. The coordinate information of bone joints is converted from Cartesian coordinates to cylindrical coordinates (Du, Fu, & Wang, 2015). The relative positions of the bones and joints are selected manually to build three gray images. The characteristics were sent into VGG net for human action classification. Liu et al. proposed direct input of bone joint images. The spatiotemporal information is input into the improved CNN network model for training and classification. However, this approach has drawbacks. Each behavior has a fixed number of skeletal sequence inputs. The multilayer neural networks are employed to automatically learn the spatiotemporal information of bone models (Agrawal, Girshick, & Malik, 2014).

A multimodal learning approach (Rahmani & Bennamoun, 2017) was set forth for object recognition and classification of human action through isolated networks. 3D-CNN and ConvLSTM was proffered to train multi-odal spatiotemporal information and complete the behavioral recognition task by means of SVM classifier (Li, Zhang, & Shen, 2017). A deep dynamic neural network (DDNN) was designed to realize gesture recognition under multimodal input data, which extracts spatiotemporal information from RGB and RGB-D images (Wu et al. 2016).

A scene-flow dynamic model (Menze & Geiger, 2015) was propounded to extract features from RGB and depth images. A Persistent RNN (Shi & Kim, 2017) based on privileged information for human action recognition was suggested. A three-dimensional deep convolutional neural network (Liu, et al., 2018) was designed to extract high-level features from the original depth images, and calculate the low-level features of the bone joint position and angle information. The two features are fused for action classification. CNN

and RNN networks are integrated together in a project to deal with the spatiotemporal information of human actions, and achieved satisfactory results (Fan, Lu, Li & Liu, 2016).

Depth maps encapsulate rich representation, this method is very effective in human action recognition. However, depth images are not suitable for the input of traditional CNN. In order to solve this problem, visual features were extracted from frame sequences by superimposing shapes and features. This information is then converted to texture information by encoding the depth information. The textural information is taken used for large-scale image recognition and training. Relevant experiments show that this method achieved satisfactory results (Yang, Zhang & Tian, 2012).

Digital universal meter adjustment (Wang et al. 2015) has been brought to a pseudo-RGB image and transform its spatiotemporal action information into texture information which was trained and identified by merging three ConvNet networks. A human action model with an invariant view of deep sequence learning (Uddin, Khaksar, & Torresen, 2017) was proposed. The method is to input each depth image into a specific CNN net to extract advanced features. The human action in the unknown images is transmitted to the model for training and classification. A new RGB-D image recognition framework (Liu, et al., 2008) was designed. By calculating the position deviation of 3D bone joints, the video frames take use of the spatial independent property of joints in the word bag model to complete vector migration and identify human action.

In other experiment, three types of dynamic depth images are constructed. In other words, dynamic depth images, dynamic depth conventional images, and dynamic depth motion images are employed to extract the features of human actions in depth image sequence (Plagemann, Ganapathi, Koller & Thrun, 2010).

2.4.2 Methods of Online Human Action Recognition

Most methods of human action recognition rely on a public dataset of already classified actions. These training sets consist of one movement per video. Therefore, when the detection video contains multiple actions, the detection results will be affected. In addition,

the algorithms that can effectively detect low-latency actions are needed so that the changes can be responded instantly (Koo, Lim & Kwon, 2008). For example, when dangerous behaviors occur in public monitoring, the system should immediately alert. When an elderly person falls, a medical care robot could provide assistance immediately.

In order to locate human action, most of the early studies took use of probabilistic thresholds to detect the boundaries or key behaviors of each action. For example, a method was proposed to identify the transitions between two continuous actions in the training stage and complete the real-time classification by comparing the likelihood probability in minimal entropy martingale measure (MEMM) model (Parisi & Wermter, 2013). There are multiple ways to segment behavior based on clip or frame marking methods. Dynamic time distortion and dynamic frame distortion simultaneously were utilized to segment and classify human actions (Isaacson & Shoval, 2006). In the method, a label is assigned for each frame of videos after comparing it with the samples. The changes of class labels between successive frames represent the beginning or end of the human actions.

An action detector based on sequential maximum edge was developed (Fernández-Llatas, Benedi, García-Gómez, & Traver, 2013). This method is able to detect each action in a continuous video frames including multiple action. The action slices are treated as the input for classifier training. The gathered fragments of daily life are classified into a number of textual words (Liu, Shahroudy, Xu, & Wang, 2016), which was used to represent the class of actions and reflect the spatial relationship amongst actions. According to each continuous action, the segmentation is detected.

Sliding windows is also applied to real-time action recognition. With a sliding window, the video stream is usually split into a set of overlapping segments. The action is then corresponded to each segment. An action label is then associated to each video frame. The predicted label of human action was calculated by utilizing the sliding windows so as to determine the boundaries of each action in the video stream. However, the calculation efficiency of this sliding window is very low, meanwhile it is very hard to find the right window size (Yao, et al., 2011).

An improved 3D-CNN network was proposed (Ji, et al., 2012). Gesture classification can be carried through simultaneously from continuous depth, color, and IR data sequences. The connection time classification (CTC) was proposed for gesture recognition classification, without the need for specific segmentation (Barros, Parisi, Jirak, & Wermter, 2014). The problem of behavioral timing localization of multistage convolutional neural networks was resolved (Sermanet, Chintala, & LeCun, 2012). These include action recognition and time boundary detection.

Recurrent neural network (RNN) and its associated networks (such as LSTM) have been widely applied in human action recognition. An action recognition based on RNN was proposed (Tu, et al. 2018). Firstly, a detector based on Fast R-CNN network was employed to segment the continuous actions into separated ones. Then the segmented action framework is identified by fusing the multimodal characteristics of the double-flow RNN (2S-RNN). A regressive neural network (Gopi, Lakshmanan, Gokul, & KumaraGanesh, 2006) was proposed for the end-to-end multitask joint classification, which simultaneously performed well in human action recognition.

Chapter 3 Methodology

In this chapter, we firstly put forward the overall scheme and structure of human action recognition. The methods are explained thoroughly. The parameters and details of the algorithm are analyzed. The implementation is based on accurately recognizing human actions in given videos. This chapter is the key one of this thesis.

3.1 Research Design

The ultimate goal of this thesis is to implement human action recognition from the given videos. We sparse the footages and imported the frames as the input data. Three deep learning methods are applied to generate feature maps for human action recognition. Throughout network training, we recognize human actions and finally export the class tags. As shown in Figure 3.1, the overall pipeline of this design is shown as follows.

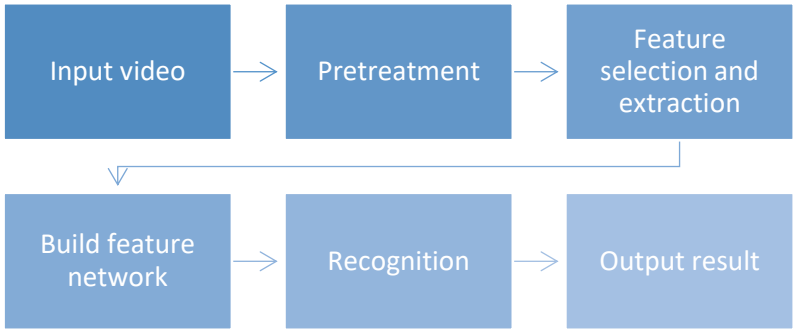


Figure 3.1 The pipeline of human action recognition

The first step is to input video information. The second step is to preprocess the input video data and parse video frames. The third step is to analyze the characteristics of each movement and extract them. The fourth step is to build the feature network of the action with the extracted features. The fifth step is to use the feature network to identify the actions. Finally, we output the result.

3.2 CNN+LSTM

3.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a kind of feedforward neural network, which is chiefly composed of input layer, convolutional layer, pooling layer, full connection layer, and output layer. The convolutional layer and pooling layer are alternated. The activation functions of the CNN are different with various structures (Wigington, et al. 2017).

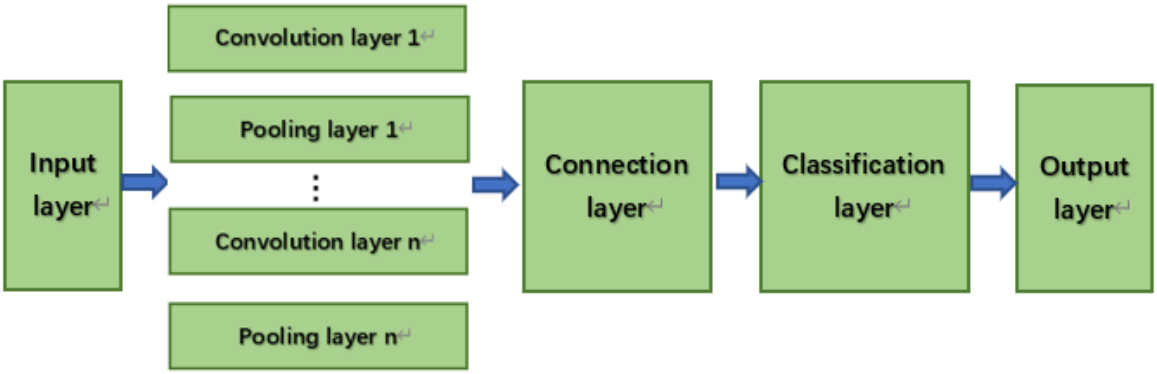


Figure 3.2 Basic structure of CNNs

The convolutional layer of CNN contains one or more feature planes. Each feature plane is composed of neurons with a pattern, the neurons in the same feature plane share the weights. The shared weights are related to the convolution kernel, the reasonable weights are obtained by optimizing the parameters of the network through model training. By extracting local features and synthesizing them, the CNN network not only obtains global features but also reduces the number of neuron nodes. At this point, the number of neurons is still very large, by setting the weight of each neuron equally, the number of network parameters will be greatly reduced.

On the first m convolution layer, its output is y_m, the output of the first k convolution kernels is y_k^m,

$$y_k^m = \delta(\sum_{y_i^{n-1} \in M_k} y_i^{m-1} * W_{ik}^m + b_k^m), \tag{3.1}$$

where $\delta(\cdot)$ is activation function, M_k is a layer of characteristic collection, W_{ik}^m is the convolution kernel, $*$ is convolution. b_k^m is bias or offset. The pooling layer follows the convolutional layer to reduce the dimension of input data and accelerate the convergence of the network training.

The other is to remove redundant features and prevent network overfitting. Each neuron in the full connection layer is connected with all neurons in the previous layer. Throughout

the full connection layer, all local features extracted in the previous layer can be integrated to form the overall features. Each neuron in the full connection layer operates with an activation function, which is then transferred to the output layer.

3.2.2 Long Short-Term Memory

In recurrent neural network (RNN), it is impossible to distinguish the importance of the information at each layer, which results in the useless information being stored in the memory unit. However, the truly valuable information is squeezed out (Ma & Hovy, 2016) based on RNN. Therefore, Long Short-term Memory (LSTM) was proposed. By using memory unit and gate mechanism, the problems of gradient vanishing and gradient exploding occurs in the training process of RNN were effectively overcome. Each unit of LSTM network contains memory unit, input gate, forget gate and output gate.

The structure of LSTM is shown in Figure 3.3, which is applied to control the amount of information updated by using memory unit. That is, how much information about the current state of the network needs to be saved as internal state. f_t refers to forget door, which is used to control before how much memory unit C_{t-1} is saved to the memory unit C_t. O_t stands for the output, which is used to control C_t namely, how much output to the next hide status h_t, h_{t-1} is the hidden layer at time t-1.

LSTM memory unit receives the current information at time of *t*, external state h_{t-1} and internal state C_{t-1} at time *t*-1. X_t and h_{t-1} are the input. How to calculate the input gate, forget gate, and output gate is shown as eq. (3.2),

$$\begin{cases} i_t = \sigma(W_i x_t + v_i h_{t-1} + b_i) \\ f_t = \sigma(W_f x_t + v_f h_{t-1} + b_f). \\ o_t = \sigma(W_o x_t + v_o h_{t-1} + b_o) \end{cases} \tag{3.2}$$

The memory unit C_t is calculated through different gating data memory and forget update as shown in eq. (3.3)

$$\begin{cases} \widetilde{c_t} = \tanh(W_c x_t + v_c h_{t-1} + b_c) \\ c_t = c_{t-1*} f_t + \widetilde{c_t} * i_t \end{cases}. \tag{3.3}$$

The updated hidden state is calculated as

$$h_t = o_t \otimes \tanh(c_t), \tag{3.4}$$

where σ is sigmoid functions, w is weights, v is weight, b is biase, $*$ is the operator for dot product.

LSTM memory unit receives X_t, h_{t-1}, in the form of σ activation function, produces control input respectively by using input gate (i_t), forget gate (f_t), the output gate (O_t), σ is sigmoid function. LSTM is based on the current moment, updating the memory unit C_t and generate the current state of the moment output h_t, as subsequent additional input, $t+1$ time and again.

In LSTM, $\tilde{c_t}$ and i_t are used to update the internal state, if i_t tends to 0, $\tilde{c_t}$ is very small amounts of information that was saved to the internal states; on the other hand, this method is used to decide how much the information will be saved. Forget gait f_t controls C_{t-1} and updates C_t. Throughout the training process, LSTM network optimizes weights w, v and offset b, makes sure that unit value c spontaneously is adjusted in temporal domain.

Human actions contain not only spatial information but also temporal information. CNN networks cannot fully use of the temporal information of the videos. Although the recognition method based on 3D CNN and the method based on dual-flow CNN take use of the timing information simultaneously, the main consideration is the short-term action information of adjacent frames.

