Deep Learning Methods for Human Behavior Recognition

Jia Lu

A thesis submitted to the Auckland University of Technology

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer and Information Sciences

2020

School of Engineering, Computer & Mathematical Science

Abstract

With the decreased costs of security monitoring equipment such as cameras, video surveillance has been broadly applied to our communities and public places. However, at present most of the surveillance systems acquire anomalies and visual evidences only through video playback. Hence, it is necessary to develop the methods of real-time human behavior recognition so as to reduce security staff's workload and improve work efficiency.

The existing work needs feature extraction from the video frames to detect human body and achieve human behavior recognition. In this thesis, our focus is on the state-of-the-art methods for human behavior recognition based on deep learning. Since deep learning methods have been well investigated in the past decades, as an end-to-end computational method, it simplifies feature extraction as the operations in a black box.

In this thesis, we explore and exploit the state-of-the-art methods, which are utilized for human behavior recognition. More importantly, in order to attain our goal, spatiotemporal information was collected and employed to the implementation of our research project. We firstly adopted ensemble learning with deep learning methods. We proposed Selective Kernel Network (SKNet) and ResNeXt with attention mechanism, which generate positive results to recognize human behaviours.

The contributions of this thesis are: (1) The ResNeXt and SKNet with attention mechanism make the best accuracy of overall human behavior recognition at the rate up to 98.7% based on public datasets; (2) The YOLOv3 + LSTM network to reply on both spatiotemporal information with class score fusion is able to achieve 97.58% accuracy based on our dataset for sign language processing.

Keywords: Deep learning (DL), Convolutional neural network (CNN), Long short-term memory (LSTM), You Only Look Once (YOLO), Ensemble learning, Selective kernel network (SKNet), Attention mechanism

Table of Contents

List of Figures

List of Tables

List of Algorithms

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: 14 September 2020

Acknowledgment

This research work was fulfilled as the part of the Doctor of Philosophy in Computer and Information Sciences at the School of Engineering, Computer & Mathematical Sciences (SECMS) in the Faculty of Design and Creative Technologies (DCT) at the Auckland University of Technology (AUT) in New Zealand. I would like to deeply appreciate my parents for the financial support they provided during my entire time of academic study in Auckland. I am also obliged to my homestay who supports me in Auckland living and language study.

My deepest thanks should be given to my primary supervisor Dr. Wei Qi Yan who has offered me with much technological guidance and support. I believe that I could not achieve my PhD degree without his invaluable support and supervision. In addition, I am grateful to my secondary supervisor Dr. Minh Nguyen and school administrators for their support and guidance through my PhD study in the past few years.

Jia Lu

Auckland, New Zealand

September 2020

Chapter 1
Introduction

The first chapter of this thesis consists of five sections. In the first section, the background and motivation of this thesis are introduced, human behavior recognition using deep learning methods as an active research topic can attain the recognition in real time, which has the functionality to enhance work efficiency. Our objectives will be discussed in the fourth section. This chapter also covers the details of research questions after in-depth understanding of the relevant literatures and research background. Finally, an overview of the structure of this thesis will be presented.

1.1 Background and Motivation

With decreased costs of digital monitoring equipment such as cameras, video surveillance has been broadly applied to public places such as banks, transportation, shopping malls, etc. which allows security staff to monitor abnormal events. However, at present most of the video surveillance software platforms are still being run in traditional mode, which acquire anomalies and visual evidences only through offline videos. It is challenging to make real-time alarms, pop up notifications, and monitor incident scenes uninterruptedly. Hence, it is necessary to develop novel methods for real-time human behavior recognition so as to curtail the security staff's workload and ameliorate the work efficiency.

Moreover, deep learning models have exhibited excellent capabilities in human behavior recognition, how to make the model more stable and robust for human behavior recognition has become the new challenge of this study. In a nutshell, this research project aims to develop the cutting-edge deep learning methods so as to conduct human behavior recognition without too many manual interruptions. As the outcome of this research work, we anticipate reaching an overall up to 90% accuracy of recognition in real time.

In intelligent surveillance, human detection (Chu et al., 2020; Huang et al., 2020; Zhang et al., 2020; Luo et al., 2020; Wu et al., 2020; Jiang et al., 2020; Yang et al., 2020; Bai et al., 2020; Varamesh et al., 2020), motion capture (Kiciroglu et al., 2020; Zhang et al., 2020; Xu et al., 2020; Habermann et al., 2020;), human re-identification (Ahmed et al., 2020; Zeng et al., 2020; Huang et al., 2020; Liu, Chang, & Sheng, 2020; Zhou, Su, & Wu, 2020; Jin et al., 2020; Zhang et al., 2020; Chen et al., 2020; Yang et al., 2020; Yan et al., 2020; Zhang et al., 2020; Chen et al., 2020; Lin et al., 2020; Wang, & Zhang, 2020; Gao, Wang et al., 2020; Zhai et al., 2020), gait recognition (Li et al., 2017; Wang et al., 2019; Fan et al., 2020; Li et al., 2020; Liu et al., 2020; Li et al., 2020; Wang et al., 2020) and human behavior analysis (Yan, 2021; Lu, Yan, & Nguyen, 2020) are employed to analyze and identify individuals in various scenes.

The term of action is described as an atomic movement at the limb level, it consists a diversity of movements, which include simple movements and cyclic body movements. The behavior is treated as a number of subsequent actions (Poppe, 2010). Thus, human action recognition is the fundamental of human behavior analysis, the process of behavior analysis is to monitor and analyze the undergoing actions. Monitoring the behaviors of pedestrians is the key task in human behavior recognition. The process of monitoring pedestrian is to capture the relevant context information so that the system is able to identify the behaviors of the pedestrian. There are two different recognition types, which include vision-based behavior recognition and sensor-based behavior recognition. In this thesis, our main focus is on the vision-based behavior recognition. The vision-based behavior recognition utilizes visual sensing devices, the output data of devices is either a video sequence or digitized visual data (Chen & Nugent, 2019). The surveillance system contains object detection, recognition, tracking, etc. (Li, Zhang, Yan, & Klette, 2016; Yan, 2019), for traditional machine learning, the methods in vision-based behavior recognition are summarized as feature extraction, background segmentation, and pattern recognition.

The investigation of human behavior in video footages is a research redhot topic in the field of vision-based surveillance. The most recent projects such as “Green Light Project” in Detroit, USA and “Skynet Project” in China, which have implemented the functionalities of human behavior recognition in intelligent surveillance for public health, security, and safety. In these projects, human behaviors are not only for individuals (e.g., running, fainting, walking, etc.), but also for multiple people in crowded events (e.g., talking, fighting, thefting, etc.). Moreover, depending on outdoor environments, human behaviors also are grouped into two categories: Normal and abnormal behaviors (Xiang & Gong, 2005).

However, all of these projects require large amount of human labors, which are time-consuming and inefficiency. The methods for video surveillance (Aggarwal & Cai, 1997; Xiang & Gong, 2005; Zheng, Shen, Hartley, & Huang, 2010; Aggarwal & Ryoo, 2011; Cui, & Yan, 2016; Gu, Nguyen, & Yan, 2016; Lu, Shen, Yan, & Bacic, 2017; Zheng, Yan, & Nand, 2017) request multiple models to work together. A BCM-spiking neural

network was proposed for human behavior recognition, but the limitation is that the model can only achieve the feat of single person recognition (Meng, Jin, & Yin, 2011).

Deep learning is a part of the contents of machine learning, as the state-of-the-art technology, deep learning (DL) becomes more and more popular because of its superiority to conventional machine learning (Yan, 2021). Deep learning methods (LeCun, Huang, & Bottou, 2004; Hinton, Osindero, & Teh, 2006; Sarikaya, Hinton, & Deoras, 2014) were implemented not only for the vision-based object detection, but also for text-based natural language processing (NLP) as well as speech recognition. Moreover, deep learning as an end-to-end model normally does not require low-level processing which is able to cut off human labor and gain time efficiency though the training is costly.

Recent research work related to human behavior recognition is primarily based on deep learning. Ji et al. propounded a convolutional model for human behavior recognition which imported both spatiotemporal information into a 3D neural network. Moreover, the handcrafted features were not required, but a large number of labelled samples are needed (Ji, Xu, Yang, & Yu, 2013). A spatiotemporal CNN was put forward in order to recognize 3D human activity which applies grayscale images as the inputs (Wang, Wang, Lin, Wang, & Zuo, 2014). Dobhal et al. suggested to use Gaussian mixture model (GMM) to snatch the binary frames of a person in foreground, the binary motion images (BMI) were taken into account to feed the CNNs through training and testing processes for single person behavior recognition (Dobhal, Shitole, Thomas, & Navada, 2015).

In 2016, a hierarchical model in deep temporal based on long short-term memory (LSTM) was recommended to recognize multi-person behavior, two LSTM models were adopted for representing the action of single person in sequence and aggregating the information of single person for behavior understating (Ibrahim, Muralidharan, Deng, Vahdat, & Mori, 2016). Osayamwen et al. set forth that the softmax is normally applied to CNNs as the traditional loss function which is able to group different classes and achieved an effective training model. Thus, a Bayesian distribution-based regularization was accommodated in order to estimate the class probability, which uplifted the class

discrimination, maximized the distance between human behaviors, and minimized the distance between same behaviors (Osayamwen & Tapamo, 2019).

Moreover, most of deep learning models require either deepened network layers or widened layers to betterment the accuracy of the model. However, as the number of hyperparameters increases, i.e., the number of channels, filter size, etc., the difficulty and computational costs of the deep learning networks will be also surged. Thus, the state-of-the-art methods in deep learning for classification like ResNeXt model (Xie et al., 2017), Convolutional Block Attention Module (CBAM) (Woo et al., 2018), Selective Kernel Networks (SKNet) (Li et al., 2019) were investigated and implemented in our literature review. In the end, the results reflecting the positive accuracy are compared with previous methods for human behavior recognition, which will be further analyzed and discussed. A spatiotemporal convolutional network for human behavior recognition (ST-GCN) (Yan et al., 2018) was offered by automatically learning both the spatiotemporal patterns from visual data to achieve the human behavior recognition.

In this thesis, we design and implement deep learning methods which are time efficiency and outperform in model training and testing. In this thesis, data preparation is the most critical step for this project. We mainly work with public datasets to implement our deep learning models, which include Weizmann dataset and KTH dataset. Four deep learning models were probed in this study. YOLOv3 model was implemented for human behavior recognition, which achieved up to 96.29% mAP (average precision) based on Weizmann dataset and 84.58% mAP based on KTH dataset.

1.2 Research Questions

Human behavior recognition as an active research field has been investigated in the past decades. With the increase of population, more and more research attention has been concentrated on the issues related to public health, safety, and security. Therefore, how to find security problems in time, effectively control any incidents, and quickly resolve the issues has become one of the major challenges that all governments have to face. Intelligent surveillance, as an effective mean of security, can reliably and effectively carry out the mission of monitoring and protecting important spots, it has been widely employed to military, banking, shopping malls, and transportation stations, etc. In previous work, the focus was on traditional approaches which require several models to achieve human behavior recognition. However, with the increase of our computing capability, deep learning has become the mainstream to resolve the relevant issues. Thus, our research questions in this project are:

Question 1:

Traditional machine learning methods already have exhibited their performance in human behavior recognition. So, what are the advantages and disadvantages by bringing in deep learning methods?

Question 2:

Which kind of deep learning methods can be implemented for human behavior recognition? How about the research outcomes?

Question 3:

How to construct deep neural networks and make them more accurate, robust, and stable?

Question 4:

How to reduce the misclassification of similar behaviors in deep learning?

Question 5:

How to resolve the evaluation problem that the accuracy of specific behavior recognition is relatively low?

1.3 Contribution

The focus of this thesis is principally on the implementation of human behavior recognition based on deep learning methods. The contribution of this thesis is to apply the end-to-end methods that are developed step by step according to the processes which are presented in this thesis. The methodology in deep learning for human behavior recognition encompasses four parts in this thesis: (1) data collection, (2) training the deep learning model, (3) fine-tuning the model, (4) resultant analysis for human behavior recognition. The proposed method and implementation will be delineated in Chapter 3.

Moreover, the algorithms which are developed in this thesis will be also analyzed, the methods export positive results and suit for human behavior recognition. A comparison of various state-of-the-art deep learning methods is presented in Chapter 2. The introduction of various approaches will also be detailed in the following chapter.

Furthermore, in this thesis, we work for human behavior recognition with multiple deep learning methods. This thesis also presents complete and detailed information for human behavior recognition using Python as a programming language correspondingly.

Last but not least, multiple deep learning methods will be presented and compared with machine learning methods which were set forth by other researchers.