If temporal information between video frames is not taken into account, misclassification easily occurs. For a simple example, “Pick up the water glass”, “Drink water”, and “Put down the water glass” as three video frames. We expand the three frames out of order, regardless of the sequence of the video frames. It is tough to say whether current action is “picking up the glass” or “putting it down”. The output of LSTM is determined by using the combined action of the current input and the previous historical output. Temporal information is used to represent a sequence. The CNN+LSTM model of this thesis is shown in Figure 3.3.

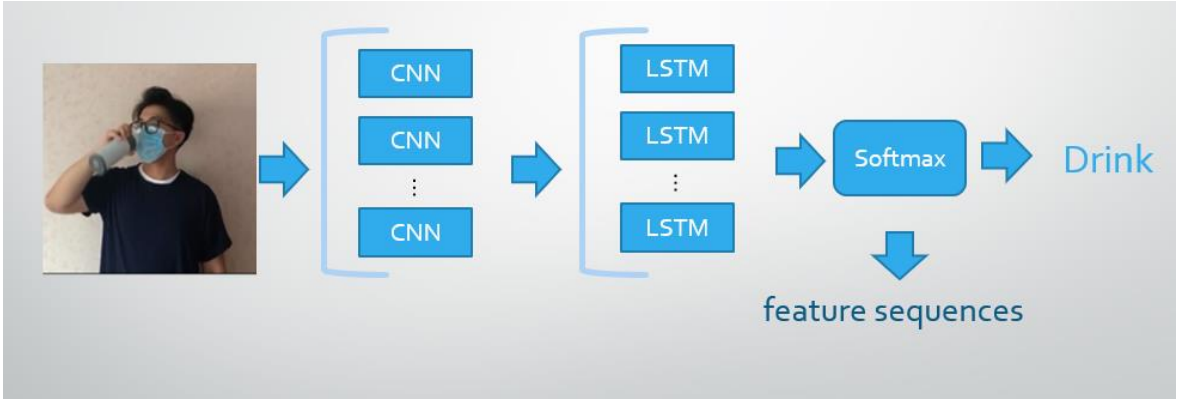


Figure 3.3 The structure of CNN+LSTM network

First of all, the test video is split into frames to form an image dataset. This dataset is employed as the input of single-channel CNN+LSTM for CNN pretraining. The training results are stored in a feature dataset and the feature map is generated. The characteristic dataset is thus input into the LSTM network as the input data. LSTM network is used to train the sequence data. After model training, the fixed network parameters of CNN are used for human action recognition.

In each video sequence, eight video frames are extracted at an interval as the input. The spatial features are extracted from the given video frames. The spatial feature is input into the LSTM to explore the temporal relationship among the video frame sequence. We obtain the fixed network parameters after model training. We complete the learning and training process of single-channel CNN+LSTM model. At the test stage, eight video frames extracted with equivalence distance are taken as the input data of the CNN+LSTM model. After spatial feature extraction and temporal relationship classification, the predicted output value of LSTM is taken as the final classification result.

3.3 The Two-Stream CNN Network

Two-Stream CNN employs RGB (color map) and optical flow to construct a CNN network. The core idea is to use two CNNs to tackle RGB values of video frame and dense optical flow of adjacent frame. The Two-Stream CNN models were trained respectively. The classification and recognition results of different network models are fused. Multimode

fusion and information complementation of different types of data are implemented (Li, Wu, Zhao, Cao & Tang, 2018). The dual-flow CNN network architecture is shown in Figure 3.4.

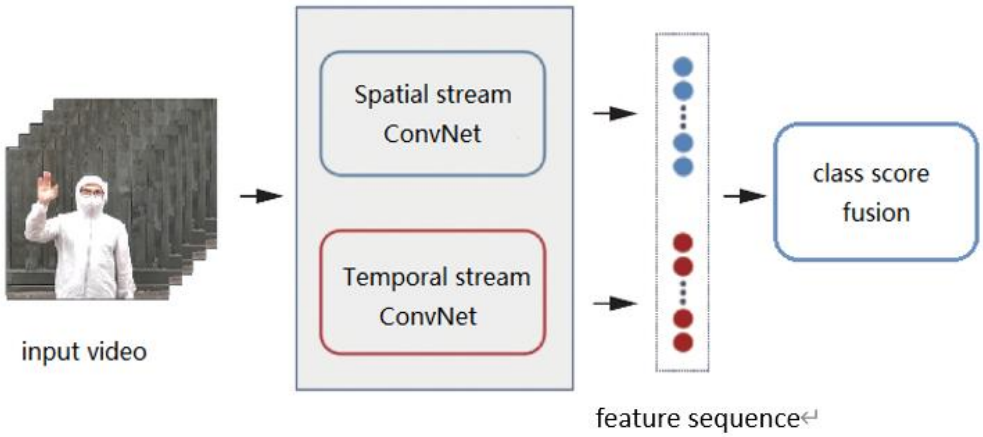


Figure 3.4 The structure of the Two-Stream CNN Network

The optical flow is calculated between adjacent frames in the videos. The optical flow reflects the short-term action changes between adjacent frames. In temporal domain, the motion is expressed in the form of relationship between video frames. In spatial domain, a single frame carries the scene and object information depicted by the video. Its appearance is a useful clue, because human actions are clearly related to a particular human body. The classification results of the two networks were obtained by using softmax function and multiclassification linear SVM fusion as the final classification results of the two CNN models.

Convolutional neural network (CNN) is one of the branches of deep learning. It is essentially a multilayer perceptron which is composed of input and output layers and multiple hidden layers. The difference is that CNN is use of convolution operation instead of one-dimensional multiplication and adopts the strategy of local connection and weight sharing between front and rear neurons.

In traditional artificial neural network, each neuro needs to be connected with the whole network. Local connection greatly reduces the number of parameters under the premise of

fully extracting the features. Weight sharing means that the same neuron shares a set of neuron parameters when extracting different regions and images of the same image. Multiple neurons can extract different features of an image. This strategy also enables CNN to extract features effectively in the case of image translation, etc. A policy for local joins and parameter sharing, which reduces the number of weight parameters required by the network, and also slashes the complexity of the model and prevents overfitting. Therefore, the CNN is widely employed in various research projects related to digital image and video processing.

3.3.1 Basic Structure

In addition to the input and output layers, a CNN has an intermediate hidden layer, through which data is passed to each hidden layer. After extracting features and data dimension reduction layer by layer, the error between the target result and the current network result is calculated by the output layer. Error back propagation updates network weights and parameters. In order to complete a model training process, convolution and pooling operations are performed in the hidden layer.

The image is input to the network through the input layer, the convolution operations are carried out based on the first layer through multiple fixed-size convolution kernels. Each convolution kernel is regarded as an image filter. The convolution kernel will traverse the entire input image so as to produce a characteristic image. Each convolution kernel has different weights to extract different features of the same image. After passed through the first convolutional layer, multiple feature maps representing different information are output. The traversal process usually has a step size of 1. Various features of digital images are extracted by using convolution kernels. The convolution operation is shown in Figure 3.5.

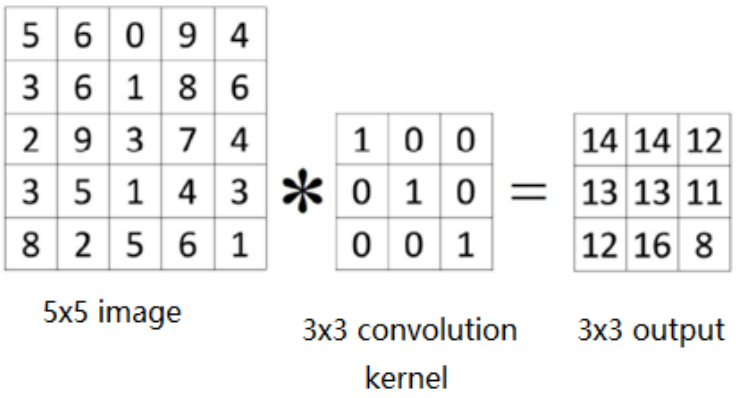


Figure 3.5 An example of convolution operations

The data dimensionality is reduced based on preserving the effective information of the given image as much as possible. Pooling is similar to convolution computation which is also a filter that goes through the entire feature map. But the step length of the pooling is usually as same as the filter size. The calculation takes the maximum or average value of the current point and its neighborhood in place of the entire region. It plays the role of data reduction. The pooling process is shown in Figure 3.6. The convolutional layer and the pooling layer are connected. Multilayer superposition is used for feature extraction and dimensionality reduction.

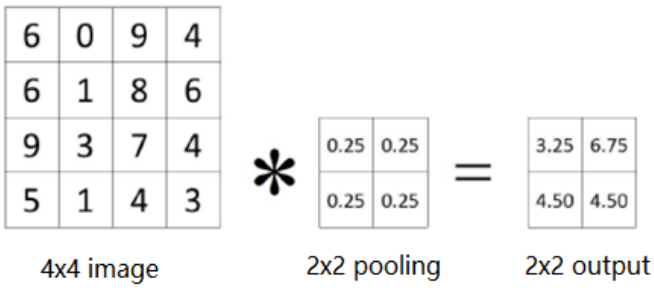


Figure 3.6 An example of pooling operations

Convolutional layer and pooling layer extract and simplify image features. The full connection layer is responsible for synthesizing the extracted features to obtain the global features and playing the role of data dimension reduction. The full connection layer takes the previous layer of neurons as input. The local features extracted by convolution layer and

pooling layer are integrated through nonlinear combination. The output results of the full connection layer will be input into softmax, SVM and other classification functions or classification in the classifier. The classification results are obtained based on the image features extracted by the network. In the network training stage, the classification results need to be used for backpropagation through calculating the loss function and the class tag or label after the classification.

3.3.2 Model Training Process

In the parameter optimization process of CNN, backpropagation (BP) algorithm is used to update parameters at each layer of the neural network, which includes two stages of forward propagation and backpropagation. During network training, an initial value should be set for all network parameters before the first forward propagation. After the data is imported into the network, it goes through the convolutional layer, pooling layer, and full connection layer, each layer of neurons takes the output of the previous layer as an independent variable. The dependent variables will be calculated layer by layer as the independent variables of the next layer of neurons. The output of the last layer is fed into the classifier to obtain the classification result.

The loss function is used to calculate the errors between the forward propagation classification result and the class labels after classification. The gradient descent is transmitted to each layer of the network. Updating parameters in the direction minimizes the loss function. Until the error is less than a fixed artificially set threshold, we finish the training. The computational process of CNN model training is stated as follows.

In the CNN, the convolutional layer, the pooling layer, and the full connection layer all contain corresponding weight parameters. We need to update the weights. It is preferred to update the parameters during model training.

Let the current input be layer k, the network consists of layer K, the output of the current hidden layer is shown as eq. (3.5) and eq. (3.6)

$$u^k = W^k x^{k-1} + b^k \tag{3.5}$$

$$x^k = f(u^k) \tag{3.6}$$

where W^k and b^k respectively represent the weight parameter and bias value of neurons in layer k, x^{k-1} represents the output of the previous hidden layer, x^k is the output of the current layer , u^k stands for the output of layer k neurons, f is the activation function such as sigmoid function, Rectified Linear Unit (ReLU) function, etc.

In machine learning, ReLU functions are often used to classification problems. After calculated the hidden layers of each layer, the prediction results containing high level semantics and multiple features are obtained and compared with the real class tag. A training dataset with C class and N samples. For each individual sample n, the training error can be expressed by eq. (3.7)

$$E^n = \frac{1}{2}\sum_{l=1}^{c}(t_l^n - y_l^n)^2, \tag{3.7}$$

where t is the true tag value of the sample corresponding to the number n sample in c classes, y is the network estimation value corresponding to the number n sample in c classes. The total error is the sum of the errors of each sample. The total error is shown in eq. (3.8)

$$E^N = \frac{1}{2}\sum_{n=1}^{N}\sum_{l=1}^{c}(t_l^n - y_l^n)^2 \ , \tag{3.8}$$

where error E is the derivative of the node b of each hidden layer. The derivative will be calculated by gradient descent method for parameter updating.

Backpropagation calculates the partial derivative layer by layer with chain rule. It updates the parameters of each layer along the direction of negative gradient. The sensitivity of the error to node b of each hidden layer reflects the rate of error E with node b. The derivative E with respect to b is shown in eq. (3.9)

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial u}\frac{\partial u}{\partial b} = \delta. \tag{3.9}$$

The sensitivity of the last layer of the network (the output layer) is obtained by differentiating the total training error of the network with respect to the input as eq. (3.10)

$$\delta^K = f'(u^K) \cdot (y^n - t^n), \tag{3.10}$$

where *f* as the activation function of the derivative, '·' represents the dot product of each element. If the input layer propagates back to the remaining layer, it goes to the *K* layer. The sensitivity of the layer to input nodes is expressed in eq. (3.11)

$$\delta^k = (W^{k+1})^T \delta^{k+1} \cdot f'(u^k), \tag{3.11}$$

where δ value of each layer of neurons is scaled to complete the update of the weight parameters of the neurons in that layer. It is going to be a vector representation. That is, the derivative of the error to each weight matrix of the layer is the cross product of the input value of the layer and the sensitivity. After partial derivative of the total training error to the weights of each layer is multiplied by the learning rate, the weight parameter update of the network is completed, as shown in eq. (3.12)

$$\Delta W^K = -\eta \frac{\partial E}{\partial W^k}. \tag{3.12}$$

This represents the learning rate and is a super parameter to control the change speed of the loss function. The lower the learning rate, the slower the loss changes. By setting an appropriate learning rate, the network keeps training and updating parameters. Until the loss function reaches a local minimum, hence, the model is optimized.

3.3.3 Network Testing Process

CNN model obtains the network parameters that make the loss function locally optimal. The test data is imported into the network with fixed parameters after model training. The classification results are compared with the test output. The average classification error of all test samples was calculated, which is used to measure the classification performance of network models.

3.4 3D Convolutional Network

In 2D CNN, convolution is applied to 2D images, the features are calculated only from spatial dimension. By using video data to analyze a problem, we expect to capture action

information encoded in multiple consecutive frames. Therefore, it is proposed to carry out 3D convolution in the convolution of CNN in order to calculate the spatial and temporal dimension characteristics. 3D convolution is the stacking of several consecutive frames to form a cube. Then we apply the 3D convolution kernel to the cube. With this structure, the feature map in the convolutional layer is connected to multiple adjacent frames in the previous layer. In order to capture action information, the position of a feature map is obtained through the local perception of the position of three consecutive frames in the convolution layer (Ouyang et al., 2019).

It is important that 3D convolution kernel only extracts the visual features from the cube. Because the convolution kernel are the same throughout the process of convolutions, they are all the same kind of convolution kernel. Therefore, we use multiple convolution kernels to extract multiple features.

For CNNs, there is a general rule that the number of feature maps at the later layer (close to the output layer) should be increased, so that more types of features can be generated from low-level feature maps.

Based on the 3D convolution, various architectures were designed. Thus, the 3D CNN architecture has been developed for human action recognition:

3D CNN architecture in this thesis includes a hardwired layer, three convolutional layers, two lower sampling layers and a full connection layer. Each 3D convolution kernel convolves the cube with seven consecutive frames, the patch size of each frame is 60×40.

On the first layer, we applied a fixed core to tackle the original frame in the experiment, generate multiple channels of information, and deal with multiple channels separately. Finally, we combine the information of all channels to get the final description. This layer actually encodes our prior knowledge of the features, which is better than random initialization.

Each frame extracts the information of five channels, namely, the gradient in the gray, horizontal and vertical directions, and the optical flow in both directions as shown in Figure

3.7, where the first three are calculated per frame. Then the horizontal and vertical optical flows need two consecutive frames to be determined. There are 33 characteristic maps in total.

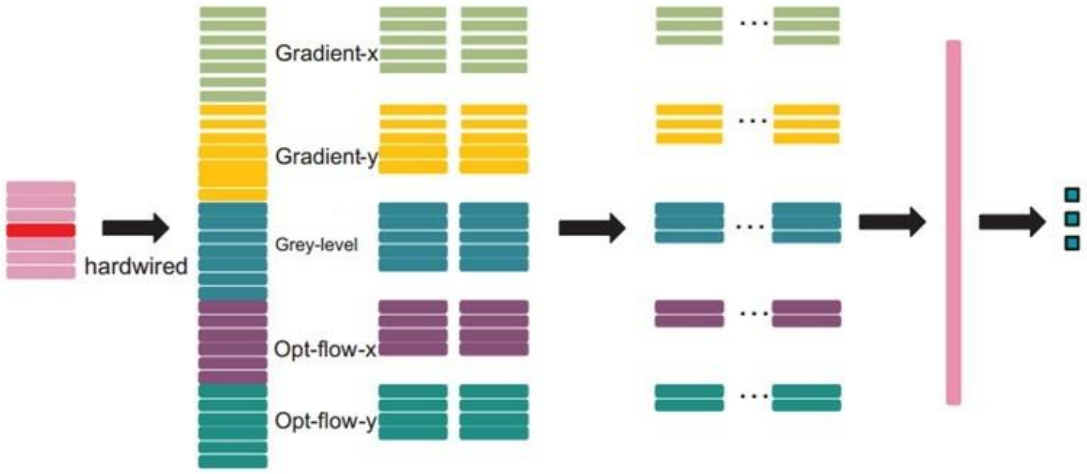


Figure 3.7 The information channel for each frame

A 3D convolution kernel (7×7 in space, 3 in temporal domain) is applied to convolve each of the five channels separately. In order to increase the number of feature maps. The experiment utilizes two different convolution kernels at each position. Thus, in the two feature maps at the C_2 layer, each group contains 23 feature maps. The patch size of each frame is 54×34.

The next lower sampling layer is the S_3 layer. In max pooling, the feature maps of C_2 layer are sampled under the 2×2 window. Therefore, the result gets the same number of feature maps with lower spatial resolution. C_4 is a 3D convolution kernel with $7 \times 6 \times 3$ in 5 channels. In order to increase the number of feature maps, three different convolution kernels are applied to each location. This gives us six different sets of feature maps, each has 13 feature maps. The patch size of each frame is 21×12, the S_5 layer uses a 3×3 down sample window, the result gets 7×4. Therefore, in this thesis, the changes after convolution operations in spatial dimension are intuitively shown in Figure 3.9.

Figure 3.8 The convolution of the spatial dimensions

At this stage, the number of frames in the time dimension is already very small. At this level, the experiment convolves only with spatial dimensions. The core is 7×4, and the outputs are reduced to 1×1. The C_6 layer contains 128 feature maps, each of which is fully connected with all 78 (13×6) feature maps in the S_5 layer. Then each feature map is going to be 1×1. After multilayer convolution and downsampling, the input image of each successive seven frames is converted into a 128-dimensional feature vector. This eigenvector captures the action of the input frame. The number of nodes in the output layer is as same as the number of action classes, each node is fully connected to the 128 nodes in C_6. As shown in Figure 3.9, a linear classifier is applied to classify the 128-dimensional feature vectors so as to implement human action recognition.

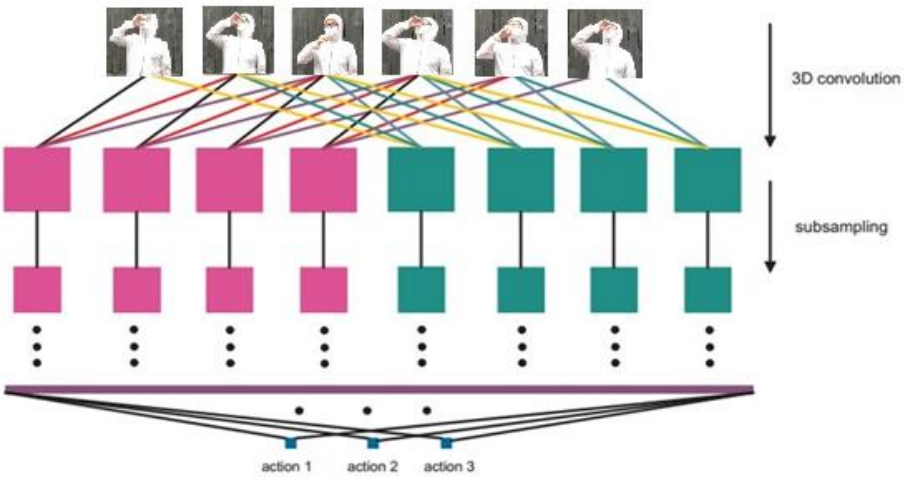


Figure 3.9 Human action recognition based on C3D algorithm

3.5 Evaluation Methods

In this thesis, three different recognition algorithms will be evaluated. The performance of

each algorithm is assessed by calculating accuracy rate, recall rate, and so on. The variables involved in the evaluation are explained in detail.

Table 3.1 Confusion matrix

		Predicted	
		Positive	Negative
Actual	True	TP	TN
	False	FP	FN

- True Positive (TP): The prediction is Positive, and it is actually positive.
- True Negative (TN): The prediction is Negative, and it is actually negative.
- False Positive (FP): The prediction is Positive, but in fact negative.
- False Negative (FN): The prediction was Negative, but in fact positive.

The accuracy rate is for all the original samples, which represents how many samples have been accurately predicted.

$$\text{ACC (accuracy)} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \tag{3.13}$$

where *accuracy* is relative to the predicted results. What it says is how many of the samples that are predicted to be positive are actually positive samples. Then there are two possibilities for a positive prediction. One is to predict a positive class as a positive class (*TP*). The other is to predict the negative class into the positive class (*FP*).

$$\text{P (precision)} = \frac{TP}{(TP+FP)} \tag{3.14}$$

The recall rate is for the original positive sample. This is how many of the positive samples were predicted correctly. There are two possibilities. One is to predict the original positive class into a positive class (*TP*). The other is to predict the original positive class as a negative class (*FN*). That is:

$$\text{R (Recall)} = \frac{TP}{(TP+FN)} \tag{3.15}$$

Chapter 4 Results

In this chapter, we present the experimental results of the three algorithms addressed in Chapter 3 and detail the experimental dataset as well as demonstrate the test samples. We also brief our experimental environment, how to collect the experimental dataset following our experimental design. Through tables and figures, we intuitively display our achievements. Finally, the outcomes of our experiments are evaluated.

4.1 Data Collection and Experimental Environment

In this experiment, HMDB-51 dataset was used as the training set. HMDB-51 has a total of 51 classes and 6,766 short videos. (URL:http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/#dataset)

The data comes from a vast range of sources, including movies, YouTube videos, etc. The dataset contains 51 classes, each class contains more than 101 videos. HMDB-51 is split into five groups:

- Facial movements (smile, laugh, chew, talk)
- Facial movements that are coordinated by something else (smoke, eat, drink)
- Common body movements (climb, dive, jump)
- Common body movements that are coordinated by something else (brush hair, catch, draw sword)
- Human interaction between the body movements (hug, kiss, shake hands)

The difference between the visual data for human action recognition and the data for general action recognition is that, in addition to the label of each clip, the data is manually labelled:

- Visible body parts
- Camera motion
- Camera view point
- Video quality
- The number of inviduals involved

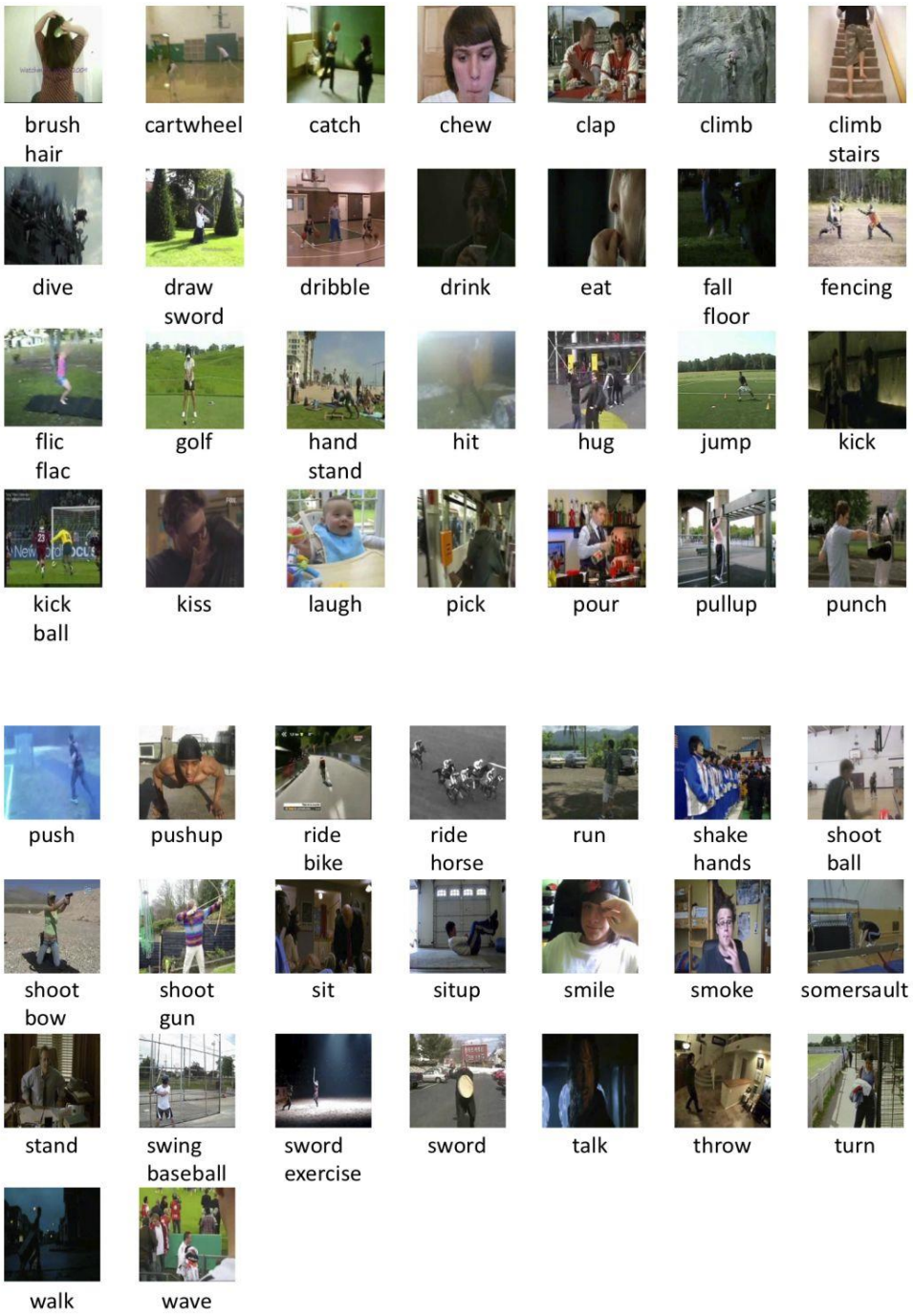


Figure 4.1 The 51 classes in HMDB-51

In this experiment, human actions that can be accomplished by using a single person are selected as the test video. In our experiments, seven actions were selected, clap (well done), wave (hello), hug, drink, run, hit and stand.