1.4 Objectives of This Thesis

Firstly, this thesis introduced various machine learning methods and the state-of-the-art methods in deep learning which are able to be implemented for human behavior recognition. The comparison for both machine learning methods and deep learning methods will be iterated in this thesis. In addition, the outcomes of these methods will

also be evaluated in this thesis. Moreover, our proposed methods will also be studied and evaluated, the deep learning methods and our proposed methods will be compared in this thesis to test the stability and robustness.

Secondly, in order to achieve the recognition for human behaviors in real time, a deep learning framework is presented for further understanding. Therefore, the overall objective of this thesis is split into four parts which include data collection, data augmentation, models training and fine tuning, resultant evaluations, and analysis for human behavior classification. Moreover, we also collect our own datasets to test the proposed deep learning methods that can be used in real scenarios. The data augmentation is used to increase the amount of training data so as to improve the generalization ability and robustness of our proposed model, also prevent the over fitting problem. The fine tuning was selected to use for initializing our model to speed up our training time, also to overcome the small dataset size.

Ultimately, deep learning methods and traditional machine learning methods for human behavior recognition will be compared, the comparison of different deep learning methods and proposed methods for human behavior recognition will be presented in this thesis. Moreover, multiple methods will be justified according to their performance using empirical metrics in this thesis.

1.5 Structure of This Thesis

The thesis is structured as follows:

In Chapter 2, a literature review will be depicted, such as previous studies of traditional machine learning and deep learning methods. Moreover, typical models in deep learning for human behavior recognition will be also exposited in this chapter. Thus, in Chapter 2, we will mainly narrate the fundamentals of deep learning methods and the state-of-the-art methods, which is able to be applied to human behavior recognition.

In Chapter 3, the research methodology of this thesis will be stated. Moreover, potential solutions and answers will also be elaborated to answer the research questions in Chapter 1. Moreover, data collection and data preparation as well as data augmentation, will also be outlined for this research topic. The implementations with the evaluation metrics will also be demonstrated at the end of this chapter.

In Chapter 4, our deep learning methods, proposed in Chapter 3, will be implemented, and evaluated, experimental outcomes will be expounded with the support of tables and figures. Moreover, the limitations of this research will be addressed as well.

In Chapter 5, the discussions and analysis are depicted based on experimental results and outcomes we acquired in Chapter 4. Finally, the conclusion and future work will be set forth in Chapter 6.

Chapter 2
Literature Review

With the in-depth analysis of the research questions and related work, the focus of this thesis is on the methods in deep learning for human behavior recognition in real time. For example, the previous methods in deep learning are reviewed; moreover, the comparisons of traditional machine learning and deep learning methods are conducted. The state-of-the-art methods in deep learning for human behavior recognition will be summarized in this chapter. We will propose our methods in the next chapter to solve the problems which are found in this chapter.

2.1 Introduction

Surveillance cameras are ubiquitous in our society which have been utilized in both private premises such as highly secure areas and public precincts like streets, etc. (Petrushin, 2005; Chen & Zhang, 2006; Popoola & Wang, 2012). Moreover, with the increasing population, public security and safety become more and more crucial. Human behavior recognition as an active research field in intelligent surveillance has received extensive attention. Successfully solving the problems of human behavior recognition benefits not only for public security (abnormal scenarios), but also for private safety (smart homes). The key task of this research project is to extract visual information from digital video footages; more specifically, Region of Interest (ROI) for human body needs to be detected. A myriad of digital video and image processing techniques were offered to extract and analyze the true characteristics of human behaviors. It needs the technologies from digital image processing, pattern recognition, machine learning, and deep learning, etc.

With the rapidly increasing number of surveillance cameras in various scenarios, locating and extracting the region of interest (ROI) from thousands of surveillance videos have become prominent problems. Along with the continuous expansion of surveillance systems, a vast number of video footages have been archived, which become more and more tough to obtain meaningful and valuable information from the large amount of visual data, identifying a visual object is becoming less efficient. Therefore, how to make surveillance more efficiency in the environment with tremendously big data needs to be taken into consideration.

Video surveillance has been explored and exploited for decades, research work based on video analysis such as VSAM (visual surveillance and monitoring), IVPL (image and video processing laboratory), AIRVL (artificial intelligence, robotics, and vision laboratory), etc. has taken a great step. Vehicles and pedestrians along with object trajectories (Krumm, et al., 2000; Siebe & Maybank, 2002; Chen & Zhang, 2006) have made digital surveillance more “intelligent”. Intelligent surveillance is able to analyse

streaming videos automatically without human interventions and achieve further amazing research outcomes such as image segmentation and scene understanding.

Most of the surveillance systems only pay attention to detect, recognise, and track objects; however, the actions and activities of visual objects were not fully taken into consideration, which makes surveillance not intelligent enough. For the approaches in human behavior recognition, both low-level (without semantic understanding) and high-level processes (with semantic understanding) are needed. The low-level process assists to localize the region of interest, extract features, and minimize the useless information from video footages, the high-level process analyses these features to attain human behavior understanding ultimately. Generally speaking, visual features such as edge, shape, statistical features are described and adopted for pedestrian detection. Also, the representative features such as Harr wavelets (Gerónimo, López, Ponsa, & Sappa, 2007), Histogram of Oriented Gradient (HOG) (Dalal & Triggs, 2005), Local Binary Pattern (LBP) (Zheng, Shen, Hartley, & Huang, 2010) and edgelet features (Wu & Nevatia, 2005) exhibit positive outcomes for pedestrian detection as well.

Nowadays, with the growing capacity of computing devices, deep neural networks (DNNs) have gained massive attention to detect visual objects, which lead to a new era of computer vision (Lu, Shen, Yan, & Bacic, 2017; Asadi-Aghbolaghi, Clapes, Bellantonio, Escalante et al., 2017; Herath, Harandi, & Porikli, 2017; Zhang, Yan, & Narayanan, 2017; Pan, & Yan, 2019; Pan, & Yan, 2020; Lu, Yan, & Nguyen, 2020; Yu, & Yan, 2020). DNNs (Hinton, Osindero, & Teh, 2006; Szegedy, Ioffe, Vanhoucke, & Alemi, 2017) encapsulate multiple hidden layers, the pretraining method is adopted to resolve the problems of an optimal local solution, the number of hidden layers is run up with the “depth” in the neural network model. The state-of-the-art methods heavily rely on artificial neural networks, such as convolutional neural networks (CNNs), R-CNNs, Fast R-CNNs, Faster R-CNNs, and single shot multi-box detector (SSD). Moreover, deep learning was implemented in both supervised and unsupervised learning (Ji, Xu, Yang, & Yu, 2014). Apparently, the work has unmistakably unveiled the differences between deep neural networks and shallow neural networks in various aspects (Liu, et al., 2016).

In human behavior recognition, bounding box was taken into consideration for deep neural networks to resample the proposed pixels. Because there are inherent local patterns in an image such as eyes, nose, mouth, etc., CNNs were derived from combining digital image processing and artificial neural networks. Traditional CNNs enclose multiple convolutional layers and subsampling operations, the outputs of convolutional layers are extracted as the feature maps which are flattened and fed into a fully connected layer.

CNN links the upper and lower layers through convolution kernels. A convolutional neural network is a deep neural network with a convolutional structure. The convolutional structure diminishes the requirements of memory space occupied by normal neural networks. It has three crucial operations that are local receptive fields, weight sharing, and pooling layer, which effectively trims the number of network parameters and alleviates the problem of overfitting the model. Convolutional neural network is a kind of multilayer neural networks. The convolutional layer and pool sampling layer are the core modules to carry out the feature extraction of convolutional neural networks.

This network model takes advantage of gradient descent to minimize the loss function so as to adjust the weights in the network layer by layer and heighten the accuracy of the network through frequent iterative training. The low hidden layer is comprised of max pooling layer and sampling layer. The high hidden layer is the layer being employed as logistic regression classifier of the fully connected layer corresponding to the traditional multilayer perceptron. The input of the first fully connected layer is an image generated by using feature extraction from the convolutional layer and the subsampling layer. The output layer is a classifier that combines logistic regression, softmax regression, or the support vector machine (SVM) to classify the input image.

The structure of convolutional neural networks includes convolutional layer, pooling layer, and fully connected layer. Each layer has multiple feature maps, each feature map extracts a feature of the input through a convolution filter, each feature map has multiple

neurons. After the input images are convolved or filtered, the local feature is extracted. Once the local feature is obtained, its positional relationship is also determined.

The input of each neuron is connected to the local receptive field of the previous layer. Each feature extraction layer is followed by using a calculation layer to obtain local average and secondary extraction. It is also called a feature mapping layer. Each layer of the network consists of multiple feature mapping planes. A connection between neurons has one weight. The mapping of the input layer to the hidden layer is usually called a feature map, that is, a feature extraction layer is acquired through a convolution layer, and a feature mapping layer is obtained after pooling operations.

The core idea of convolutional neural networks is local receptive field, weight sharing, and pooling operations, in order to achieve simplified network parameters and make the network having a wide degree of displacement, scale, scaling, nonlinear deformation stability. The local receptive field is vital because the spatial connection of the image parts is partial, each neuron does not need to scan the entire image, it only requires finding the local features, these local features are integrated at a higher level. Thus, the global information will be gained, which greatly diminishes the number of neuron connections. Parameter sharing amongst multiple neurons cuts off the number of parameters, multiple feature maps are thus got by deconvoluting images with multiple filters.

In fact, weight sharing is to convolute the image with the same kernel, which means, all neurons in the first hidden layer execute exactly the same operations at various locations of the input image. Weight sharing is to detect the same type of features in different positions, that is, convolution networks are adaptive to the small range of image translation, which has better translation invariance. Moreover, compared with general artificial neural networks, convolutional neural network has the following advantages: Firstly, the structure of CNNs is apt to the structure of image; secondly, feature extraction and classification are carried out to make feature extraction better for feature classification at the same time; thirdly, weight sharing can trim off the total number of

parameters of a neural network and make the structure of the neural network simple and adaptive.

The same convolution kernel is shared among all images, the images still retain the original position after convolution operations. For the typical CNNs, ReLU (Nair, & Hinton, 2010) as the activation function, it can alleviate the problem of gradient vanishing, by using ReLU function, the network is able to converge faster and also sparsely be activated. The definition of ReLU function is showed in Eq.(2.1.1).

$$f(x) = max(0, x) \tag{2.1.1}$$

The pooling layers as one of the important components in CNNs which are applied to compress data and parameters so as to lessens overfitting. Through compressing the input feature map, on the one hand, it makes the feature map smaller and simplifies the computational complexity of the neural network; on the other hand, it performs feature compression and extracts the principal features.

There are two general types of pooling operations, one is average pooling and the other is max pooling, where the average pooling performs the downsampling by dividing the input into rectangular pooling regions and then calculating the average values of each region. The max pooling carries out downsampling by splitting the input into rectangular pooling regions and calculating the maximum values of each region. In general, the max pooling is much effective for the model. Although both max pooling and average pooling are able to fulfil data downsampling, max pooling much likes feature selection, selects features for better classification and recognition, and provides nonlinearity. The fully connected layers are as same as the typical ANNs, which will output an *n*-dimensional vector.

Moreover, region proposal methods and region-based convolutional neural networks (R-CNN) are most successful with high precision in object detection, the advantage of this model is that feature maps are initially generated by using semantic segmentation,

which are able to reflect the image content by adopting region proposal to ensure a high recall, when selecting fewer windows (Ren, He, Girshick, & Sun, 2017).

Faster R-CNN is based on selective search (Uijlings, Sande, Gevers, & Smeulders, 2013) and leads to outstanding achievement in a series of competitions (Ren, He, Girshick, & Sun, 2017; He, Zhang, Ren, & Sun, 2016), which is an object detection algorithm that utilizes a variety of selection strategies and merges the final results. Region proposal network (RPN) is applied to train data in the end-to-end way and generate the accurate location of regions. The method merges RPN and Fast R-CNN into a single network which is able to accelerate the detection (Ren, He, Girshick, & Sun, 2017). Compared to traditional approaches, deep learning approaches accomplish the end-to-end processing along with higher classification confidences which also show an invariant to illumination, pose, etc. (LeCun, Huang, & Bottou, 2004; Lee, Gallagher, & Tu, 2016).

In this thesis, we will introduce not only traditional machine learning, which was offered to human behavior recognition, but also deep learning. After all, we are keen on developing a new methodology using deep learning for detecting various human behaviors so as to attain real-time recognition. This research work is based on two public datasets and our own dataset, four deep learning methods were selected for the experiments in order to compare the performances. Meanwhile, deep learning model based on attention mechanism is also investageted in this thesis.