This experiment was conducted based on the software platform MATLAB R2020a. In recent years, MATLAB has been developed quickly in deep learning and computer vision,

with new features and showcases are released. For example, followed Computer Vision toolbox, MATLAB simulink also launched its support package for computer vision. In our experiments, MATLAB Image Processing Toolbox and Computer Vision Toolbox are employed. The image processing toolbox includes visual features such as image and video preprocessing and postprocessing, image analysis, spatial transformation, and color image processing. After the preprocessing, the video can be imported into the nets for human action recognition effectively.

4.2 Results of Human Action Recognition

In this section, we present the recognition results of three algorithms for human action recognition. Table 4.1 shows the accuracy of the three models.

Table 4.1 The accuracy of these three models for human action recognition

Models	Accuracy (%)
CNN+LSTM	**89.74%**
Two-Stream CNN	82.37%
3D CNN	86.54%

The experimental results of the three models are shown in Table 4.1. CNN+LSTM has the highest accuracy rate 89.74%. The 3D CNN method was the second and reaches up to 86.54%. Finally, the Two-Stream CNN model is 82.37%. Two of the three methods are more accurate, all are over 85 %. The best model is CNN+LSTM.

The experimental results show that CNN+LSTM has the best result for human action recognition among the three models. At the beginning of this thesis, the research question we asked also got a very clear answer. This model is the best way to deal with temporal problems and is worthy of further investigations.

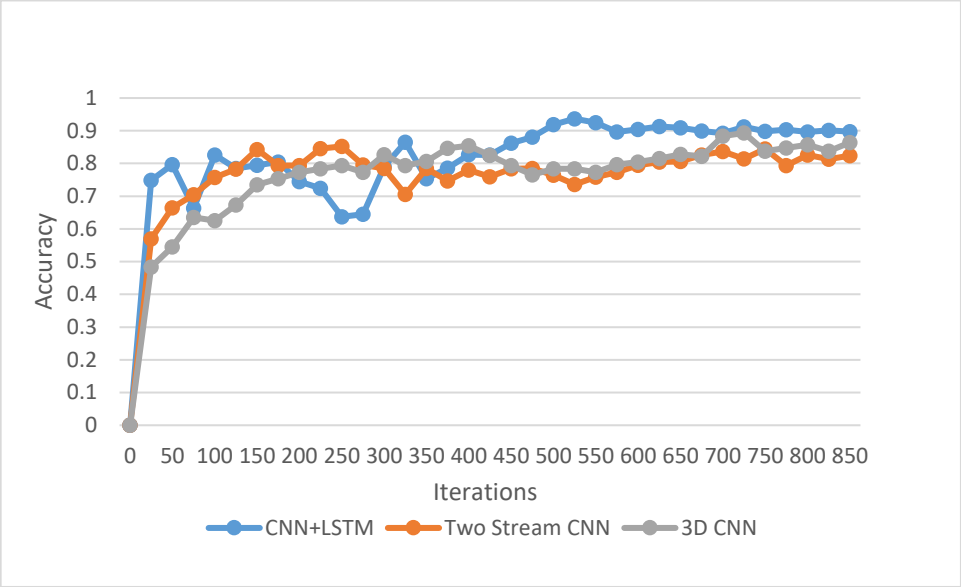


Figure 4.2 The accuracy of three algorithms for human action recognition

Figure 4.2 shows the accuracy of the three algorithms in our experiments. The blue color shows CNN+LSTM. The orange color represents the double flow method. The grey one stands for 3D CNN. Throughout 850 iterations, the final accuracy of each algorithm is obtained. The results in Figure 4.2 show the best result of CNN+LSTM. At the beginning of the iterations, the accuracy rate is fluctuating. However, as the number of iterations grows, the accuracy of each algorithm is basically stable.

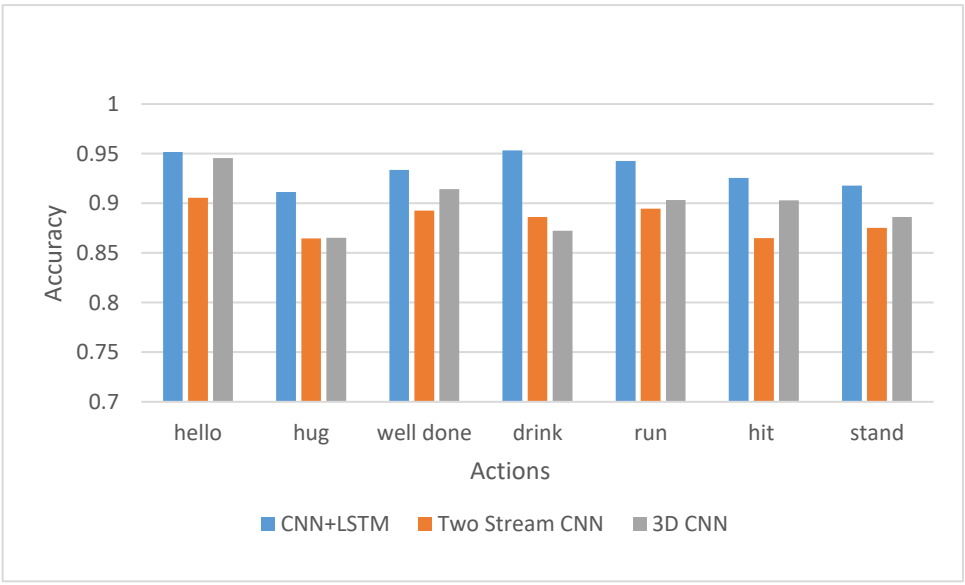


Figure 4.3 The accuracy rates of the three methods for human action recognition

Figure 4.3 shows the accuracy rates of the three algorithms in recognizing each human action. This bar stamp shows the recognition results of seven different actions in the three models. The blue color is for CNN+LSTM. Each action has a recognition rate of more than 90 %. Hello and Drink are up to 95%. Orange color stands for the Two-Stream CNN method, and the best action which has been identified as Hello. The other three were all below 90%, but above 85%. Gray color represents the 3D CNN algorithm, with the accuracy rate over 85%, Hello is also the highest one. But the recognition accuracy of Well-done was also very high, up to 92 %.

4.3 Demonstrations

In this section, we showcase the training process and practical examples. The actual recorded video is used as a test set to detect the training results and visualize the training process, we show the training process of the algorithms. Figure 4.4 shows the accuracy rate and loss rate of CNN+LSTM algorithm in the training process.

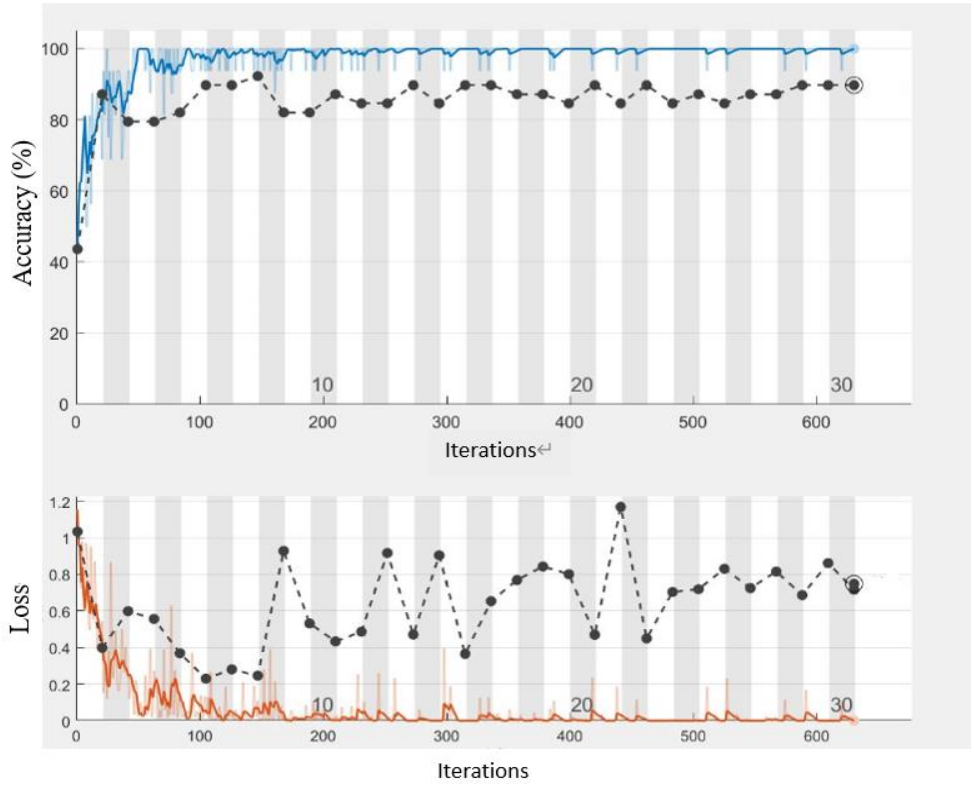


Figure 4.4 Training accuracy and loss rate of the CNN+LSTM

In Figure 4.4, the blue lines represent the training results, the yellow line shows the loss result, the black dots represent the verification results. Finally, the correct rate and loss rate are represented by using black dots. The results of human action recognition by using CNN+LSTM model are shown in Figure 4.5.

Figure 4.5 The results of human action recognition by using CNN+LSTM

As the results show, there is no recognition except for the last action. The rest of the movements have been accurately identified. The final action is Hugging. This should be fulfilled by two persons. But with COVID-19, we only have one person moving back and forth, which is a challenge. Because the dress color of this test video is white only. The impact on the outcome is also significant. Therefore, general recognition of the final action is very low. CNN+LSTM is already good at recognizing other movements. Figure 4.6 shows the accuracy rates and loss rates of the Two-Stream CNN algorithm in the training process.

The LSTM network performs well in processing sequential data. Global processing and memory cells are two important factors to solve the identification problem. Globalizing represents a complete input that contains all information, and localizing processing information can lose information and lead to modeling failure. The memory cell retains information about each step to dynamically adjust the next. In this way, LSTM can be applied to recognize the action information when dealing with time series problem.

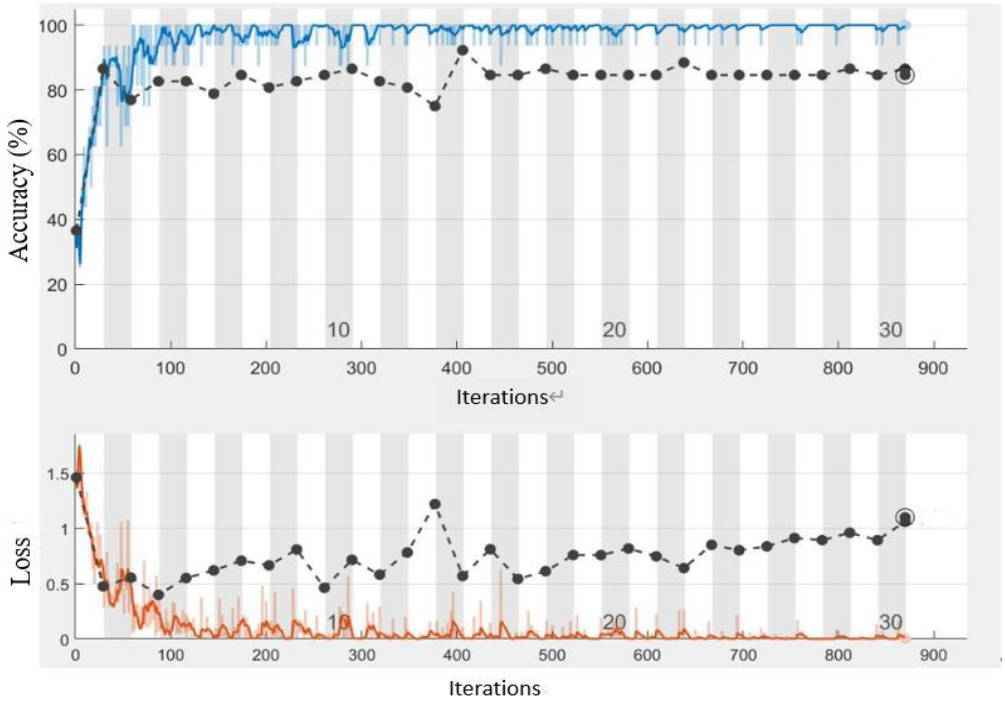


Figure 4.6 Training accuracy and loss rate of the Two-Stream CNN network

The blue line in Figure 4.6 shows the training results. The orange line shows the loss result. The black line represents the verification result. The final accuracy rate was calculated after 850 iterations. The recognition results of the Two-Stream CNN are shown as Figure 4.7.

Figure 4.7 The results of human action recognition by using the Two Stream CNN network

In the test video data, the white clothing made little difference between the two frames. Two-stream CNN model obtains the action characteristics in time through the information between adjacent frames. This makes the recognition results are not good. Under the same conditions, only the wave action was accurately identified. The action Clapping was recognized as drinking water. It was identified, but the result was wrong. However, the action of drinking water was not recognized. Finally, the action Hug was not recognized. The training process of 3D-CNN algorithm is shown in Figure 4.8.