2.2 Machine Learning

With the development and applications of computer science and artificial intelligence, digital video analysis is rising rapidly. One of the core applications in video analysis is human behavior recognition, as the active research field has attracted much attenction in the intelligent surveillance, owing to the surge of security issues. The accuracy and rapidity of behavior recognition will directly influence the follow-up work of video analysis. Therefore, how to improve the accuracy and speed of human behavior recognition by using digital videos has become a key issue in video analysis.

Traditional machine learning methods for human behavior recognition require multiple processes to be combined together, which encapsulates preprocessing, feature extraction, training and classification, etc. Moreover, preprocessing and feature extraction play a pivotal role in human behavior recognition, which directly takes actions based on the final results.

According to the existing research work (Aggarwal & Ryoo, 2011), there are multiple approaches to attain human behavior recognition through traditional machine learning, which is grouped as single-layered approaches, hierarchical approaches, and histogram-based approaches. Single-layer approaches are directly utilized the image sequences to recognize human behaviors, which are more suitable for recognizing behaviors with sequential characteristics; Hierarchical approaches aim to describe high-level human behaviors by split the simple behaviors into sub-events, so that complex behaviors can be analyzed and understanded; Histogram-based approaches are proposed and well-studied since it have the scale invariance and rotation invariance when the gradient direction is unified. Each approach shows the positive results in human behavior recognition, lays a solid foundation for human behavior recognition.

Typical machine learning methods for human behavior recognition mainly include interest points, dense tracks, etc. The key points are accommodated to recognize human behaviors by detecting the corner points in video and extracting the features of the corner points. However, corner points are generated by fully suppressing background noises, which will not only influence the final results, but also curtail the speed of recognition.

Dense track is to take dense samples on multiple scales into consideration, then trace the sampled points so as to get the track, and finally extract the characteristics of the track for behavior recognition. However, this method has highly computational complexity and intensity for high dimension features, which will take up a wreath of memory, so it is difficult to achieve real-time recognition. In 2013, Wang et al. proffered a dense trajectories (DT) approach (Wang, Kläser, Schmid, & Liu, 2013) using optical flow to get the motion of videos, and extract features along the trajectory to attain the action

recognition. Moreover, Wang et al. enhanced the propounded dense trajectory (iDT) approach (Wang, & Schmid, 2013), the main improvements are optimization of optical flow, feature regularization, and feature encoding. These betterments have greatly uplifted the algorithm, the accuracy of UCF50 dataset has grown from 84.5% to 91.2%.

2.2.1 Motion Detection

Motion detection and tracking are affected by complexity and variety of the environment. Thus, none of the current techniques will perfectly fit all the environments. Moving object detection as the primary step for further analysis should be considered first (Papageorgiou, Oren, & Poggio, 1998; Poppe, 2007; Joshi & Thakore, 2012), it will directly affect the final recognition results. Moreover, analysing image sequence is the predominant objective to detect moving objects which relates to a background scene (Elhabian, El-Sayed, M., & Ahmed, 2008), video footages are regarded as the combination of continuous static frames which can be tackled by using image processing methods as well. The major object detection approaches which contain background subtraction, temporal differencing, and optical flow together (Kulchandani & Dangarwala, 2015).

The background subtraction, as the most straightforward method by subtracting current frame and background frame, it is able to be utilized in the complex background and adopted for the static cameras (Wren, Azarbyejani, Darrell, & Pentland, 1997; Piccardi, 2004). However, for outdoor environments such as bad weather, lighting, etc., background subtraction becomes less sensitive (Rakibe & Patil, 2013). In consecutive images, time difference between two adjacent frames is equivalent to extract motion region in the images.

Optical flow detects moving objects by using changes of pixels in temporal domain and the correlations between adjacent frames. In 2000, a motion foreground detection algorithm based on morphological changes was proposed (Stringa, 2000). Brendel *et al*. (Brendel & Todorovic, 2009) implemented the Dynamic Time Warping algorithm for visual object segmentations. Zhou *et al*. (Zhou, Xu, Tao, & Gong, 2005) proposed a

spatiotemporal model by utilizing Markov random field (MRF) method to represent the changes of foreground and background. However, for complex scenes, the single background extracted from a video sequence is no longer applicable. Therefore, the methods for modelling complex background have been investigated. Haritaoglu *et al.* (Haritaoglu, Harwood, & Davis, 1998) suggested an algorithm which works for the scope of grayscale intensities, the background was modelled by using the minimum and maximum intensity. The mixture of Gaussian model was proposed (Stauffer & Grimson, 1999) which utilizes Gaussian mixture distributions as statistical models for each pixel. The parameters of each Gaussian distribution are updated continuously to generate the gradual changes of the background, which solved the problem of multimodal distribution of pixels related to lighting changes in background.

2.2.2 Human Behavior Recognition in Machine Learning

After found the region of interest (ROI), the high-level processing for semantic and annotation of human behavior will be processed to achieve our goal, which encloses the recognition and classification processes to finally export the results. Normally, ROI as the global representation is obtained from background subtraction which includes silhouettes, edges, and optical flow.

Both silhouette and shape are applied to describe actions, two templates based on contour-based mean motion shape (MMS) and motion-based average motion energy (AME) were employed to extract features, the nearest neighbor classifier was taken for recognizing human actions (Wang, & Suter, 2006). In 2007, Wang et al. suggested silhouettes feature extraction and then use HMM to achieve human action recognition (Wang, Huang, & Tan, 2007). Euclidean distance was used to measure the similarity of two silhouettes (Weinland, Boyer, & Ronfard, 2007), Chamfer distance was adopted which eliminates the preprocessing step of background subtraction (Weinland, & Boyer , 2008).

In addition to use silhouette information, motion information of human body is often calculated, such as background difference, optical flow etc. When the background difference method cannot work well, the optical flow method was taken into account. However, it also is affected by noises. Optical flow is calculated at the center of human body, which effectively low down the impact of noises (Efros, Berg, Mori, & Malik, 2003).

In 3D space, spatiotemporal volume (STV) is gained by using given data. Blank et al. obtained STV from silhouette information in video sequence for the first time (Blank, Gorelick, Shechtman, Irani, & Basri, 2005; Blank, Gorelick, Shechtman, Irani, & Basri, 2007). A series of STVs have been offered for each video, each STV only covers part of the information in temporal domain (Achard, Qu, Mokhber, & Milgram, 2008). Ke et al. combined the silhouette with optical flow as the global features to get human behavior recognition (Ke, Sukthankar, & Hebert, 2007). The 2D SURF features (Bay, Tuytelaars, & Van Gool, 2006) are extended to 3D, which earned 84.26% total accuracy based on KTH dataset, each cell of these eSURF features contains the sum of Harr-wavelet features (Willems, Tuytelaars, & Van Gool, 2008).

Laptev extended Harris corners to 3D. The pixel intensities of the neighborhood of these spatiotemporal feature points have significant changes in the spatiotemporal domain (Laptev, 2005). Laptev et al. took use of local histogram of oriented gradient (HOG) and histogram of oriented optical flow (HOF) to recognize human behavior based on KTH dataset which reaches up 91.8% accuracy (Laptev, Marszalek, Schmid, & Rozenfeld, 2008). Klaser et al. increased the dimensionality to 3D, which achieved 91.4% accuracy based on KTH dataset (Klaser, Marszalek, & Schmid, 2008). The same work was investigated by Scovanner et al., which proposed a 3D SIFT, and achieved 82.6% accuracy based on Weizmann dataset (Scovanner, Ali, & Shah, 2007). Wang et al. applied bag-of-features (BOF) with SVM for human behavior recognition, in the most cases, the best description operator is a combination of gradient and optical flow (Wang, Ullah, Klaser, Laptev, & Schmid, 2009).

2.3 Deep Learning

Artificial neural networks (ANNs) were deeply investigated in past decades (Sondak & Sondak, 1989; Bajpai, Jain, & Jain, 2011), the networks fully take advantage of human brain neurons from the aspect of information processing, establish simple models, construct networks following the neuron connection methods. A neural network is an operational model for information processing using neural networks, which consists of a large number of neurons connected to each other. Each node represents a specific function, called activation function. The connection between two neurons is represented by using a weight for passing the connected signal, which is stored in the memory of artificial neural networks. The output of the network varies depending on the connection, weight, and activation function of the network. Most of ANN models are based on supervised learning.

In 1943, McCulloch and Pitts put forward the first artificial neuron model (M-P model), known as linear threshold gate (McCulloch & Pitts, 1943). The M-P model is the abstractive and simplified one that was constructed following the structure and principle of biological neurons. In the M-P model, x_1, x_2,… , x_n are a set of inputs of neurons, w_1, w_2,… , w_n represent the weights associated with the inputs, θ represents the bias, f stands for the active function, y is the output of neurons which means the binary value. The model is shown as Eq. (2.3.1),

$$y = f(\sum_{i=1}^{n} x_i w_i - \theta). \tag{2.3.1}$$

Moreover, each neuron in the neural network receives the output value of the upper layer as the input value of this neuron, transfers the input value to the next layer, the input layer will directly transfer the input attributes to the next layer (hidden layer or output layer). In multilayer neural networks, there is a functional relationship between the output of the upper node and the input of the lower node. This function is called activation function which is one of concepts of ANNs and DNNs. The activation functions include

sigmoid function (Hahnloser & Seung, 2006), hyperbolic tanh function, ReLU function (Hahnloser, Sarpeshkar, Mahowald, Douglas, & Seung, 2000; Hahnloser. & Seung, 2002). In literatures, sigmoid and hyperbolic tanh functions were normally treated as activation function, ReLU function mostly is applied to the multilayer neuron networks. Sigmoid is a nonlinear activation function, transfers the continuous real value of the input into the output between 0 and 1,

$$f(z) = \frac{1}{1+e^{-z}}. \tag{2.3.2}$$

In deep neural network, gradient exploding and gradient vanishing are frequently occured when the gradient is transferred reversely, the probability of gradient exploding is very small, but the probability of gradient vanishing is relatively large. Also, sigmoid function is not zero-centered activation, which will affect the gradient.

Hyperbolic tanh function is picked up in order to solve the gradient problems, the output of hyperbolic tanh function is zero-centered between -1 to 1. This function is taken affects by using exponentiation (an arithmetic operation on numbers), which ramps up the training time.

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2.3.3}$$

ReLU function $f(x) = \max(0, x+)$ as the most frequently-used activation function, the convergence rate of SGD algorithm is faster than sigmoid and tanh functions. According to Eq. (2.1.1), if $x > 0$, it does not have gradient vanishing and gradient exploding problems. Moreover, it does not require the exponentiation, which greatly cuts off the training time.

The perceptron (Rosenblatt, 1958) was profferd to enhance the M-P model; it adds extra input to represent the bias, the weights can be adjusted. Compared to the models, the perceptron model requires identical weights and thresholds. Moreover, in the perceptron model, weights can be both positive and negative. The basic perceptron model

encloses two layers of neurons, the input layer receives external data and then transfers it to the output layer.

Generally, in order to train and learn neural networks, a set of input data and output data are required. After the neural network gets converged, the weights are corrected according to the errors between the actual output and the expected output. One of the most important capabilities of neural networks is its ability to learn from the environment by continuously adjusting the weights and thresholds of the neurons till the output error of the network attains to the desired level. However, once the network is well trained, the weights cannot be changed; by adding new data into the network, the model needs to be retrained (Miller & Khan, 2011). The basic ANNs like multilayer neural networks encapsulate three or more layers including input layer, output layer, and one hidden layer at least. The neurons in each layer are fully connected with the ones in the next layer, there is no connection in the same layer or cross layer between neurons.

Backpropagation (BP) in ANNs has been successfully applied to various applications, typically including speech recognition, pattern recognition, and object recognition, etc. Through complicated learning rules, BP algorithms overcome the limitation of single layer network. With the learning rules, it will lead to a great success of multilayer neural networks. A BP algorithm contains both forward and backward passes. In the forward pass, the input pattern from the input layer is processed layer by layer through the hidden layer and the output layer; the weights will be calculated by using the connections between proceed layer and next layer. The state of neurons in one layer only influences the state of neurons in the next layer. Moreover, if the desired output cannot be obtained in the output layer, it will be transferred to the back pass, the error is returned along the original path; by iterating the weights of each neuron, the error will be minimized.

Moreover, the gradient descent is adopted in deep neural networks in order to decrease the output errors of the Root Mean Square (RMS) as Eq. (2.3.4), where d_j represents the desired output of the input pattern, y_j represents the output.

$$E_k = \frac{1}{2}\sum_{j=1}^{l}\left(d_j - y_j\right)^2 \tag{2.3.4}$$

Deep neural networks have the capability to reflect any complex nonlinear function, which has the ability of self-learning and adaptivity; during the training, it automatically extracts "reasonable" solutions by learning the set of examples with correct answers; it also adaptively memorizes the weight of networks; artificial neural networks do not have a significant impact on the training results even if its local or partial neurons are damaged.