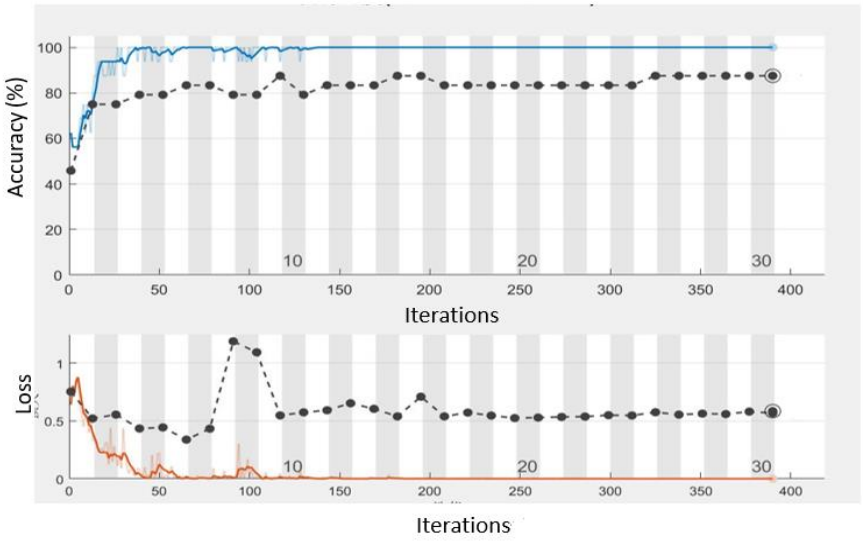


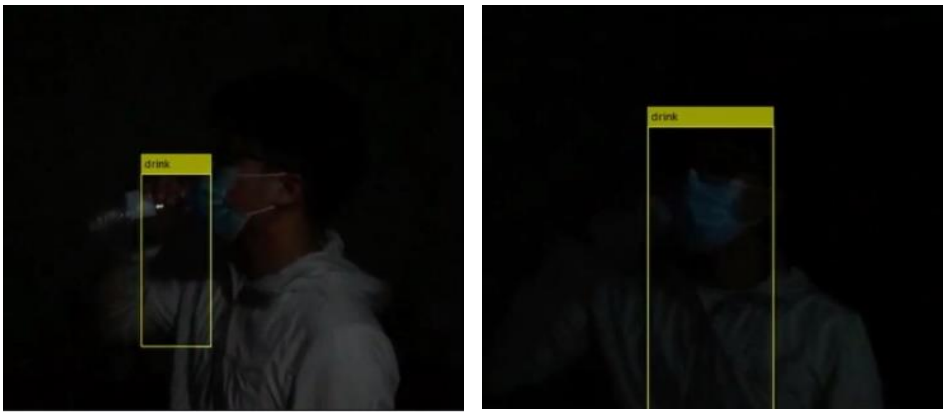
Figure 4.8 Training accuracy and loss rate of the 3D-CNN

Figure 4.9 The results of human action recognition by using 3D-CNN

The recognition results of 3D-CNN method are shown in Figure 4.9. 3D-CNN expands the controllable scope of convolution kernel to the time domain, which is more flexible than 2D convolution and can learn more action information. In addition, compared with RNN series method, it is much advantageous to learn the advanced representation of information, which is also a popular method in the field of motion recognition. However, the time feature extraction capability of 3D-CNN is not comparable with that of RNN. The 3D-CNN algorithm is able to accurately recognize the two actions Clapping and Waving. The action Drinking Water was recognized as a Waving. The same action Hug cannot be recognized.

Figure 4.10 The recognition results of human action: Drinking with a water glass.

Due to using a water glass, the result shows that it is easier to identify the action that has a cup in hand than the action that doesn't have a cup. Black sleeves worked well with white T-shirts when the color contrast is apparent. We successfully identified the action Hugging. The experiment was applied to identify the action under even worse conditions. Most experiments on intelligent recognition are conducted during the day. Lighting was moderate during the day for better capturing the motions. As a result, movement recognition is challenging in total dark endorsement. Human action recognition in the dark scene are very useful for unmanned driving technology, surveillance technology and so on. For example, we conduct the experiments at night when the lighting condition is very dim. The video color is almost black, even with our human eyes. There are a vast majority of noises in the test video, the recognition results are shown in Figure 4.11.



(a) (b)

Figure 4.11 Recognition results in dark environment

The results show that our actions even are able to be recognized in dark environment. The image on the left shows a man drinking from a bottle. On the right, however, it is hard to spot a person drinking water. The arms and water bottles are hard to be observed in the dark screen. But we still see the movement in our experiments. In addition, we have altered the view angles of a camera. In the front and side views, human actions are able to be accurately identified.

(a) (b)

Figure 4.12 The recognition results of human action: Wearing with reflective clothing

It’s dangerous danger during working on the road. Although reflective strips can be used as a warning for humans, they may not be feasible for computers, thus it's hard to recognize in a completely dark environment. However, as shown in Figure 4.12, reflective clothing is also recognizable in low lamination condition.

Overall, this chapter answers the three research questions raised at the beginning of this thesis. First of all, all three algorithms are able to recognize human actions in a video. Among the three models, the result of CNN+LSTM is the best. This indicates that the LSTM network has a better result for extracting the action features in terms of time series analysis. The corresponding model for human action recognition has better capability. In the deep learning algorithm, CNN+LSTM model can effectively extract the motion features in the given video and generate the recognition network for object detection.

At the end of the experiments, we made a test in a completely dark environment. If our human eyes cannot distinguish an action in a video, machine vision can't work either. Therefore, we wear reflective clothing to increase the recognizability of the main features. Then, the results show that the CNN+LSTM model is able to recognize the motions in a dark environment with a lot of noises.

4.4 Limitations of This Research Project

The three algorithms in this experiment are very straightforward and successful. However, due to the different background and environment, the experimental results have greatly been influenced. First of all, we see that the colour differences between clothes and hands are relatively minor. The action has side effect on object detection of target contour. It is difficult for a computer to decide the bounding box of the detected target, there is no way to obtain the visual features of the actions, which led to mistakes or wrong classification. Hence, the background colour has a side effect on computer vision. The number of samples in the training dataset are limited. For each action, if we have more training samples, the recognition accuracy will be better. By extracting various features, human actions are able to be recognized much accurately.

Therefore, the quantity and quality of training sets are important. High-quality samples are possible to reduce the noises of samples and eventually we can get better results. The video dataset used for model training is the key information influencing the experimental results. Better datasets can lead to better results. With the continuous accumulation of a large

amount of data on the Internet, we will find more suitable video data for the training of human action recognition.

Chapter 5 Analysis and Discussions

In this chapter, we chiefly state the analysis of the experimental results and discuss the relevant issues. After a large amount of data is obtained, the control variable method is employed to compare and analyze the experiments. We analyze the data and explain the possibility of the results. Throughout the analysis and discussion, the results of our experiments are clarified, the shortcomings of the experiments will be identified.

5.1 Analysis

In terms of evaluating the methods or algorithms, our experimental results are analyzed. The classes contained in the dataset, the characteristics of the dataset and related classification are described in this chapter. We clarify the characteristics of the selected dataset. Then, the whole method of this experiment is expounded. We evaluate the overall experimental design. The advantages and disadvantages of this experiment are pointed out and analyzed. Each algorithm identifies the characteristics of the action aspect. The experimental data in Chapter 4 are used for evaluation. Finally, the whole experiment assessment is accomplished.

5.1.1 Dataset

The HMDB-51 dataset contains 51 different classes. This dataset is ideal for evaluating human action recognition. However, the dataset was collected in 2011. There are plenty of video clips from digital movies. The description of human actions, especially the characteristics of actions, is not obvious. The video collects the movements of people from different angles or ages. For intelligent recognition, the dataset is very limited. In order to achieve the desired outcomes, a larger dataset is required. Of course, larger datasets mean more powerful hardware and efficient algorithms.

In the HMDB-51 dataset, there are 56.3% of the videos contain a complete human body. Only upper body videos are taken into account for 30.5%. Only the lower half of the video accounted for 0.8%. Finally, only the head accounts for 12.3%. 59.1 % of them were sports shots. That means the background is more complex. The background will change with the person. The remaining 40.1% was filmed in still.

The shooting angle is different in dataset. That is the direction of the person shown in the video. 40.8% are heads and 18.2% are tails. The left and right angles are 22.1% and 19.0%, respectively.

The clarity of the video includes good and low. “Good” means high quality video. In 17.1% of the high-quality videos, noses and eyes of human faces and the details of small

parts could be seen clearly. The majority of video quality is 62.1%. “Low” refers to the low-quality video. This kind of videos only include the movement of the human body, which is 20.8%.

5.1.2 Experimental Design

For the main experimental design of this experiment, the main variable is the difference of recognition algorithm. So, the rest of the structure of the experiment is as consistent as possible. In other words, the recognition video selected is the same. By identifying the same video data, the advantages and disadvantages of different algorithms are compared. In addition, the data as a training set is the same. Therefore, in this experiment, the dataset of HMDB-51 is selected as the dataset of all algorithms. The results are even more telling. The emphasis of this experiment is to analyze and compare various algorithms. It's important to keep the rest of the variable’s constant.

The experimental results show that CNN +LSTM has a strong recognition ability. The convolutional neural network algorithm can extract action features effectively. The LSTM performs well in timing processing. It's over 90% in single action recognition. This is shown in Table 5.1.

Table 5.1 The accuracy of recognition with different method

Method	Hello	Hug	Well done	Drink	Run	Hit	Stand
CNN+LSTM	**95.2%**	**91.1%**	**93.4%**	**95.3%**	**94.3%**	**92.3%**	**91.8%**
Two Stream CNN	90.6%	86.5%	89.3%	88.6%	89.5%	86.5%	87.5%
3D CNN	94.6%	86.5%	91.4%	87.2%	90.3%	87.3%	88.6%

Overall, all three methods accurately identified body language. More accurate identification is obtained by adjusting the parameters. In this experiment, though CNN+LSTM has the best effect, the Two-Stream CNN and 3D-CNN methods will also show better results by using continuously improvement.

5.1.3 Action Recognition Based on the Two-Stream Network

The advantage of two-stream network structure is that two convolutional neural networks are used to obtain different features of behavioral video. It can increase the diversity of action description characteristics and obtain more discriminative action characteristics. But the convolutional neural network is limited by its own properties. Only the Two-Stream network structure of the convolutional neural network is used. This structure ignores the temporal relation of the complete video frame sequence. It is difficult to cope with video samples with complicated visual temporal relationships and large visual appearance changes. In the following research on action recognition based on dual-stream network structure. In addition to optimizing the performance of the convolutional neural network. It should also consider modeling the timing of video frames. Such as adding time convolution module or cyclic neural network.

Space flow ConvNet operates on individual video frames. Because some behaviors are strongly associated with a particular scene or object, the algorithm can effectively recognize behavior from still images. Because CNN network is already a powerful image recognition algorithm, a video recognition network can be built based on large-scale image recognition algorithm. The pretraining network based on the existing image classification dataset is utilized.

Time-Flow ConvNet is different from the normal CNN network. The input requirement of Time-Flow ConvNet is formed by stacking optical flow displacement fields between successive frames. This input is characterized by action between video frames. This makes the identification process much easier.

5.1.4 Action Recognition Based on 3D-CNN Method

3D-CNN method increases the dimension of data input. If the training sample of dataset is not increased, in a disguised way, the training samples required by the neural network are reduced. Therefore, 3D CNN needs more human action samples to train the network. Under the same training conditions, the recognition effect is not outstanding. In addition, 3D-CNN

method only processes several adjacent frames simultaneously. Only short-term action information of the behavior is available. This method is still unable to model the complete video sequence. In many subsequent studies, 3D convolution kernel is used to replace 2D convolution. It is used as a basic network to extract visual appearance features and short-time action features in complex human action recognition.

In the 3D CNN network, 3D convolution has more time dimension than 2D convolution. The algorithm has four features: Generic, compact, efficient and simple. In addition, 3D convolutional network is effective and suitable for video spatiotemporal feature extraction. Through experiments, it is found that the convolution kernel is 3×3×3 and the feature extraction effect is the best. So this method has a wider application for video recognition.

5.1.5 Action Recognition Based on CNN+LSTM Model

The CNN-LSTM model makes up for the difficulty of the convolutional neural network in modeling behavioral temporal relations. The human action recognition learns the visual appearance information in video frames and the temporal relationship between video frames simultaneously. Although CNN+LSTM's working principle and structure are clear, what information the structure needs is still a more ambiguous question. So the ability to recognize a single action is ideal. Spatial and temporal information for multimodal data input. It is known that modeling leads to differentiated behavioral characteristics. By effectively integrating multi-modal behavior characteristics, we can learn from each other and obtain more discriminative behavior descriptors.

CNN is used in the algorithm to get the global description of the video level, increasing the number of frames can significantly improve the classification performance. Five pooling structures are proposed through comparative analysis.

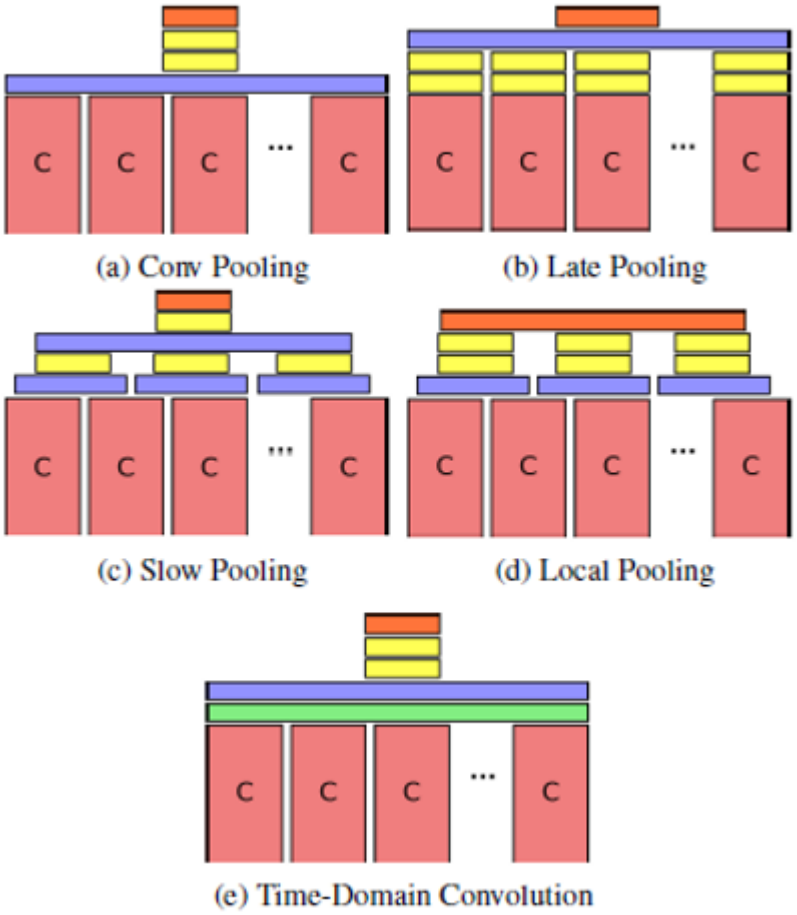


Figure 5.1 Different feature pooling architectures

In Figure 5.1, the stacked convolutional layers are denoted by 'C'. Blue, green, yellow and orange rectangles represent max-pooling, time-domain convolutional, fully connected and softmax layers respectively.

(a) Conv pooling: Use max pooling after the final convolutional layer of the video frames. The main advantage of this model is that the output information of the convolutional layer can be retained by the maximum pooling operation in the time domain.

(b) Late pooling: Firstly, input the features after convolution into the two full connection layers, and then pool the output. The parameters of all convolutional layers and full connection layers are shared. Different from others, this method directly pools the inter-frame features of higher level.

(c) Slow pooling: Use small time windows to pool the features of the frame level. The two-step pooling strategy is adopted, max pooling is used to pool the 10 frame features. The step is 5 (pooling here can also be regarded as the convolution kernel with size 10 convolving

the one-dimensional characteristics of the input with step 5. After each maximum pooling layer is the full connection layer with shared parameters. The second step is to maximize pooling on all the output of the full connection layer. The characteristic of this method is to combine the local time features before pooling the inter-frame advanced features.

(d) Local pooling: Similar to slow pooling, this method is piecewise pooling of frame-level features after the final convolutional layer, except that there is only one pooling stage. After pooling, we connect two full connection layers and share parameters. Finally, we input all output into a large softmax layer. Reducing a pooling layer can avoid the potential loss of timing characteristics.

(e) Time-domain convolution layer is added before the pooling layer for this method. The convolutional layer is composed of 256 3×3 convolution kernels with a stride length of 5. This method can capture the local relationship between frames through a small-time window.

The Conv-pooling method works best and outperforms all other feature pooling architectures. The late-pooling method is the worst. Relevant experiments show that it is important to retain the original spatiotemporal information before pooling. In this method, the frame features are directly input to the full connection layer, and the spatial and temporal information is lost to obtain the high-level features. The effect of the Time-Domain Convolutional method is also poor, which indicates that the effect of the convolutional layer in the time domain is not as good as max pooling.

5.2 Discussions

In the traditional recognition method of human body movement, the process of feature extraction is mainly implemented manually. Its importance is dependent on a great deal of experience and background knowledge. The parameter tuning process is too complex and subjective. For most algorithm models, they only perform well on the dataset used in the experiment, and their portability is poor. There are a lot of images and video data on the Internet. Only relying on manual extraction of features is far from meeting the needs of development. Subsequently, a large number of feature extraction methods were proposed.

For example, there are four methods:

- Histogram of Oriented Gradient (HOG): Firstly, the horizontal and vertical errors of the image plane pixels are calculated, and then summed into the gradient amplitude and gradient direction. Finally, the direction of gradient is divided, and the weighted statistics are performed according to the gradient size.

- Histogram of Optical Flow (HOF): The horizontal and vertical velocity of the pixel are calculated using gradient and time gradient graphs, the amplitude and direction of the resultant velocity are calculated according to the above method.

- Motion Boundary Histograms (MBH): The horizontal and vertical optical flow gradients are calculated on the optical flow diagram and the amplitude and direction of the resultant optical flow gradient are calculated.

- Trajectories: The coordinate of matching point is obtained according to optical flow velocity, and coordinate difference of adjacent frame matching point is obtained. It's a series of tracks. The regularization is then a trajectory feature.

With the continuous development of the field of intelligent identification, the deep learning methods were proposed. Human actions in a video are identified directly by using the end-to-end way. According to feature extraction methods, the approaches are grouped into two categories, e.g., human action recognition based on skeletons, human action recognition based on feature maps. It is implemented for automatic extraction from low-level features to high-level features layer by layer, and its portability is stronger than traditional methods. It has made remarkable achievements in the field of computer vision.

But deep learning methods are in the infancy stage. Human action is rather complex, action analysis is influenced by background clutter, different lighting conditions, different image acquisition equipment, and insufficient human motion library categories. Deep learning methods still have large room to be promoted in the applications of action recognition. Therefore, deep learning methods still have extremely important value to implement self-learning so as to identify human actions. The three methods in this

experiment are all based on CNN networks. The advantages of convolutional neural networks are:

- Local connection: Each neuron is no longer connected to all the neurons in the upper layer, but only to a small number of neurons. This reduces a lot of parameters.
- Weight sharing: A group of connections can share the same weight, instead of each connection having a different weight, which again reduces many parameters.
- Lower sampling: Pooling layer utilizes the principle of local correlation of images to conduct sub-sampling of images, which can reduce data processing capacity and retain useful information. By removing unimportant samples from feature map, the number of parameters can be further reduced.

A CNN network is essentially an input-output mapping. It can be trained by using a large number of mappings between inputs and outputs without requiring any precise mathematical expressions between inputs and outputs. As long as the convolutional network is trained with known patterns, the network has the ability to map between input and output pairs. The convolutional network performs supervised training, so its sample set is composed of samples and labels.

In addition, pooling is able to reduce the spatial resolution of the network. This eliminates small shifts and distortions in the signal. Therefore, the translation invariance of input data is not required high. For human action recognition, the convolutional neural network achieves better learning effect by keeping as many important parameters as possible and removing a large number of unimportant parameters.

On the other hand, the in-depth research work for human action classification is based on pattern classification of single person action. Understanding human action in real life is still a challenge:

- Complex backgrounds in real video scenes: The performance of the depth behavior classification model is affected by such uncertain factors as light change, human

appearance change, camera angle and motion speed.

- The real video stream contains a large number of long-time non-action redundant video segments. Based on the classification of human action. Human action detection is to further extract and clarify the time boundary of human action. This task is expected to be solved based on the deep learning-based classification model, but the accuracy and speed are lower than the current requirements.

- The action classification model based on RGB video is able to obtain detailed appearance and texture features from video image frames. But it is difficult to model different human movements in space and time. Therefore, it is difficult to apply to multi-person scenarios.

Each frame of RGB video contains a wealth of shapes, colors, and texture information. The CNN network is used to extract visual information from each frame of such 2D image video. Although the depth data contains rich scene information, human action recognition is not ideal for three reasons:

- The depth map lacks color, texture and body posture information, which weakens the discriminant expression ability of convolutional neural network (CNN) model.

- The depth data contains a lot of noise.

- The existing depth data are still small-scale, which is easy to overfit the CNN model driven by data to learn and express.

In the skeleton-based human action recognition method, 3D skeleton real-time tool was developed by Microsoft, the reliable and effective real-time posture estimation and detection algorithm were proposed in recent two years, which becomes easier to obtain the 3D skeleton information of each frame from RGB video and depth video data in real time. Bone data is from the positions of key points of the human body, which is expressed in 2D and 3D coordinates that are more abstract expressions of the human body.

For the dataset, HMDB-51 is used in our experiment. The videos in this dataset are

already segmented. Each segment contains only one action. Segmented videos are more conducive to the learning of motion features. However, this also limits the evaluation of the three models in this experiment. If more datasets are taken into account, we are able to compare the recognition results of each method based on different datasets. Analyzing datasets is very beneficial in identifying human actions. In order to evaluate the recognition ability of the three recognition models, we choose the same dataset as the training set to reduce the interference of external factors. To make the experiment achieve better results.

There are a lot of noises in the test samples we chose. For example, ambient light, shooting angle, and other actions other than target action will interfere with our experiments. However, the extraction of features by the convolutional neural network has reached an ideal level. Not only can three algorithms correctly identify the actions in the test set video, but also it is able to recognize human motions.

In this chapter, our experimental results are analyzed and discussed in detail. We analyze the three models in detail and list their advantages and disadvantages. In addition, we also commented on the experimental design and the dataset. Throughout the analysis of CNN detection series in deep learning, the importance of deep learning to human motion recognition is expounded in detail.

Chapter 6 Conclusion and Future Work

In this thesis, we have deeply investigated the importance of deep learning methods. Three deep learning methods for human action recognition are proposed and tested. As the final chapter of this thesis, we summarize and overview the previous content from a high-level perspective. Simultaneously, we point out our future work at the end of this thesis.

6.1 Conclusion

Feature extraction is the most critical step in human action recognition, which relies on domain knowledge and human experience and cannot meet the demands of data growth. Therefore, we take deep learning methods as our start point in this thesis. The mainstream method in deep learning is based on CNNs. Therefore, three recognition algorithms, i.e., the Two- Streams CNN, CNN+LSTM, and 3D-CNN nets are mainly taken into account in this thesis. Throughout feature selection, the algorithms successfully recognized human actions from a given video footage. However, they are distinct in dealing with time series problems. Our experiments show that LSTM can better deal with this problem. Therefore, LSTM+CNN recognizes human actions from the videos effectively.

Human action recognition is a major research direction in the field of intelligent recognition, which is closely related to our ordinary life. The deep learning methods for human action recognition are employed to the interactions between human and computers. We applied these methods to video surveillance, which not only curbs incidents but also curtails criminals. In medical treatment and assistance, it has the auxiliary for medical therapy, which greatly saves human labors and time costs, thus promotes our living quality.

The rise of deep learning in recent years makes the era of artificial intelligence real. Because deep neural networks automatically adjust the parameters according to training dataset, visual features of the training and test samples need to be extracted in the end-to-end way. The demands of deep learning are being expanded and bloomed everywhere. However, the methods of human action recognition have been exploited slowly due to the model training and feature extraction from image sequences. In this thesis, deep learning is brought in to train the models of human actions, the continuous movement of human body is successfully recognized. The main research outcomes are reiterated as follows:

- Three deep learning methods based on CNNs are proposed. The experiments were conducted for testing these methods, namely, the Two-Stream CNN, 3D CNN, and CNN+LSTM, for human action recognition, respectively.

- The three methods were implemented to identify individual actions. Compared with the experimental results, the accuracy of CNN+LSTM is the best one.

- In the Two-Stream CNN model, feature extraction was carried out from spatiotemporal domain. The two sets of feature maps were fused together to gain the final result.

- In the CNN+LSTM model, the feature maps extracted from the CNNs were organized in temporal sequence. The method greatly upgraded the accuracy and efficiency of human action recognition.

- In 3D CNN model, five channels were selected, which are horizontal gradient, vertical gradient, grayscale gradient, horizontal and vertical flows, respectively. The dataset was applied for model training. Finally, the human action recognition is achieved.

- Human action recognition in night environment has been raised amid the experimenting, we expect to solve the problem of human action recognition if a reflective suit or gloves are worn.

- We anticipate conducting the experiments with multiple angles of view if human action is perceived without occlusions. Our experimental results reveal that this group of actions can be identified.

Feature extraction is the most critical step in human action recognition. Therefore, in this thesis, we take deep learning methods as the base stone. The mainstream methods of deep learning are based on CNNs. Hence, the three recognition methods, namely, the Two-Stream CNN, CNN+LSTM, and 3D-CNN nets are chiefly implemented in this thesis. Throughout analyzing feature maps of deep learning methods, the algorithms successfully recognized human actions in the given videos. However, they suffer handling time series problems. Our experiments show that LSTM is better to deal with these issues. Therefore, LSTM+CNN model is able to recognize human actions in the videos much effectively.

Finally, our experiments were conducted based on a large-scale number of surveys. The goal of this thesis is to implement human action recognition and find out the best method to uplift the accuracy of human action recognition. Pertaining to the experimental dataset, we chose HMDB-51. Although the number of samples of this dataset is not very large, it is appropriate for our experiments. Moreover, the videos to verify the results are all taken by ourselves. In order to attain the best results, we recorded the videos with human actions in a diversity of experimental environments, which increases the difficulty of our action recognition and makes it easily to find the best recognition method.