However, deep neural networks have a myriad of disadvantages, such as slow convergence and local minima. Traditional DNNs are a local search optimization which solves a great deal of complicated nonlinear problems. The weights of the neural networks are adjusted by using the direction of local improvement, which will cause the algorithm to fall in the local extrema. The selection of a structure is too large, the efficiency in training is not high, overfitting may occur which cuts off fault tolerance. If the selection is too small, the network may not converge.

In 2006, Hinton set up the concept of deep learning, specifically the method of pretraining was used to alleviate the problem of local optimal solution, the hidden layer was increased to many layers, which deepen the depth in neural networks. Moreover, multilayer neural networks have better learning ability and its training complexity is effectively alleviated by using layer-by-layer initialization (Hinton & Salakhutdinov, 2006).

Unlike traditional approaches, deep learning (DL) methods are inspired by CNNs (ConvNets) which lead to the new era of modern computing. Although deep learning methods import video frames as the input data directly, the images will be input into the network without feature extraction. Thus, deep learning is regarded as a kind of novel methods, the feature maps are generated correspondingly. The feature maps are integrated into the algorithm without any interventions, the input of the end-to-end methods is an image, the output is classification probability.

CNNs as one of the most popular methods at present have been applied to directly tackle the original images. There are inherent local patterns in images such as mouth and eyes, etc. in a face; therefore, the CNNs are regarded as the combination of digital image processing and artificial neural networks. It plays a pivotal role in object detection, object recognition, and image segmentation (Ciresan, Giusti, Gambardella, & Schmidhuber, 2012; Donahue, Jia, Vinyals, Hoffman, Zhang, Tzeng, & Darrell, 2014; Farabet, Couprie, Najman, & LeCun, 2013; Girshick, Donahue, Darrell, & Malik., 2014; Krizhevsky, Sutskever, & Hinton, 2012; Karpathy, Toderici, Shetty, Leung, Sukthankar, & Li, 2014). CNNs connect both upper and lower layers together through using convolutional kernel, a "weight sharing" strategy can make fully use of a group of neurons with the same connection weights (LeCun & Bengio, 1995; LeCun, Bottou, Bengio, & Haffner, 1998).

In CNNs, pooling operation ensures image translation invariance, which makes the model not affected by changing the positions. Pooling operation makes the network have a larger receptive field, which accepts a larger input. The increase of receptive field will allow the network to learn more features.

The end-to-end model is convolutional neural networks (CNNs or ConvNets); compared to traditional methods, the convolution kernel in convolutional layer can be used as the feature map extractor. Although each layer needs to be designed by itself, there are not additional operations that are required to obtain the feature map. The end-to-end model eliminates the need for data annotation before each individual learning task is completed.

In this end-to-end model, a prediction result will be acquired in the end. Compared to the predicted result with ground truth, an error will be obtained. This error will be transmitted to each layer of the model by using backpropagation, the representation of each layer will be adjusted according to this error till the model converges or the expectation is reached. Compared to the deep learning and traditional machine learning methods, deep learning extract feature itself using its convolution kernels, thus, no extra feature extraction process is required, but the traditional machine learning methods normally requires several models to work together.

For traditional learning, the features extracted from the original data are very particular in the image, because the number of image pixels is too large, the dimension is very high, which leads to the curse of dimensionality problem. Thus, the difference between deep learning methods and the traditional methods is that it is able to directly input a two-dimensional image into the model so as to export the classification results at the side of output. The advantage is that it does not need complex preprocessing, feature extraction, and pattern classification, the parameters needed by the deep neuron network are obtained through continuous optimization, the required classification is given in the output layer. The performance of this architecture is effective than that of other traditional algorithms. Table 2.3.1 shows the summary of two different methods with their advantages and disadvantages for human behavior recognition.

Typical deep learning methods include convolutional neural networks (CNNs or ConvNets), deep belief networks (DBN), and autoencoder. A number of layers in a neural network directly determine its ability to depict reality by using fewer neurons per layer to fit more complex functions. With the deepened neural networks, optimization function is more likely to fall into the local optimal, which is deviated from global optimal. Therefore, the performance of deep neural networks, trained with the limited data, may not be good enough. Meanwhile, with the increased layers of a neural network, gradient vanishing turns to be more serious. In order to overcome the gradient vanishing problem, sigmoid activation function is substituted by using ReLU, maxout, and other activation functions. Thus, deep learning is not only subject to the number of layers, but also dependent on the number of training data.

Most of the state-of-the-art deep learning methods were based on “detection by classification” framework, which finds the proposals and conducts the classification for each proposal. However, the bounaries of the proposals have been set during the clssfication. A single shot action detector (SSAD) was figured out to detect action instances in untrimmed videos, which adopts the temporal convolutional layers to skip the proposal generation step (Lin, Zhao, & Shou, 2017). Saha *et al*. came up with a spatio-temporal detection and classification networks for mulitple concurrent actions in the

untrimmed videos. It utilizes two pairs of region proposal networks (RPN) to localise and score actions from the original images for further classificaitons (Saha, Singh, Sapienza, Torr, & Cuzzolin, 2016). A single shot multi-box detector (SSD) was adopted to achieve regression and classification of the detected boxes in each frame, a novel greedy algorithm was designed, which can generate multiple action tubes incrementally (Singh, Saha, Sapienza,, Torr, & Cuzzolin, 2017).

Table 2.3.1 The summary of two different methods for human behavior recognition

	Advantages	**Disadvantages**
Traditional methods	Easy to be implemented Redundant features can be reduced before the training. Feature engineering (easy to explain and understand) It does not require GPU acceleration.	It cannot achieve the real time recognition usually. It requires several models to work together. The results may be affected by external environments, less accurate.
Deep learning methods	It can achieve real-time recognition. It can train the data without additional feature extraction. The results may not be affected by external environments, more accurate. It is adaptable and easy to be converted (transfer learning).	It requires a large amount of data. It requires GPU acceleration. It requires more time for training. Black box (hard to be explained and understood)

Szegedy *et al*. treated object detection by using deep neural networks as a regression problem (Szegedy, Toshev, & Erhan., 2013). Girshick et al. put up a region-based convolutional neural network (R-CNN), which took use of the region proposal (Sande, Uijlings, Gevers, & Smeulders, 2011) to gain multiple local regions of image, CNNs are utilized to grab these features of each region (Girshick, Donahue, Darrell, & Malik., 2014). The region proposal is to find out the possible position of the target in the image, the

intersection-over-union (IOU) is guaranteed while selecting candidate windows by using visual information, such as texture, edge, and color in the image.

Moreover, the region proposal for R-CNN is implemented by using selective search algorithm (Sande, Uijlings, Gevers, & Smeulders, 2011). In order to achieve object detection, in general, sliding window is adopted to select all the possible region boxes on the image, then generate the feature maps of these boxes by using classification method to get all the selected regions, then suppress the output results by using non-maximum value.

However, the disadvantages of this method are apparent. The complexity is too high, resulting in a lot of redundant candidate regions. Moreover, it is impossible to take every scale into account, the target location is not so accurate and feasible. Thus, selective search was suggested which can effectively remove redundant candidate regions and greatly reduce the computations. Selective search combines both exhaustive search and image segmentation for object recognition, and hierarchy structure for recognition. Different from exhaustive search, selective search approach recommends whether the useless regions should be deleted and only the regions enclosing possible objects are left, which will have time efficiency.

However, R-CNNs still have the shortages, which require to extract the local regions that increase the usage of disk space. Traditional CNNs require fixed size of input images, but normalization may stretch the objects, the information may get lost; moreover, most of the region proposals may be overlapped with others which extract features from the overlapping portions.

Spatial pyramid pooling networks (SPP-Nets) were adopted to overcome the information loss and usage problems of R-CNNs by replacing it with the last pooling layer (He, Zhang, Ren, & Sun, 2015). A MultiBox was proposed which trained CNN instead of using the same strategy in R-CNN by applying selective search algorithm to predict the Region of Interest (ROI) (Erhan, Szegedy, Toshev, & Anguelov, 2014), it can perform the single object detection by replacing the confidence with the single class

prediction. The shared computation was applied to uplift the speed and accuracy, the features are selected with pooling operation having various filter sizes and then connected as the input into a fully connected network. The global and local visual features were based on SPP-Net structure, SPP-Net is faster than the R-CNN.

Fast R-CNN (Girshick, 2015) was propounded to improve R-CNN which achieves higher mAP (average precision) based on PASCAL VOC2012 (Everingham, Van Gool, Williams, Winn, & Zisserman, 2010), it combines convolutional neural network, classifier, and bounding box regression into a simple network. Fast R-CNN has the ability to update the training process of all network layers through using pooling layers and backpropagation. Moreover, Fast R-CNN maps region proposal directly on feature map, the image only needs to be fetched once, which greatly cuts off time consuming, softmax with a fully connected layer is used to replace SVM and output the final class. A linear regression layer parallel to the softmax layer was also added to output the bounding box coordinates. Thus, all the required output comes from a single network which diminishes the training time.

The Faster R-CNN was implemented in 2017, which achieves high accuracy and fast computational speed compared with previous models. Unlike the R-CNN and Fast R-CNN reply on selective search algorithm, which is time-consuming. Faster R-CNN replaces the region proposal network (RPN) with the selective search method so as to share the convolutional features of whole image and produce the feature maps (Ren, He, Girshick, & Sun, 2017).

R-CNN with its associated models belongs to a two stages method (Lin, Goyal, Girshick, He, & Dollár, 2017) which generates the region proposal, a CNN is utilized for further classification, a number of extended frameworks based on this method have been also proposed (He, Gkioxari, Dollár, & Girshick, 2017; He, Zhang, Ren, & Sun, 2016; Lin, Dollár, Girshick, He, Hariharan, & Belongie, 2017; Shrivastava, Gupta, & Girshick, 2016). Different from R-CNN, a one stage method was proposed which applies the

algorithm directly to the input image and outputs the class label and corresponding locations.

You Only Look Once (YOLO or DarkNets) was proposed in 2016, which treats visual object detection as a regression problem based on end-to-end neural networks. Compared with Fast R-CNN, YOLO has a large positioning error and a lower recall rate compared with the region proposal-based methods. However, YOLO has a higher accuracy for locating and recognizing background, while Fast R-CNN has a high false positive rate. The core idea of YOLO is to import a whole picture as the input of the neural network, directly return the position of the bounding box and the label of the class.

YOLO networks have the ability to train loss functions directly corresponding to the detection performance, the entire model is trained jointly. YOLO algorithm is : (1) a fixed 7×7 grid applied to the input image; if the center of a sample falls on the corresponding grid, then the grid will correspond to the position of this object; (2) each grid prediction encompasses object position and its confidential information which is formed as a vector; (3) the network output layer corresponds to each grid, which is trained as the end-to-end model for classification. The loss function of YOLO network contains multiple parts, which is shown in Eq. (2.3.5),

$$
\begin{aligned}
L = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y - \hat{y}_i)^2] \\
+ \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right] \\
+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\
+ \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 \\
+ \sum_{i=0}^{S^2} 1_i^{obj} \sum_{C \in classes} \left(p_i(c) - \hat{p}_i(c)\right)^2
\end{aligned}
\tag{2.3.5}
$$

where 1_{ij}^{obj} denotes the $j-$th bounding box which corresponds to object prediction in the i-th cell, 1_{i}^{obj} indicates that the object is appeared in the $i-$th cell, λ_{coord} is a constant for the cell with objects, λ_{noobj} is a constant for the cell without objects. S^2 is the cell grid. If the grid cell contains an object and the center of an object is also in that grid cell, the grid cell will respond for detecting this object. B denotes the bounding box that each grid cell will be predicted. Each bounding box will be assigned a confidence score to output the probability which is predicted by the network. x, y refer to the coordinates related to the center of the box to the grid cell, w, h mean the predict width and height which are related to the image. The confidence c is the IOU between the predicted bounding box and the ground truth box.

However, the loss function was implemented into YOLO network by utilizing the sum-squared error, which makes convergence become worse; meanwhile, it cannot align with the maximized average precision correctly. YOLO network contains 24 convolutional layers and two fully connected layers which is inspired by GoogLeNet that is more efficiency for classification (Szegedy et al., 2015; Redmon, Divvala, Girshick, & Farhadi, 2016).

YOLO9000 (YOLOv2) was presented to overcome the limitations of the initial version of YOLO, which detects up to 9,000 object classes in real time and improve the accuracy and recall of the object location. YOLOv2 applies WordTree to conduct joint training for classification, the joint training algorithm was trained for object recognition and detection (Redmon & Farhadi, 2017).