Throughout the implementation and analysis of these three methods, namely, the Two-Stream CNN, 3D-CNN, and CNN+LSTM, we finally draw the conclusion based on the outcomes of our experiments. All the three methods effectively identify human actions from the given videos. The CNN+LSTM model is the best one based on experimental performance. The LSTM network is suitable for dealing with the issues related to time series analysis.

6.2 Future Work

Digital videos have temporal characteristic compared to 2D images, which is complex and diverse. The annotation of video data is time-consuming, laborious and expensive. Therefore, deep learning models for video-based classification are relatively slow compared with the static images. The performance of deep learning models in human action classification, interaction recognition, and motion detection has been verified. However, in real complex scenarios such as intelligent video surveillance, there are too many problems in feature learning, interactive recognition, and spatiotemporal action locating from multimodal data. We will devote to resolve these problems in future.

6.2.1 More Datasets

Human action recognition is chiefly based on training video samples, which includes human gesture recognition, gait recognition, sign language recognition, etc. Different from human

action recognition from digital images, there are an abundant of datasets such as MNIST and ImageNet. In the field of human action recognition, the datasets are relatively limited and usually occupy a large amount of storage space. Therefore, it is necessary to carefully select an appropriate dataset before human action recognition.

ActivityNet is a dataset offered by Google in 2016. The source of this dataset is mainly from YouTube, with a strong deep learning background and a large number of samples. ActivityNet is the largest dataset in human action analysis at present, which includes action classification and action detection. The current version of ActivityNet dataset includes 20,000 YouTube videos, the training set contains about 10,000 videos, the validation set and test set include more than 5,000 videos, respectively, 700 hours video footages in total. On average, there are 1.5 action instances for each video. ActivityNet covers over 200 daily activities, such as Walking, Long Jump, and Vacuuming Floor. The distributions of data volumes are 2:1:1 for training dataset (~50%), validation dataset (~25%), test dataset (~25%). Therefore, this dataset is relatively complex and suitable for object detection. However, the dataset is much suitable for human action recognition.

The 20BN-Jester dataset is a large one of video clips with dense markup, which shows human gestures in front of a laptop camera or webcam. The datasets are created by using a large number of population workers, which allows robust machine learning models to be trained so as to recognize human gestures. In the gesture recognition, the background of the videos is relatively static, the actions are much simple. Therefore, it is suitable for human action recognition. In addition, the site also provides object-based action recognition, two sets of visual data will be much meaningful to the actual scenarios.

The NTU RGB + D dataset for human action recognition consists of 56,880 videos, including RGB videos, feature map sequence, 3D skeleton data and infrared video for each sample. This dataset was captured simultaneously by using three Microsoft Kinect cameras. The resolution of RGB videos is 1920×1080, the depth maps and infrared videos are with the size 512×424, the 3D skeleton data contains 3D positions of 25 human body joints per frame.

3D bone node information is obtained by using the Kinect cameras. It is composed of the 3D position coordinates of 25 human body joints. Skeletal tracking with depth data is applied to establish the coordinates of various joints of human body, which identifies all parts of the body like hands, head, body, and the position where they are. The dataset provides a pretty rich amount of visual data, the video background is relatively fixed, which makes it suitable for human action recognition. Meanwhile, the data is characterized by using RGB and depth at the same time. The total amount of the dataset is up to 1.3TB, furthermore, bigger datasets are provided later, such as NTU RGB+D 120.

We also have collected our own training sets. By recording a video or capturing an appropriate video, we establish our dataset, especially with the videos taken at night. Because most of the current datasets were designed in daytime, our contributions are the unique one and could be applied to various scenarios. The identification for human actions is paramount at night. More experimental results will be further conducted by using this night dataset for human action recognition.

6.2.2 Future Experiments

Multimode biometrics are an assortment of important applications of human action recognition in the near future (Okereafor, Osuagwu, & Onime, 2016). The wide applications of biometrics aim at improving the access security and personal authentication in various scenarios. However, the biometric identification methods still have defects, more and more solutions based on multimode biometrics are under investigation.

By combining two or more methods, the multimodal methods are able to make up the defects of single biometrics. Multimodal visual perception and motion representation are based on multimodal fusion. At present, the multimode methods are related to deep learning paradigm which are effectively trained for human action recognition. Multimodal data integrated from RGB data, depth data, and skeleton data is a critical problem in human action recognition projects.

At present, human action recognition under dark conditions is a hot research topic. Most

of the identification problems took place in daytime. But at night, the ability to collect images and identify human actions is weak. A great deal of noises and interferences may affect the identification. Therefore, it is particularly important to improve the quality of input videos. We work on collecting video data using infrared night vision or infrared thermography. The videos are very different from the light-filled video data, but the main motion features are captured. This is of great values to accurately identify actions and improve the accuracy of identification.

6.2.3 Evaluation Methods and Applications

Based on the available methods of human action recognition at present, most of the methods are employed to evaluate the accuracy rates. By counting the numbers of TP (true positive), TN (true false), FP (false positive) and FN (false negative) in the tests, accuracy, precision, and recall are calculated. We design an effective evaluation method for the efficiency of human action recognition. In this way, we accurately compare the distinctions between these methods, and apply our findings to furthermore experiments.

Last but not least, in the near future, we will apply human action recognition to a much wide spectrum of fields. We will improve the quality of our daily life using deep learning and computer vision. At present, machine vision, deep learning, and artificial intelligence are still at the early stage. There are a plethora of research problems awaiting to be solved. We will put our efforts to these ubiquitous and empirical computing problems so as to uplift our living quality.

References

Aggarwal, J. K., & Xia, L. (2014). Human activity recognition from 3D data: A review. *Pattern Recognition Letters, 48*, 70-80

Agrawal, P., Girshick, R., & Malik, J. (2014). Analyzing the performance of multilayer neural networks for object recognition. In *European Conference on Computer Vision*

An, N., Yan, W. (2021) Multitarget tracking using Siamese neural networks. *ACM Transactions on Multimedia Computing, Communications and Applications*

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*

Baisware, A., Sayankar, B., & Hood, S. (2019). Review on recent advances in human action recognition in video data. In *International Conference on Emerging Trends in Engineering and Technology-Signal and Information Processing (ICETET-SIP-19)* (pp. 1-5)

Barros, P., Parisi, G. I., Jirak, D., & Wermter, S. (2014). Real-time gesture recognition using a humanoid robot with a deep neural architecture. In *IEEE-RAS International Conference on Humanoid Robots*

Beijbom, O. (2012). Domain adaptations for computer vision applications. *arXiv preprint arXiv:1211.4860*

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., & Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision*

Bruno, B., Mastrogiovanni, F., Sgorbissa, A., Vernazza, T., & Zaccaria, R. (2013). Analysis of human behavior recognition algorithms based on acceleration data. In *IEEE International Conference on Robotics and Automation*

Burgos-Artizzu, X. P., Dollár, P., Lin, D., Anderson, D. J., & Perona, P. (2012). Social behavior recognition in continuous video. In *IEEE Conference on Computer Vision and Pattern Recognition*

Chang, H. S., Fu, M. C., Hu, J., & Marcus, S. I. (2016). Google deep mind's AlphaGo. *OR/MS Today, 43*(5), 24-29

Chen, C., Liu, M.-Y., Tuzel, O., & Xiao, J. (2016). R-CNN for small object detection. In *Asian Conference on Computer Vision*

Chen, Y. (2010). Study of moving object detection in intelligent video surveillance system. In *International Conference on Computer Engineering and Technology*

Chen, Y., Li, W., Sakaridis, C., Dai, D., & Van Gool, L. (2018). Domain adaptive Faster R-CNN for object detection in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*

Cui, W., Yan, W. (2016) A scheme for face recognition in complex environments. *Int. J. Digit. Crime Forensics* 8(1): 26-36

Dašić, P., Dašić, J., & Crvenković, B. (2017). Improving patient safety in hospitals through usage of cloud supported video surveillance. *Macedonian Journal of Medical Sciences, 5*(2), 101

Dong, Y., & Li, J. (2018). Video retrieval based on deep convolutional neural network. In *International Conference on Multimedia Systems and Signal Processing*

Du, Y., Fu, Y., & Wang, L. (2015). Skeleton based action recognition with convolutional neural network. In *Asian Conference on Pattern Recognition (ACPR)*

E-Martín, Y., R-Moreno, M. D., & Smith, D. E. (2015). A fast goal recognition technique based on interaction estimates. In *International Conference on Artificial Intelligence*

Fan, Y., Lu, X., Li, D., & Liu, Y. (2016). Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *ACM International Conference on Multimodal Interaction*

Fernández-Llatas, C., Benedi, J.-M., García-Gómez, J. M., & Traver, V. (2013). Process mining for individualized behavior modeling using wireless tracking in nursing homes. *Sensors, 13*(11), 15434-15451

Freeman, H. (2012). *Machine Vision: Algorithms, Architectures, and Systems*. Elsevier.

Girshick, R. (2015). Fast R-CNN. In *IEEE International Conference on Computer vision*

Gopi, E., Lakshmanan, N., Gokul, T., & KumaraGanesh, S. (2006). Digital image forgery detection using artificial neural network and auto regressive coefficients. In *Canadian Conference on Electrical and Computer Engineering*

Grauman, K., & Leibe, B. (2011). Visual object recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning, 5*(2), 1-181

Gu, D., Nguyen, M., Yan, W. (2016) Cross models for twin recognition. *Int. J. Digit. Crime Forensics* 8(4): 26-36

Gu, Q., Yang, J., Yan, W., Klette, R. (2017) Integrated multi-scale event verification in an augmented foreground motion space. In *PSIVT* (pp.488-500)

Guo, S., Luo, H., & Yong, L. (2015). A big data-based workers behavior observation in China metro construction. *Procedia Engineering, 123*, 190-197

Holcomb, S. D., Porter, W. K., Ault, S. V., Mao, G., & Wang, J. (2018). Overview on deepmind and its AlphaGo. In *International Conference on Big Data and Education*

Hsu, S. C., Huang, C. L., & Chuang, C. H. (2018). Vehicle detection using simplified fast R-CNN. In *International Workshop on Advanced Image Technology (IWAIT)* (pp. 1-3)

Isaacson, M., & Shoval, N. (2006). Application of tracking technologies to the study of pedestrian spatial behavior. *The Professional Geographer, 58*(2), 172-183

Ji, H., Liu, Z., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. In *ACPR* (2, pp.503-515)

Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(1), 221-231

Jiang, H., & Learned-Miller, E. (2017). Face detection with the Faster R-CNN. In *IEEE International Conference on Automatic Face & Gesture Recognition* (FG 2017)

Jing, L., Ye, Y., Yang, X., & Tian, Y. (2017). 3D convolutional neural network with multi-model framework for action recognition. In *IEEE International Conference on Image Processing (ICIP)* (pp. 1837-1841)

Jung, I., Son, J., Baek, M., & Han, B. (2018). Real-time MDNet. In *European Conference on Computer Vision* (ECCV)

Junior, J. C. S. J., Musse, S. R., & Jung, C. R. (2010). Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine, 27*(5), 66-77

Khan, R. Z., & Ibraheem, N. A. (2012). Hand gesture recognition: A literature review. *International Journal of Artificial Intelligence & Applications, 3*(4), 161

Khan, S., Rahmani, H., Shah, S. A. A., & Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, *8*(1), 1-207

Koo, S.-y., Lim, J. G., & Kwon, D.-s. (2008). Online touch behavior recognition of hard-cover robot using temporal decision tree classifier. IEEE International Symposium on Robot and Human Interactive Communication

Koresh, M., & Deva, J. (2019). Computer vision based traffic sign sensing for smart transport. *Journal of Innovative Image Processing (JIIP), 1*(01), 11-19

Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian detection based on YOLO network model. In *IEEE International Conference on Mechatronics and Automation* (ICMA)

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning.*Nature, 521*(7553), 436-444

Liu, C., Yan, W. (2020) Gait recognition using deep learning. *Handbook of Research on Multimedia Cyber Security* (pp.214-226)

Li, C., Wu, X., Zhao, N., Cao, X., & Tang, J. (2018). Fusing two-stream convolutional neural networks for RGB-T object tracking. *Neurocomputing, 281*, 78-85

Li, C., Song, D., Tong, R., & Tang, M. (2019). Illumination-aware faster R-CNN for robust multispectral pedestrian detection. *Pattern Recognition*, *85*, 161-171

Li, F., Zhang, Y., Yan, W., Klette, R. (2016) Adaptive and compressive target tracking based on feature point matching. In *ICPR* (pp.2734-2739)

Li, H., Li, Y., & Porikli, F. (2015). DeepTrack: Learning discriminative feature representations online for robust visual tracking. *IEEE Transactions on Image Processing, 25*(4), 1834-1848

Li, H., Lin, Z., Shen, X., Brandt, J., & Hua, G. (2015). A convolutional neural network cascade for face detection. In *IEEE Conference on Computer Vision and Pattern Recognition*

Li, J. (2017). Parallel two-class 3D-CNN classifiers for video classification. In *International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)* (pp. 7-11)

Li, J., Liang, X., Shen, S., Xu, T., Feng, J., & Yan, S. (2017). Scale-aware Fast R-CNN for pedestrian detection. *IEEE Transactions on Multimedia, 20*(4), 985-996

Li, R., Nguyen, M., Yan, W. (2017) Morse codes enter using finger gesture recognition. In *DICTA* (pp.1-8)

Li, W., Hsieh, C., Lin, L., & Chu,W. (2012). Hand gesture recognition for post-stroke rehabilitation using leap motion. In *International Conference on Applied System Innovation (ICASI)* (pp. 386-388)