YOLOv2 applies anchor box from Faster R-CNN to predict the coordinates of each bounding box. COCO (Common Objects in Context) dataset supports YOLOv2 for object detection (Lin, Maire, Belongie, Bourdev, Girshick, & Hays, 2014) and ImageNet dataset for object classification (Miller et al., 1990; Russakovsky et al., 2015), compared with YOLO, YOLOV2 has been significantly amended in recognition class, accuracy, speed, and positioning accuracy.

Batch normalization (Ioffe, & Szegedy, 2015) was adopted to all convolutional layers, which normalizes the data of all layers to prevent gradient vanishing and gradient exploding and increases 2% mAP in YOLOv2. Since there are only convolutional and pooling layers in YOLOv2, the input of YOLOv2 is limited to those fixed size images. In order to enhance the robustness of this model, YOLOv2 takes advantage of a multiscale training strategy, specifically changes the input image size after several rounds of iterations during training process. Therefore, by utilizing multiscale training strategy, YOLOv2 is adaptive to the images with different sizes and predicts very satisfactory results. During our tests, YOLOv2 has been fed with various sizes of images as its input.

YOLOv3 was proposed to amend YOLOv2, which takes use of logistic regression to locate an object based on a bounding box; the logistic regression is applied independently to each class; the binary cross-entropy loss is taken to replace the softmax loss so as to increase class predictions. Moreover, YOLOv3 adopted the block structure in ResNet and predict it from three feature scales which is similar to feature pyramids network (Lin, Dollár, Girshick, He, Hariharan, & Belongie, 2017). In YOLOv3, *k*-means clustering is selected, each grid cell in the feature map is able to predict three bounding boxes, the prediction outcomes including position of each box, which is defined by the center coordinates (t_x, t_y), box width and height (b_w, b_h), object prediction, and class predictions. The parameters of bounding box prediction are shown as:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{2.3.6}$$

where b_w and b_h are the width and height of the bounding box, b_x and b_y indicate the position of the predict bounding box. The coordinates (t_x, t_y) and (t_w, t_h) are generated by

using the network. c_x and c_y are the lengths of the grid cell, p_w and p_h are the width and height of prior bounding box, $\sigma(\cdot)$ is used to constrain its possible offset.

YOLOv4 was given in 2020 which optimizes the algorithms such as spatial attention module (SAM), path aggregation network (PAN) and cross-iteration batch normalization (CBN) etc. (Woo, Park, Lee, & So Kweon, 2018; Liu, Qi, Qin, Shi, & Jia, 2018; Yao, Cao, Zheng, Huang, & Lin, 2020). The model was trained based on a single GPU and owns the advantage of time efficiency. YOLOv4 takes the place of the spatial-wise attention by using point-wise attention; moreover, the concatenation was substituted by using the original shortcut connection. Afterwards, a Bag of Freebies (BoF) was utilized to be associated with a Bag of Specials (BoS) so as to uplift the overall accuracy and performance, which encapsulates enlarging the receptive field by using attention mechanism, etc.. Moreover, various of new features such as weighted-residual-connections (WRC), cross mini-batch normalization (CmBN), self-adversarial-training (SAT), Mish activation function (Misra, 2019) and Mosaic data augmentation etc. were adopted, which shows the promising results on public dataset (Bochkovskiy, Wang, & Liao, 2020).

In recent years, deep learning has made positive progress. The main approaches are able to be grouped into two categories. (1) Two-stage approaches, such as R-CNN series. The chief idea is to use the selective search or CNN network such as RPN to produce a series of sparse region proposals, classify and regress these region proposals. The advantage of two-stage approach is highly accurate. (2) One-stage approaches, such as YOLO and SSD, the main objective is to carry out intensive sampling uniformly at different locations of the image. Multiple scales and aspect ratio have been applied to sampling, CNNs extract features directly for classification and regression, the whole process only needs one step. Thus, its advantage is obvious, but an important disadvantage of uniform dense sampling is that it is difficult to be trained. This is the reason why the positive samples and negative samples (background) are extremely unbalanced, which results in a slightly low accuracy.

The single shot multibox detector (SSD) was proposed in 2017 which is a one-stage approach (Singh, Saha, Sapienza, Torr, & Cuzzolin, 2017; He et al., 2017; Huang et al., 2017). It is much better than YOLO model in balancing the accuracy and computational speed in classification. For Faster R-CNN, it first obtains the candidate frame through CNN, and then performs classification and regression, and YOLO and SSD can complete the detection in one step. SSD is trained to handle the multiple object classes (Erhan, Szegedy, Toshev, & Angulelov, 2014), and compare with YOLO. SSD directly detects the objects, rather than after the fully connected layer like YOLO. SSD extracts feature maps with multiple scales for detection, the large-scale feature maps (lower feature maps) are applied to detect small objects, small scale feature maps (upper feature maps) are imported to detect large objects. SSD is use of prior boxes with multiple scales and aspect ratios which permits the predictions in multiple scales. Thus, SSD and YOLO share the same CNN network for detection, but it takes use of multiscale feature maps. The input image size of the model is 300×300. SSD selects VGG-16 as the basic convolutional network and adds several of convolution layers at the end of VGG-16 to obtain much feature maps for object detection.

Pertaining to a feature map with the size of $m \times n \times p$, the convolution kernel having the size 3×3 is applied to and generate a value at each pixel location of the given image, which is a score of a class or an offset from the default bounding boxes. The core idea of SSD is that both lower and upper feature maps will be selected for object detection. The feature maps of different sizes are employed in multiscale way. In general, the feature maps of CNN network are relatively large, the convolution with *stripe* = 2 or pooling will be used to shrink the size of feature maps. The advantage is that the larger feature map is employed to detect smaller target, while the smaller feature map is applied to detect larger target.

The matching strategy for SSD basically is based on Jaccard index, which matches each ground truth with a default bounding box having the largest overlapping, to ensure that each ground truth has a corresponding default box; and match each default box with any ground truth. As long as the value of Jaccard index is greater than a threshold, a

ground truth box may correspond to multiple default boxes. The calculation of Jaccard index is given as eq.(2.3.7)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|} \in [0,1]. \tag{2.3.7}$$

The loss function of SSD is defined as the weighted sum of the location loss (*los*) and the confidence loss (*conf*),

$$L(x,c,l,g) = \frac{1}{N}\Big(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\Big) \tag{2.3.8}$$

where N denotes the number of positive samples of the default boxes which are matched to the ground truth boxes. $x_{ij}^{p} \epsilon \{1,0\}$ is the indicator which means that the i^{th} bounding box matches to the j^{th} ground truth box of category p. The confidence denotes as c, l is the predicted box, g refers to the ground truth box, and α stands for the weight factor which is set to 1.0 for the cross validation. $L_{conf}(x,c)$ is the confidence loss which is the softmax loss over the class confidence c, given as eq.(2.3.9),

$$L_{conf}(x,c) = -\sum_{i \epsilon pos}^{N} x_{ij}^{p} \log\big(\hat{c}_i^{p}\big) - \sum_{i \epsilon neg} log\big(\hat{c}_j^{p}\big) \tag{2.3.9}$$

$$\text{where} \quad \hat{c}_i^{p} = \frac{\exp(c_i^{p})}{\sum_p \exp(c_i^{p})}$$

$L_{loc}(x,l,g)$ refers to the localization loss, the equation as followed:

$$L_{loc}(x,l,g) = \sum_{i \epsilon pos}^{N} \sum_{m \epsilon \{cx,cy,w,h\}} x_{ij}^{k} \text{smooth}_{L1}\big(l_i^m - \hat{g}_j^m\big)$$

$$\hat{g}_j^{cx} = \frac{\big(g_j^{cx} - d_i^{cx}\big)}{d_i^{w}} \qquad \hat{g}_j^{cy} = \frac{\big(g_j^{cy} - d_i^{cy}\big)}{d_i^{h}} \tag{2.3.10}$$

$$\hat{g}_j^{w} = \log\left(\frac{g_j^{w}}{d_i^{w}}\right) \qquad \hat{g}_j^{h} = \log\left(\frac{g_j^{h}}{d_i^{h}}\right)$$

where the localization loss is a smooth loss L_1 (Girshick, 2015) between the predicted box l and the ground truth box g, which are the central coordinate position (cx, cy) of the default bounding box (d), weight (w) and height (h). Moreover, the smooth L_1 loss is defined as

$$\text{smooth}_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & \text{otherwise.} \end{cases} \tag{2.3.11}$$

For one-stage approaches, SSD shows that it is faster than YOLO in real-time detection, and the precision of SSD is higher than the Faster R-CNN based on PASCAL VOC and COCO datasets. However, the basic size and shape of the prior box in the SSD network cannot be gained directly through learning. The recall rate of SSD network for small objects does not reach the performance of Faster R-CNN, because the low-level feature convolution layer is few, which leads to the issue of insufficient feature extraction.

There are improvement and enhancement for SSD network, such as the Deconvolutional SSD (DSSD) to add the deconvolutional layers and increase the mAP based on PASCAL VOC and COCO datasets (Fu, Liu, Ranga, Tyagi, & Berg, 2017). The Rainbow SSD (R-SSD) applies the rainbow concatenation method to merge the features from multiple layers which solves the problem not only for repeating the detected boxes existing in SSD algorithm, but also for small object detection (Jeong, Park, & Kwak, 2017). An attention mechanism was proposed to identify the regions and achieve a good result (He, Huang, He, Zhu, Qiao, & Li, 2017). A PyramidBox was proposed to handle with face detection problem (Tang, Du, He, & Liu, 2018). A one-stage framework that combines the advantages of fine tuning and pretraining was proposed which effectively transfers semantic information of low and high levels to improve the accuracy (Wang, Anwer, Cholakkal, Khan, Pang, & Shao, 2019).

In recent years, with the increased capacity of computing devices, deep neural networks (DNNs) gained a major attention to attain a variety of computer vision tasks (Lu, Shen, Yan, & Bacic, 2017; Gu, Yang, Yan, Li, & Klette, 2017; Zhang, & Yan, 2018; Shen, & Yan, 2018; Al-Sarayreh, Reis, Yan, & Klette, 2018; Liu, Nguyen, & Yan, 2019;

Ji, Yan, & Klette, 2019; Song, He, Yan, & Nand, 2019). DNNs own multiple layers with the pretrained method to alleviate the problem of local optimal solution, the difference between ANNs and DNNs is the hidden layer, DNNs expand hidden layers to many layers, namely, deep learning (Hinton, Osindero, & Teh, 2006).

Moreover, there are still other networks which have the possitive results in classification. Lin et al. proffered a Network in Network (NIN) which extracts more complex features by adding the micro multi-layer perceptrons into the filters of the convolutinal layers (Lin, Chen, & Yan, 2014). Ladder Networks (Pezeshki, Fan, Brakel, Courville, & Bengio, 2016; Rasmus, Berglund, Honkala, Valpola, & Raiko, 2015) brought lateral connections in the autoencoder, resulting in a positive accuracy on semi-supervised learning. Wang et al. put forward a Deeply-Fused Nets (DFNs) which imporves information flow by mixing intermediate layers in the base networks (Wang, Wei, Zhang, & Zeng, 2016). By enhancing the network with paths, the loss is minimized so as to improve image classification (Zhang, Lee, & Lee, 2016).

Deep learning methods rely on ANNs, such as convolutional neural networks (CNNs), R-CNNs, Fast R-CNNs, Faster R-CNNs, single shot multi-box detector (SSD), and You Only Look Once (YOLO). Moreover, the previous deep learning methods were well studied in different domains. The region proposal was proposed in 2017 which shows high precision in object detection (Ren, He, Girshick, & Sun, 2017). Faster R-CNN based on selective search leads to exceptional achievement in a series of competitions (Ren, He, Girshick, & Sun, 2017; He, Zhang, Ren, & Sun, 2016). Moreover, deep learning approaches gain the end-to-end process along with higher classification confidences which also show an invariant to illumination, pose, etc. (LeCun, Huang, & Bottou, 2004). In 2015, Ng et al. proposed a deep neuron network is pretrained by using AlexNet or GoogleLeNet based on imageNet to extract frame-level features, and import frame-level features and optical flow features into LSTM for training so as to attain the classification up to 88.6% accuracy based on UCF-101 dataset (Ng, Hausknecht, Vijayanarasimhan, Vinyals, Monga, & Toderici, 2015).

Nowadays, most of deep learning methods require either deepening the depth or widening the width of deep networks so as to improve the accuracy (He, & Sun, 2015), which have shown good performance compared with the traditional machine learning methods. However, as the number of hyperparameters rising such as channels, filter size, etc., the computational costs of deep learning network will be also ramped up. In most of these cases, deepening the deep learning model may not be able to solve the problem completely, which will encounter the overfitting or global minimization problem.

VGG network (Simonyan, & Zisserman, 2014) has been tested to find out how deep the deep neuron networks can be improved so as to continuously ramp up the classification accuracy. The previous work shows that more complex deep neural network with more parameters should have stronger representation ability. According to this basic rule, deep neuron networks have been developed from the seven layers AlexNet (Krizhevsky, Sutskever, & Hinton, 2012) to 16/19 layers VGG network (Simonyan, & Zisserman, 2014), and to the later 22 layers GoogLeNet (Szegedy et al., 2015). However, by increasing the number of convolution layers after the depth of CNN reaches a degree of depth, it not only could bring further improvement in classification, but also might lead to slow down the training, furthermore, perhaps bring in a worse classification accuracy based on the same test set. Moreover, after eliminating the problems of model overfitting provided giving small datasets, we find that the classification accuracy may still drop off while we deepen extra depth of the deep neural networks comparing with those shallow neuron networks.

After raised the number of layers in VGG network, it starts eliminating the classification performance. Thus, ResNet was inspired by using highway network (Sirvastava, Greff, & Schmidhuber, 2015) and the gate unit in LSTM (Hochreiter, & Schmidhuber, 1997). ResNet takes use of the multiple layers to learn the residual representation between the input and the output, instead of parameterized layers to directly learn the mapping between input and output (He, Zhan, Ren, & Sun, 2016a). The core idea of ResNet is to bring in the concept “identity shortcut connection”, which skips one or more layers directly, the residual learning unit establishes a direct correlation

channel between the input and the output throughout the identity mapping, so that the parameterized layer can concentrate based on learning the residual between the input and the output.

It is easier to learn residuals directly by using generally parameterized layers than to learn the mapping between inputs and outputs in terms of convergence rate, higher classification accuracy is able to be achieved by using more layers.

The output of residual unit is obtained by adding the concatenated output and input elements of multiple convolutional layers so as to ensure that the output and input elements of the convolutional layer are the same. By cascading these structures, a residual network is obtained. Moreover, ResNet has the following characteristics: (1) The network is thinner, which controls the number of parameters; (2) the number of feature map is progressively layer-by-layer to ensure the output feature expression ability; (3) The pooling layer uses a large number of downsampling operations to improve propagation efficiency.

In the neural network for calculating gradient optimization based on backpropagation method, since chain rule has been employed in the backpropagation to seek the gradient of hidden layer, the gradients of the shallow hidden layers will turn up severe attenuation, which is the source of gradient vanishing. This problem is particularly serious for sigmoid activation function, so later, deep neural networks have to use ReLU activation function to alleviate this problem. However, even ReLU activation function cannot avoid gradient vanishing problem brought by hundreds or thousands of times of continuous multiplications under the condition of extreme depth due to the scalar relationship between unit outputs of the network. By broadening the width of ResNet, it can outperform a deep ResNet with 1,001 layers and have the positive results based on CIFAR benchmarks (Zagoruyko, & Komodakis, 2016). After a long-standing investigation, a deep residual network was set by adopting identity mapping (He, Zhan, Ren, & Sun, 2016b) to solve the gradient vanishing problem. Because of batch normalization, ReLU and other methods have limited the mitigations for this level of gradient vanishing.

Recent work shows that if a convolutional network has shorter connections between the layers from the input to the output, then the network can be trained properly, the result will be much accurate and efficient. Huang et al. proposed a dense convolutional network (DenseNet) connecting each layer in a feedforward manner (Huang, Liu, Van Der Maaten, & Weinberger, 2017). In DenseNet, there are direct connections between two layers, the number of direct connections in this L-layer network is $\frac{L(L+1)}{2}$. Each layer utilizes the feature maps of all the previous layers as the input, its own feature maps are treated as the input of all subsequent layers. In order to ensure the maximum information flow between network layers, DenseNet directly connects all layers together.

Moreover, in order to maintain the feedforward characteristic, each layer gets extra input from the proceed layer and passes its feature map to the next layer. Compared with ResNet, DenseNet does not combine features by summing up them before they pass to the layer, but mingle features by connecting them. Therefore, the l^{th} layer (excluding the input layer) will have l inputs. These inputs are the feature information extracted by using all proceed layers.

As we know, for the convolution networks, the output of l^{th} layer will be the input of the $(l+1)^{th}$ layer (Krizhevsky, Sutskever, & Hinton, 2012), the transition is presented as

$$\mathrm{x}_{\ell} = \mathrm{H}_{\ell}(\mathrm{x}_{\ell-1}) \tag{2.3.12}$$

where x_0 is defined as the input of the entire network structure. The network consists of L-layers. $H_{\ell}(\cdot)$ denotes the operation of each layer which is represented by a nonlinear transformation, where l represents the l^{th} layer. $H_{\ell}(\cdot)$ is treated as a compound operation defined by batch normalization, ReLU activation function, pooling, and convolution operations. Simultaneously, the output of the l-th layer is defined as x_{ℓ}. ResNet (He, Zhang, Ren, & Sun, 2016) propounded the identity function, a skip-connection bypasses the nonlinear transformation, the layer transition is shown as

$$\mathrm{x}_{\ell} = \mathrm{H}_{\ell}(\mathrm{x}_{\ell-1}) + \mathrm{x}_{\ell-1} \tag{2.3.13}$$

The advantage of ResNet is that the gradient is directly transmitted from a layer to its previous layer through the identity function. However, the identity function and the output of $H_\ell(\cdot)$ are combined by summation, which hinders the information flow in the network.

Therefore, in order to further improve the flow of information between layers, Huang et al. proffered direct connections from any proceed layer to all next layers (Huang, Liu, Van Der Maaten, & Weinberger, 2017), the layer transition is presented as

$$x_\ell = H_\ell[x_0, x_1 \dots x_{\ell-1}] \tag{2.3.14}$$

where $[x_0, x_1 \dots x_{\ell-1}]$ denotes as the feature maps concatenated and produced in layers 0, 1, …, $\ell - 1$.

Because there is no need to retrain the redundant feature maps, this dense connection model requires fewer parameters than convolutional networks. The feedforward architecture is viewed as an algorithm with the state that is passed from one layer to another. Each layer reads the state from its proceed layer and writes it to the next layers. It changes state, but passes information that needs to be retrained. The dense network architecture clearly distinguishes between information added to the network and information retrained. In addition to have fewer parameters, another advantage of DenseNet is that it improves the information flow and gradients of the entire network, which makes the network easy to be trained. Each layer has direct access to the gradients from the loss function and the original input signal. (Lee, Xie, Gallagher, Zhang, & Tu, 2015).

DenseNet allows a layer to access feature maps from all of the proceed layers. It naturally scales to hundreds of layers without any optimization difficulties. All layers hold the weights within the same block, the features extracted from very early layers are supplied directly to very deep layers throughout the same dense block. Eventually, DenseNet produces a consistency of accuracy without any degradation or overfitting as the number of layers and associated parameters are raised.

Moreover, in order to dissolve the problem of overfitting and trim down the parameters of deep neuron networks. Xception (Chollet, 2017) and ResNeXt (Xie et al. 2017) were proposed to increase the cardinality. They show empirically that cardinality not only saves the total number of parameters, but also is expressive than the depth and width. The attention has been studied which implies not only where to focus, but also how to improve the representation of interests (Mnih, Heess, & Graves, 2014; Xu et al., 2015; Jaderberg, Simonyan, & Zisserman, 2015).

Xie et al. suggested a highly modularized deep learning network (ResNeXt) for image classification. It raises up accuracy without ramping up the complexity of the deep learning method, meanwhile it effectively cuts off the number of hyperparameters (Xie et al. 2017). The ResNeXt was motivated from the idea of VGG stacking blocks of the same shape and the split-transform-merge idea of the Inception model, which holds robust scalability and is able to meliorate the accuracy without substantially altering the complexity of the model. ResNet model is stacked by using the basic residual modules.

ResNeXt is similar to Inception module (Szegedy et al., 2015), they both follow the "split-transform-merge" paradigm. However, in ResNeXt, the outputs of different paths are combined by addition operations, in Inception module, they are deeply concatenated. Another difference is that each path in Inception module has bias from each other, while in the ResNeXt module, all paths follow the same topology. It replaces the three-layer convolution block of original ResNet model by using a parallel stack of blocks with the same topology, which uplifts the accuracy without significantly raising the number of the parameters.

Simultaneously, the hyperparameters are reduced because of the topological structure of ResNeXt model. Moreover, cardinality in ResNeXt is the size of the set of transformation and an essential factor to the dimensions of depth and width. By adopting ResNeXt model, the training error is much lower compared with the ResNet parameters. Moreover, by extending the cardinality, the model is much efficient compared with only extending the network depth or width on the ResNet model, thus dips 1.6% error rate.

On the one hand, Convolutional Neural Networks (CNN) comprise of a series of convolution layers and pooling layers. We apprehend the characteristics of the input image from the global receptive field so as to precisely savvy the image. On the other hand, the current convolutional network carries out convolutional operations in 2D space, which is regarded as the networks that model spatial and channel-wise information within relevant receptive field. A good deal of work embarks on convolutional networks for digital image processing from the spatial domain. For instance, embedding multiscale information in Inception mode (Szegedy et al., 2015) was offered to aggregate visual features of verity of sizes of receptive fields so as to snatch a better performance; attention mechanism was brought in the spatial domain which gains positive results.

A simple and effective attention module for feedforward convolutional neural networks was set forth in 2018 which is called Convolutional Block Attention Module (CBAM). In order to make the performance of CNN model better, the focus of recent work is on three important aspects: Depth, width and cardinality. ResNet makes it possible to establish very deep networks, while GoogLeNet shows that the width is another important factor in improving model performance. In addition, Xception and ResNeXt increase the cardinality of the network. Our experience shows that cardinality not only saves the total number of parameters, but also produces a stronger representation ability than depth and width. Moreover, the module is able to integrate into any CNN architectures seamlessly with negligible overheads and is trainable in the end-to-end way along with the base CNNs.

The goal of attention mechanism is based on essential features and suppress unnecessary features (Woo et al. 2018). The structure of channel attention module utilizes both max pooling and average pooling with a shared network composed of multilayer perceptron with one hidden layer. The input feature map passes through global max pooling and global average pooling based on its width and height respectively, and goes through the shared MLP network. The output features are based on elementwise, and activated by using sigmoid function to generate the final channel attention feature map. The channel attention is computed as follows

$$M_c(F) = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right)$$

$$= \sigma\left(\mathrm{W}_1\left(\mathrm{W}_0\left(\mathrm{F}_{avg}^{c}\right)\right) + \mathrm{W}_1\left(\mathrm{W}_0\left(\mathrm{F}_{max}^{c}\right)\right)\right) \quad (2.3.15)$$

where $\mathrm{F}_{\mathrm{avg}}^{\mathrm{c}}$ and $\mathrm{F}_{\mathrm{max}}^{\mathrm{c}}$ denotes average pooling and max pooling in the channel attention module, W_1 and W_0 are the weights which share both input and ReLU activation function by using the W_0.

The SKNet was proposed in 2019, SK convolution kernels with different kernels are presented to implement receptive field of neurons, which encloses three operators: Split, fuse, select. In the standard convolution network, the size of the receptive field of neurons in each layer is the same. In neuroscience, the size of receptive field is constructed by using a stimulation mechanism. The method makes CNNs adjusting the size of their receptive field adaptively and competently for the input information. There are many branches, the size of convolution kernel of each branch is different. Finally, convolution kernels with various sizes are fused by using softmax function. The attention models produce a diversity of receptive fields, the multiple SK units are stacked into the SKNet.

The split operator generates multiple paths with various kernel sizes which correspond to different RF (receptive fields) sizes of neurons. The fuse operator combines and aggregates the information from multiple paths to obtain a global and comprehensive representation for selection weights. The select operator aggregates the feature maps of differently sized kernels according to the selection weights.

CNN is good at capturing the existence of features, because the convolution structure was designed for this purpose. However, CNN is unable to explore the relationship between each feature attributes, such as relative positions, size, and direction of the feature etc. Thus, Sabour et al. firstly proposed a new deep learning network that is much effective for image processing, which called capsule network (CapsNet) (Sabour, Frosst, & Hinton, 2017). It combines the advantages of CNN structure and takes into account the relative position, angle and other information that are missing from the CNN, thereby

improving the recognition effect. The CapsNet structure contains two main components, which are primary capsule and digital capsule.