Li, Y., Zhang, H., & Shen, Q. (2017). Spectral–spatial classification of hyperspectral imagery with 3D convolutional neural network. *Remote Sensing, 9*(1), 67

Li, Z., & Zhou, F. (2017). FSSD: Feature fusion single shot multibox detector. *arXiv:1712.00960*

Liu, F., Zhou, Z., Jang, H., Samsonov, A., Zhao, G., & Kijowski, R. (2018). Deep convolutional neural network and 3D deformable approach for tissue segmentation in musculoskeletal magnetic resonance imaging. *Magnetic Resonance in Medicine, 79*(4), 2379-2391

Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016). Spatio-temporal lstm with trust gates for 3D human action recognition. In *European Conference on Computer Vision*

Liu, J., Udupa, J. K., Saha, P. K., Odhner, D., Hirsch, B. E., Siegler, S., Winkelstein, B. A. (2008). Rigid model-based 3D segmentation of the bones of joints in MR and CT images for motion analysis. *Medical Physics, 35*(8), 3637-3649

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European Conference on Computer Vision*

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. *Neurocomputing, 234*, 11-26

Liu, X., Deng, C., Chanussot, J., Hong, D., & Zhao, B. (2019). STFNet: A two-stream convolutional neural network for spatiotemporal image fusion. *IEEE Transactions on Geoscience and Remote Sensing, 57*(9), 6552-6564

Liu, X., Yan, W., Kasabov, N. (2020) Vehicle-related scene segmentation using CapsNets. In *IVCNZ* (pp.1-6)

Liu, X., Neuyen, M., Yan, W. (2019) Vehicle-related scene understanding using deep learning. In *ACPR Workshops* (pp.61-73)

Liu, Z., Yan, W., Yang, B. (2018) Image denoising based on a CNN model. In *ICCAR* (pp.389-393)

Lu, J., Shen, J., Yan, W., Bacic, B. (2017) An empirical study for human behavior analysis. *Int. J. Digit. Crime Forensics* 9(3): 11-27

Lu, J., Yan, W., Nguyen, M. (2018) Human behaviour recognition using deep learning. In *AVSS* (pp.1-6)

Lu, J., Nguyen, M., Yan, W. (2020) Deep learning methods for human behavior recognition. In *IVCNZ* (pp.1-6)

Lu, J., Nguyen, M., Yan, W. (2020) Comparative evaluations of human behaviour recognition using deep learning. *Handbook of Research on Multimedia Cyber Security* (pp.176-189)

Lu, J., Nguyen, M, Yan, W. (2021) Sign language recognition from digital videos using deep learning methods. In *ISGV* (pp.108-118)

Lu, X., Chen, Y., & Li, X. (2017). Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features. *IEEE Transactions on Image Processing, 27*(1), 106-120

Luo, X., Li, H., Cao, D., Yu, Y., Yang, X., & Huang, T. (2018). Towards efficient and objective work sampling: Recognizing workers' activities in site surveillance videos with two-stream convolutional networks. *Automation in Construction, 94*, 360-370

Ma, C., Huang, J.-B., Yang, X., & Yang, M.-H. (2015). Hierarchical convolutional features for visual tracking. IEEE International Conference on Computer Vision

Ma, L., Lu, Z., & Li, H. (2015). Learning to answer questions from image using convolutional neural network. *arXiv:1506.00333*

Ma, X., & Hovy, E. (2016). End-to-end sequence labeling via bi-directional LSTM-CNNs-

CRF. *arXiv:1603.01354*

Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition*

Morota, G., Ventura, R. V., Silva, F. F., Koyama, M., & Fernando, S. C. (2018). Big data analytics and precision animal agriculture. *Journal of Animal Science, 96*(4), 1540-1550

Mühling, M., Korfhage, N., Müller, E., Otto, C., Springstein, M., Langelage, T., Freisleben, B. (2017). Deep learning for content-based video retrieval in film and television production. *Multimedia Tools and Applications, 76*(21), 22169-22194

Nguyen, K., Fookes, C., Ross, A., & Sridharan, S. (2017). Iris recognition with off-the-shelf CNN features: A deep learning perspective. *IEEE Access, 6*, 18848-18855

Ning, C., Zhou, H., Song, Y., & Tang, J. (2017). Inception single shot multibox detector for object detection. In *IEEE International Conference on Multimedia & Expo Workshops*

Okereafor, K., Osuagwu, O., & Onime, C. (2016). Multi-biometric liveness detection–A new perspective. *West African Journal of Industrial and Academic Research, 16*(1), 26-37

Ouyang, X., Xu, S., Zhang, C., Zhou, P., Yang, Y., Liu, G., & Li, X. (2019). A 3D-CNN and LSTM based multi-task learning architecture for action recognition. *IEEE Access, 7*, 40757-40770

Pan, C., Li, X., Yan, W. (2018) A learning-based positive feedback approach in salient object detection. In *IVCNZ* (pp.1-6)

Pan, C. Yan, W. (2020) Object detection based on saturation of visual perception. Multim. Tools Appl. 79(27-28): 19925-19944

Parisi, G. I., & Wermter, S. (2013). Hierarchical SOM-based detection of novel behavior for 3D human tracking. In *International Joint Conference on Neural Networks*

Parker, J. R. (2010). *Algorithms for Image Processing and Computer Vision*. John Wiley & Sons

Peng, X., & Schmid, C. (2016). Multi-region two-stream R-CNN for action detection. In *European Conference on Computer Vision*

Plagemann, C., Ganapathi, V., Koller, D., & Thrun, S. (2010). Real-time identification and localization of body parts from depth images. In *IEEE International Conference on Robotics and Automation*

Pleshkova, S. G., Bekyarski, A. B., & Zahariev, Z. T. (2019). Based on artificial intelligence and deep learning hand gesture recognition for interaction with mobile robots. In *National Conference with International Participation* (pp. 1-4)

Popoola, O. P., & Wang, K. (2012). Video-based abnormal human behavior recognition—A review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(6), 865-878

Qian, R., Liu, Q., Yue, Y., Coenen, F., & Zhang, B. (2016). Road surface traffic sign detection with hybrid region proposal and Fast R-CNN. In *International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* (pp. 555-559)

Rahmani, H., & Bennamoun, M. (2017). Learning action recognition model from depth and skeleton videos. In *IEEE International Conference on Computer Vision*

Räty, T. D. (2010). Survey on contemporary remote surveillance systems for public safety. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40*(5), 493-515

Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review, 43*(1), 1-54

Ren, M., Kiros, R., & Zemel, R. (2015). Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*

Rodríguez-Moreno, I., Martínez-Otzeta, J. M., Goienetxea, I., Rodriguez-Rodriguez, I., & Sierra, B. (2020). Shedding light on people action recognition in social robotics by means of common spatial patterns. *Sensors, 20*(8), 2436

Roh, Y., Heo, G., & Whang, S. E. (2019). A survey on data collection for machine learning: a big data-ai integration perspective. *IEEE Transactions on Knowledge and Data Engineering*

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Bernstein, M. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211-252

Russo, M. A., Filonenko, A., & Jo, K. H. (2018). Sports classification in sequential frames using CNN and RNN. In *International Conference on Information and Communication Technology Robotics* (pp. 1-3)

Sadoughi, F., Kazemy, Z., Hamedan, F., Owji, L., Rahmanikatigari, M., & Azadboni, T. T. (2018). Artificial intelligence methods for the diagnosis of breast cancer by image processing: a review. *Breast Cancer: Targets and Therapy, 10*, 219

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117

Sermanet, P., Chintala, S., & LeCun, Y. (2012). Convolutional neural networks applied to house numbers digit classification. In *International Conference on Pattern Recognition*

Shao, Z., Cai, J., & Wang, Z. (2017). Smart monitoring cameras driven intelligent processing to big surveillance video data. *IEEE Transactions on Big Data, 4*(1), 105-116

Shen, D., Xin, C., Nguyen, M., Yan, W. (2018) Flame detection using deep learning. In *ICCAR* (pp.54)

Shinde, S., Kothari, A., & Gupta, V. (2018). YOLO based human action recognition and localization. *Procedia Computer Science, 133*, 831-838

Shi, Z., & Kim, T.-K. (2017). Learning and refining of privileged information-based RNNs for action recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition*

Si, C., Chen, W., Wang, W., Wang, L., & Tan, T. (2019). An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*

Song, C., He, L., Yan, W., Nand, P. (2019) An improved selective facial extraction model for age estimation. In *IVCNZ* (pp.1-6)

Strong, A. (2016). Applications of artificial intelligence & associated technologies. *Science, 5*(6)

Szeliski, R. (2010). *Computer Vision: Algorithms and Applications*: Springer Science & Business Media

Tu, Z., Xie, W., Qin, Q., Poppe, R., Veltkamp, R. C., Li, B., & Yuan, J. (2018). Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition, 79*, 32-43

Uddin, M. Z., Khaksar, W., & Torresen, J. (2017). Facial expression recognition using salient features and convolutional neural network. *IEEE Access, 5*, 26146-26161

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*

Wang, A., Zhang, W., & Wei, X. (2019). A review on weed detection using ground-based machine vision and image processing techniques. *Computers and Electronics in agriculture, 158*, 226-240

Wang, P., Li, W., Gao, Z., Tang, C., Zhang, J., & Ogunbona, P. (2015). Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring. In *ACM International Conference on Multimedia*

Wang, X., Yan, W. (2019) Human gait recognition based on SAHMM. *IEEE/ACM Transactions on Biology and Bioinformatics.*

Wang, X., Zhang, J., Yan, W. (2020) Gait recognition using multichannel convolution neural networks. *Neural Comput. Appl. 32 (18)*: 14275-14285

Wang, X., Yan, W. (2020) Cross-view gait recognition through ensemble learning. *Neural Comput. Appl.* 32(11): 7275-7287

Wang, X., Yan, W. (2020) Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. *Int. J. Neural Syst. 30(1)*: 1950027:1-1950027:12

Wang, X., Yan, W. (2021) Non-local gait feature extraction and human identification. *Multim. Tools Appl. 80(4)*: 6065-6078

Wang, Y., Zhou, W., Zhang, Q., Zhu, X., & Li, H. (2018). Weighted multi-region convolutional neural network for action recognition with low-latency online prediction. In *IEEE International Conference on Multimedia & Expo Workshops* (pp. 1-6)

Wigington, C., Stewart, S., Davis, B., Barrett, B., Price, B., & Cohen, S. (2017). Data augmentation for recognition of handwritten words and lines using a CNN-LSTM network. In *International Conference on Document Analysis and Recognition.*

Wong, S. Y., Yap, K. S., Zhai, Q., & Li, X. (2019). Realization of a hybrid locally connected extreme learning machine with DeepID for face verification. *IEEE Access, 7*, 70447-70460

Wu, D., Pigou, L., Kindermans, P.-J., Le, N. D.-H., Shao, L., Dambre, J., & Odobez, J.-M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(8), 1583-1597

Xiao, X., Xu, D., & Wan, W. (2016). Overview: Video recognition from handcrafted method to deep learning method. In *International Conference on Audio, Language and Image*

Processing

Xie, C., Li, P., & Sun, Y. (2019). Pedestrian Detection and Location Algorithm Based on Deep Learning. In *International Conference on Intelligent Transportation, Big Data & Smart City* (pp. 582-585)

Yan, Q., Gong, D., & Zhang, Y. (2018). Two-stream convolutional networks for blind image quality assessment. *IEEE Transactions on Image Processing, 28*(5), 2200-2211

Yan, W. (2019). *Introduction to Intelligent Surveillance Surveillance Data Capture, Transmission, and Analytics*, Springer.

Yan, W. (2021). *Computational Methods for Deep Learning Theoretic, Practice and Applications*, Springer.

Yang, X., Zhang, C., & Tian, Y. (2012). Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM International Conference on Multimedia*

Yao, B., Jiang, X., Khosla, A., Lin, A. L., Guibas, L., & Fei-Fei, L. (2011). Human action recognition by learning bases of action attributes and parts. In *International Conference on Computer Vision*

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. In *IVCNZ* (pp.1-6)

Zhang, B., Quan, C., & Ren, F. (2016). Study on CNN in the recognition of emotion in audio and images. In *IEEE/ACIS International Conference on Computer and Information Science (ICIS)* (pp. 1-5)

Zhang, L., Yan, W. (2020) Deep learning methods for virus identification from digital images. In *IVCNZ* (pp.1-6)

Zhang, L., Lin, L., Liang, X., & He, K. (2016). Is Faster R-CNN doing well for pedestrian detection? In *European Conference on Computer Vision*

Zhang, Q., Yan, W. (2018) Currency detection and recognition based on deep learning. In *AVSS* (pp.1-6)

Zhang, S., Wu, R., Xu, K., Wang, J., & Sun, W. (2019). R-CNN-based ship detection from high resolution remote sensing imagery. *Remote Sensing, 11*(6), 631

Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR), 52*(1), 1-38

Zhang, Y., Yan, W., Narayanan, A. (2017) A virtual keyboard implementation based on finger recognition. In *IVCNZ* (pp.1-6)

Zhao, G., Tian, Q., & Sun, M. (2019). A face verification system based on DeepID algorithm. *Information Technology and Informatization* (2), 32

Zheng, K., Yan, W., Nand, P. (2018) Video dynamics detection using deep neural networks. *IEEE Trans. Emerg. Top. Comput. Intell.* 2(3): 224-234