Capsule is a group of neurons whose input and output vectors represent the parameters of a specific type like the probability of occurrence of objects, conceptual entities, etc. It took use of the length of input and output vectors to represent the probability of the existence of the entity, the direction of a vector represents the instantiation parameters. The capsules at the same level predict the instantiation parameters of higher-level capsules through the transformation matrix. The higher-level capsules become active, if the multiple predictions are consistent. The activity of the neurons in an active capsule represents the various attributes of the specific entities that are appearing in the image, which contains the parameters, such as posture (position, size, direction), deformation, speed, reflectivity, color, texture, etc. The lengths of the output vectors represent the probability of an entity, its range is between $[0, 1]$. A nonlinear function called "squashing" was proposed to ensure that the length of the short vector is reduced to almost zero, while the length of the long vector is compressed. The following equation is the expression of the non-linear function

$$\mathrm{V}_j = \frac{\|\mathrm{S}_j\|^2}{1 + \|\mathrm{S}_j\|^2} \frac{\mathrm{S}_j}{\|\mathrm{S}_j\|} \tag{2.3.16}$$

where V_j represents the vector output of capsule j, S_j denotes the total input. Moreover, the total input S_j is a weighted sum over all predicted vectors from the capsules, which acquired by multiplying the vector output u_i from capsule in the layer below by the weight matrix W_{ij}.

$$\mathrm{S}_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{2.3.17}$$

where c_{ij} represents coupling coefficients, which are updated and determined iteratively by the dynamic routing process.

For traditional neural networks, it is basically consisting three different layers, which includes input layer, hidden layer, and output layer. The weight is connected between relevant layers, the nodes between these layers are not connected. Moreover, the final output is controlled by using activation function, which is predetermined. From the artificial neural networks to the convolution neural networks, only one input is processed, the proceed input and the next input are completely irrelevant.

In order to resolve this problem, recurrent neural networks for processing the sequence data (RNN) was investigated in various research areas (Sutskever, Martens, & Hinton, 2011; Sutskever, Vinyals, & Le, 2014; Graves, & Jaitly, 2014).

Hopfield network is a neural network with a single layer fully connected to each other. Each neuron is an input as well as an output. Each neuron in the network transmits its output to all other neurons through the connection weight, at the same time, receives the information transmitted by using all other neurons, Hopfield network is applied to solve combinatorial optimization problems, which is the prototype of the earliest RNN (Hopfield, 1982).

The basic neural network only establishes the weight connection between layers. For RNN, the current output of a sequence is related to the previous output. The specific expression is that the network will remember the previous information and apply it to the calculation of the current output. Thus, the nodes between the hidden layers are no longer disconnected but connected, the input of the hidden layer includes not only the output of the input layer but also the output of the previous hidden layers.

The parameters in recurrent neural network are shared at different times. In order to convert the current state to the final output, the recurrent neural network needs another fully connected neural network to complete this process. The parameters in the fully connected neural network for output are consistent. The output of the recurrent neural network is defined as

$$O_t = \sigma(w_i I_t + u_i O_{t-1}) \tag{2.3.18}$$

where $\sigma(\cdot)$ represents the nonlinear functions, w_i and u_i are network parameters, which are used to control the relative importance of current and past information. I denotes as the input, O refers to the output. Moreover, the pseudo codes of RNN algorithm are simply shown as:

Algorithm 2.3.1 The output of RNN

Input: The input O in time t

Output: The output I in time t

For t **from** 1 **to** T **do**

$a_t \leftarrow w_{it} I_t$

$b_t \leftarrow u_{it} O_{t-1}$

$O_t \leftarrow \sigma(a_t + b_t)$

End for

The input of each recurrent unit is composed of input I_t at time *t* and the input O_{t-1} at time *t*-1. The new output representation is calculated by transferring layers in the recurrent neural network. RNN includes one-to-one, one-to-many, many-to-one, and many-to-many. Moreover, it turns up in the tasks such as music generation, sentiment classification, name entity recognition, machine translation etc. The activation functions of RNN are sigmoid function, hyperbolic tanh function, and ReLU function. The loss function of RNN is a sum of all losses,

$$L(\hat{O}, O) = \sum_{t=1}^{T_O} L(\hat{O}^t, O^t) . \tag{2.3.19}$$

The Back-Propagation Through Time (BPTT) algorithm (Werbos, 1990) is to train the RNNs. In fact, the essence is the BP algorithm, but RNN tackles time series, so it is temporal-based backpropagation. The idea of BPTT is as same as that of BP algorithm, which keeps searching for better points along the negative gradient direction of parameters to be optimized till convergence. The disadvantage of RNNs is that during

training, it may face the gradient vanishing problem. When the time span is large, the gradient of backpropagation will be getting small when we adopt sigmoid or hyperbolic tanh as the activation function, result in gradient vanishing.

With regard to recurrent neural networks, the state is transmitted from front to back in one direction. However, the output of current time is related to the proceed state and the next state. Therefore, the extension of RNN is proposed to solve problem (Schuster, & Paliwal, 1997). The bidirectional recurrent neural network is composed of two unidirectional recurrent neural networks, whose output is determined by using states of the two networks. Moreover, at a given time, the input is related to two opposite recurrent neural networks, the output is determined by using two unidirectional recurrent neural networks. However, bidirectional RNN needs whole sequence to appear before processing, but it can't proceed before receiving the sequence.

Moreover, in order to enhance the expressive power of bidirectional RNN model, the loop at each hidden layer is allowed to be executed multiple times. Deep recurrent neural network replicates the structure of recurrent state multiple times at each hidden layer. The parameters in the loop of each hidden layer are the same. A multi-dimensional RNN (MDRNN) replaces a single recurrent connection with multiple recurrent connections to unfold the RNN so as to handle the multi-dimensional data, hence, RNN is applied to digital image processing and digital video processing, and avoid scaling problems (Graves, Fernández, & Schmidhuber, 2007). Moreover, an attention mechanism in RNN is proposed to achieve the image classification (Mnih, Heess, & Graves, 2014). Bahdanau et al. adopted the attention mechanism to perform the translation and alignment simultaneously in machine translation (Bahdanau, Cho, & Bengio, 2015).

Two attentional mechanisms are proposed which enclose the global mechanism and local mechanism for machine translation, our experimental results show that local attention is better than global attention (Luong, Pham, & Manning, 2015). Based on previous studies, Yin et al. firstly implemented the attention mechanism in CNNs (Yin,

Schütze, Xiang, & Zhou, 2016). An end-to-end recurrent attention model (RAM) is achieved the pedestrian attribute recognition (Zhao, Sang, Ding, Han, Di, & Yan, 2019).

If the parameters of model are too many and the training samples are too few, the trained model is easy to produce the phenomenon of over fitting. Thus, Dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012; Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) effectively alleviated the occurrence of overfitting and achieved the effect of regularization in recurrent neural networks. While dropout as a kind of tricks has been applied for training deep neural network. In each training batch, by omitting half of the feature detectors (let half of the hidden layer node value be zero), the overfitting is significantly reduced. In RNN, dropout exists only between hidden layers of the recurrent structure, not between the same hidden layer of the recurrent structure, from the time $t-1$ to time t, recurrent neural network will not be involved in dropout, but in the same time t.

As a powerful multilayer neural network model, RNN has the long-term dependencies of the model on time. Because of gradient explosion and gradient vanishing, this limitation leads to the unstable variation of the errors in the process of model training. Thus, in order to utterly solve the problem of gradient vanishing, Hochreiter and Schmidhuber were firstly proposed long short-term memory (LSTM) (Hochreiter, & Schmidhuber, 1997). LSTM embraces input gate, forget gate, and output gate. Gers *et al.* remedied the LSTM with forget gate, the network can purge unnecessary information and establish peephole connections (Gers, Schmidhuber, & Cummins, 2000; Gers, & Schmidhuber, 2001). The traditional LSTM replaces a hidden layer in RNN with a more complex structure, which called memory block.

In LSTM, gate as the fully connected layer, its input is a vector, its output is a real vector between $[0, 1]$, which is controlled by using sigmoid function. Cell represents the current state of the memory block, which corresponds to the neurons of the hidden layer in the original RNN. For the input gate, it determines the input value x of the current time t which passes through to cell state C. The forget gate determines the cell state C

of the time $t-1$ which is reserved to the current time t. The output gate is controlled by using the cell state C of the current time t which exports the current output value h of LSTM at current time t. Generally, for input gate, output gate, and forget gate, the sigmoid function is adopted as the activation function. For input and cell, hyperbolic tanh function is offered as the activation function.

At time t, there are three inputs of the LSTM, including the input value I of the current time I_t, the output value h of LSTM at the previous time h_{t-1}, the cell state at the previous time C_{t-1}. Two outputs of the LSTM are acquired, including the output value h of LSTM at the current time h_t, the cell state at the current time C_t. The definition of each parameter was given as:

$$\begin{aligned} f_t &= \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \\ i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma(W_o x + U_o h_{t-1} + b_o) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \sigma(W_c x_t + U_C h_{t-1} + b_c) \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{2.3.20}$$

where W, U, b represents the weight and bias, respective, which control the weight related to the input gate, $\sigma(\cdot)$ denotes the sigmoid function, i_t, f_t, o_t stands for the input gate, forget gate, and output gate, h_t means the output; c_t represents the cell state at the time t; $\odot$ is the element-wise product.

LSTM uses "Truncated BPTT" for training process. The gradient of other parts will be truncated, which will not be backpropagated to the memory block of the previous moment. In 2005, a Full Gradient BPTT method is proposed to train the LSTM, which shows that the bidirectional networks are more effective than others, the result was more accurate than RNN (Graves, & Schmidhuber, 2005). Greff et al. analyzed the eight kinds of LSTM deformations, the results show that none of the eight deformations on the

original LSTM can significantly improve the performance of LSTM, it is proved that the forget gate and the output activation function are the most important components (Greff, Srivastava, Koutnik, Steunebrink, & Schmidhuber, 2016).

The original LSTM has not the forget gate and peepholes (Gers, Schmidhuber, & Cummins, 2000). Gers et al. set forth the peepholes for precise time control, which extended the peephole connections in cells to other gates, which makes it easier for LSTM to distinguish time series (Gers, & Schmidhuber, 2000). Gers et al. put forward full gradient LSTM, the method makes the gradient of LSTM available, stable, and reliable (Graves, & Schmidhuber, 2005).

In 2014, an RNN encoder-decoder model is proposed by adopting the gated recurrent unit (GRU) to achieve different tasks (Cho, Merrienboer, Gulcehre, Bougares, Schwenk, & Bengio, 2014). The model contains two RNNs, one encodes the sequence into a vector representation of a fixed length, the other decodes the encoded vector representation into another sequence. Moreover, the length of the two sequences is different.

GRU chiefly made two changes based on LSTM: (1) GRU has two different gates which combines forget gate and input gate into a single gate called update gate, in order to control the amount of data that is retained in the previous memory at the current time; another gate is called reset gate, which controls how much past information to forget. (2) GRU omits the memory cell that performs linear renewal, but takes use of the gating directly in the hidden cell to perform linear renewal.

The input and output of GRU structure are as same as those of traditional RNN, where the current input of x_t and the hidden state h_{t-1} are passed through the previous node, the relevant information of the previous node will be sent to the GRU. By combining both x_t and h_{t-1}, GRU will get the output y_t of the current hidden node and the hidden state h_t which will be passed to the next node. The definition of each parameter is

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2.3.21}$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

$$\hat{h}_t = tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

$$h_t = (1 - z_t) \odot \hat{h}_t + z_t \odot h_{t-1}$$

where r and z denote the reset gate and update gate, W, U, b represent the weight and bias, $\sigma(\cdot)$ stands for sigmoid function, which transforms the data to the interval $(0, 1)$, x_t, h_t represent the input and output vectors, $\hat{h}_t$ means the candidate activation vector. For updating gate z, the range is within interval $(0, 1.0)$, which is closer to 1.0, the more data will be represented in memory, the closer it tends to 0, the more forgotten will be represented.

Compared with LSTM, GRU only has two gates and is similar to LSTM, where LSTM is also designed to capture the long-term dependencies. Moreover, the structure of GRU is similar to the traditional RNN. In LSTM, the output gate is to control the cell memory and transfer data into the next unit. GRU directly transfers the information to the next unit without the control of cell memory and apply the reset gate to control the previous information.

Thus, the main differences between LSTM and GRU are: (1) LSTM has three different gates, but the GRU only has two different gate; (2) LSTM contains the cell memory, the GRU does not, which directly calculates the outputs; (3) the update gate in GRU is similar to the fusion of the input gate and forget gate in LSTM. Table 2.3.2 shows the advantages and disadvantages of these three methods which are trained with the time series data.

MLP contains three different layers, the input feature vector gets the output layer through hidden layer transformation, the classification result is obtained from the output layer. The activation function such as sigmoid function or hyperbolic tanh function are used to simulate the response of neurons, the backpropagation algorithm is used in the training algorithm (Werbos, 1990). However, with deepening the depth of neural network

layers, optimization function is more and more likely to fall into the local optimal solution and may raise gradient vanishing problem. In order to overcome the gradient vanishing problem, activation functions such as ReLU and Maxout replace sigmoid, which forms the basic form of DNNs. In the structure of fully connected DNNs, the lower layer neurons and all the upper layer neurons form a connection, which lead to the expansion of the number of parameters. Thus, CNNs were proposed to use weight sharing, RNN was suggested to deal with sequence data.

Table 2.3.2 The advantages and disadvantages of different methods for sequence data

	Advantages	**Disadvantages**
RNN	Processing the sequence data Processing inputs of any length Sharing the weights across the time steps	The computation is slow. The results may be affected by gradient exploding and gradient vanishing;
LSTM	It can fit the sequence data. Solve the problem of gradient vanishing. Capture the long-term dependencies. More robust	Not conducive to parallelization The calculation is time-consuming. Gradient vanishing when sequence data is large
GRU	It has less parameters. It has the faster convergence.	Gradient vanishing when sequence data is large. Poor expression performance when sequence data is large.

The fully connected LSTM does not take into consideration of spatial correlation and encompasses a wreath of redundant spatial data. ConvLSTM structure for precipitation not only establishes temporal relations in LSTM, but also describes local spatial features in CNN. Moreover, ConvLSTM is better than LSTM in the aspect of finding

spatiotemporal relations (Xingjian, Chen, Wang, Yeung, Wong, & Woo, 2015), the ConvLSTM is well investigated in gesture recognition, human behavior recognition, image and text classification etc. (Zhu, Zhang, Shen, & Song, 2017; Luo, Liu & Gao, 2017; Zhang, Zhu, Shen, Song, Afaq Shah, & Bennamoun, 2017; Liu, Zhou, Hang, & Yuan, 2017; Breuel, 2017; Si, Chen, Wang, Wang, & Tan, 2019). The equation of ConvLSTM is shown as

$$
\begin{aligned}
f_t &= \sigma\left(W_{xf} * \chi_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \circ C_{t-1} + b_f\right) \\
i_t &= \sigma(W_{xi} * \chi_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
o_t &= \sigma(W_{xo} * \chi_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \circ C_t + b_o) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \ tanh(W_{xc} * \chi_t + W_{hc} * \mathcal{H}_{t-1} + b_c) \\
\mathcal{H}_t &= o_t \circ tanh(C_t)
\end{aligned}
\tag{2.3.22}
$$

where χ and C denote the inputs and the cell outputs, $\mathcal{H}$ stands for the hidden state; f_t, i_t, and o_t represent the forget gate, input gate and output gate of the ConvLSTM, which are 3D spatiotemporal relationship reflected in rows and columns, $*$ refers to the convolution operator, $\circ$ means the element-wise production.

2.4 Human Behavior Recognition in Deep Learning

Human behavior understanding refers to analyze and recognize human motions, and describes it by using natural language. A sequence of motion pictures is considered as the traversal process of these static actions. The joint probability of traversal process is calculated, its maximum value is taken for classification. Gesture recognition is regarded as a part of behavior recognition was also studied in this research, it is a cooperative research field which is related to pattern recognition, computer vision, etc. The term of sign language recognition refers to the whole process of tracking human gestures, recognizing the representations, and converting them into semantically meaningful commands (Rautaray & Agrawal, 2015; Lu, Yan, & Nguyen, 2021). Traditional

approaches for human behavior recognition require the detection methods at first to find the region of interest (ROI).

In terms of behavior recognition, the focus of early research work was on treating human body as a geometric model (Aggarwal & Cai , 1997), such as stick-figure model (Guo, Xu, & Tsuji, 1994), cardboard person model (Aggarwal, Cai, Liao, & Sabata, 1997), and 3D cylinder model (Rohr, 1994). A star skeleton model was proposed, which extracted the five most obvious inflection points of target contour and used them for a human body model regarding to behavior analysis (Fujiyoshi, Lipton, & Kanade, 2004). Afterwards, feature extraction methods were employed to describe human motions and distinguish human behaviors, such as moving direction, trajectory, shape, velocity, etc., which utilizes principal components analysis (PCA) to reduce the dimension of features. Human behavior recognition was implemented, the relevant feature extraction methods were proposed, such as Haar (Papageorgiou, Oren, & Poggio, 1998), histogram of oriented gradient (HOG) (Dalal & Triggs, 2005), etc. Moreover, the classifiers are adopted in the feature space to achieve object classification, these classifiers are operated based on the subset of regions in each image (Zitnick, & Dollár, 2014; Uijlings, Sande, Gevers, & Smeulders, 2013; Gould, Gao, & Koller, 2009), or the whole image in sliding windows.

Human behavior recognition is to understand the predefined behaviors automatically so as to reduce the workload of security staff. To recognize and analyze the periodic motions, spatiotemporal model (Rui & Anandan, 2000) and periodic model (Cutler & Davis, 2000) were proffered. Human behavior recognition algorithms mainly split into twofold which enclose template-matching-based methods and space-state-based methods.

Pertaining to the template-matching methods, the algorithms are easy to be implemented which have less time overhead, a positive effect is based on human behavior recognition with more substantial differences. However, it is hard to recognize human behaviors that have minor differences, it is sensitive to the changes of motions and noises. The motion energy image (MEI) has been employed for motion-based object recognition

by describing how objects move and where the motion occurs (Bobick & Davis, 1996). The MEI shows spatial distribution of contours and energy of motion. MEI algorithm, relied on motion history image (MHI), is a vision-based template method that expresses the target motion in the form of image brightness by calculating pixel changes at the same position.

Furthermore, the combination of MEI and MHI was proposed (Bobick & Davis, 2001) with template matching in order to replace 3D spatiotemporal data which is embarked on recognizing very simple human behaviors. MEI and MHI were constructed for foreground images. While the MEI determines the spatial location of target motion in the image sequence, MHI reflects motion intensity of the target object in various positions. Dynamic time warping (DTW) algorithm (Brendel & Todorovic, 2009) was usually employed in the exemplar-based method for nonlinear matching. Although DTW is simple, the algorithm is robust to be applied to classification through the sequence of images. However, DTW algorithm requires a large deal of computations and is lack of considering dynamic characteristics between adjacent time sequences. In practice, adjacent frames in the sequence are highly correlated in time and space.

A template-based method was proposed by adopting 3D spatiotemporal volume for human behavior recognition (Rodriguez, Ahmed & Shah, 2008), which is a typical single-layered approach. For the space-state-based methods, each static pose or motion state in the image sequence is taken as a node, the state nodes are connected by a given probability. Hidden Markov model (HMM) has been broadly utilized in the prediction, estimation, detection, and behavior recognition of image sequences. The two hierarchical layers of HMMs can recognize human behaviors which contain much complex sequential structures (Nguyen, Phung, Venkatesh, & Bui, 2005).

Naïve Bayesian classifier (NBC) as the simplest probabilistic model is implemented for human behavior recognition. Furthermore, Dynamic Bayesian networks (DBNs) (Damen & Hogg, 2009) employed prior knowledge to establish causal relationship between visual features so as to deal with inherent uncertainty in video processing. The

variables are added or deleted to reflect various correlations amongst variables without affecting the training algorithm. The hidden states that contain multiple levels in DBNs represent human behaviors hierarchically (Dai, Di, Dong, Tao, & Xu, 2008). Therefore, it has good interpretability, the topological results are accurate and easy to be employed for understanding the semantics.

HMMs and DBNs are applied to identify consistent action behaviors. Under the premise of sufficient training data, both models can reliably identify various behaviors and have a strong anti-interference capability, but are hard for recognizing complex behaviors with time series structures. Support vector machine (SVM) (Cortes & Vapnik, 1995) has the training errors because of constraints of the optimization problem with the confidence being minimized so as to reach optimization goal. SVM has the unique advantage in solving the problems related to the small number of training samples, nonlinear model, high dimensional training data, and local minimum issue.

There are plenty of studies using syntactic meaning (Joo & Chellappa, 2006) to represent human behaviors as symbols. Stochastic context-free grammars (SCFGs) model the high-level sequential activities by adopting a HMMs layer to present the activities as an atomic-level action, use stochastic parsing to recognize high-level activities (Ivanov & Bobick, 2000). A more reliable method can recognize multitasked activities by extending SCFGs (Moore & Essa, 2002). The SCFGs also were proposed to solve the segmentation problems for multiple objects, because context-free grammars (CFGs) method shows the possibility of segmenting and tracking objects in semantic level (Minnen, Essa, & Starner, 2003). Moreover, CFGs were used to recognize human activities and describe the semantics of these activitities directly.

Human behavior recognition is implemented by using unsupervised learning which does not require pretrained data, the classes among the training data are unlabeled. The objective of unsupervised learning is to study the unlabeled training data so as to reveal the regular pattern for further analysis. Clustering is the most popular method in unsupervised learning.

A multi-observation hidden Markov model (MOHMM) was proposed to detect abnormal activities which were much effective to noisy and sparse datasets (Xiang & Gong, 2005). Self-organizing map (SOM) approach is a kind of competitive and unsupervised neural networks, which maps high-dimensional input data onto low-dimensional space, maintains the topological structure of input data in high-dimensional space, transform similar sample points in high-dimensional space to adjacent neurons of the network output layer.

SOM has been applied to detect rare events, a method for classifying new events using Gaussian mixture model (GMM) from SOM maps attains the recognition of abnormal behaviors (Petrushin V. A., 2005). Niebles *et al*. utilized an unsupervised learning method combined with the spatiotemporal words by extracting both spatiotemporal interest points to achieve human action categorizations, which categorizes multiple actions with complex video sequences (Niebles, Wang, & Fei-Fei, 2008).

The 2D CNN-based deep learning methods took the advantages to studied and developed in human behavior recognition (Carreira, & Zisserman, 2017; Donahue, Anne, Guadarrama, Rohrbach, Venugopalan, Saenko, & Darrell, 2015; Simonyan, & Zisserman, 2014; Tu, Li, Zhang, Dauwels, Li, & Yuan, 2019). The 2D CNN-based deep learning methods can be chiefly categorized as frame-based aggregation models and two-stream CNN-based models (Zong, Wang, Chen, Wang, Wang, & Potgieter, 2020). For the frame-based aggregation models, the feature maps will be extracted from each video frame by adopting CNN model and then aggregates the information to the recurrent neural network (RNN); Two-stream CNN-based models took the use of spatial CNN stream and temporal CNN stream to acquire the appearance and motion feature maps. Specifically, optical flow is normally adopted to extract the motion feature maps, thus, the speed and direction of each pixel will be acquired. Furthermore, the 3D CNN-based deep learning methods has been studied and developed for human behavior recognition, which shows the promising results than previous studies (Ji, Wei, Yang, & Kai, 2013; Karpathy, Toderici, Shetty, Leung, Sukthankar, & Li, 2014; Simonyan, & Zisserman, 2014; Tran, Bourdev, Fergus, Torresani, & Paluri, 2015).

The remarkable performance of convolutional neural network in the application of computer vision in 2D space has led to the research on the application of 3D space. The convolution structure in temporal domain is well studied in the past few years, which is usually expanded from space domain (x, y) to 2D convolution structure in time domain (x, y, t). Table 2.4.1 shows the brief introduction and comparsion of three main approaches for human behavior recogntion.

A P-CNN was proposed for action prediction, which extracts the key points and local features, aggregates them together (Chéron, Laptev, & Schmid, 2015). A 3D CNN for human behavior recognition collects spatio-temporal information for feature map extraction (Ji, Xu, Yang, & Yu, 2013). Compared with 2D CNNs, no matter how many channels 2D CNNs have, a convolutional kernel only outputs one feature map, which means, only the spatial relationship exists, the temporal relationship will be gone.

C3D extends 3 × 3 convolution to 3 × 3 × 3 convolution. For human behavior recognition, temporal information is important to improve the classification. Compared with 2D CNN, 3D CNNs have a large number of parameters, which make training more difficult and requires more training data. However, C3D processes multiple frames at a time, compared with other types of methods, the calculation efficiency is very high. 3D convolution is to stack multiple consecutive frames to form a cube, then apply the 3D convolution kernel to the cube (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015). 3D convolution kernel is factorized into 2D convolution kernel in spatial domain and 1D convolution kernel in temporal domain which shows 88.1% accuracy based on UCF-101 dataset (Sun, Jia, Yeung, & Shi, 2015). A region convolutional 3D network (R-C3D) was suggested for detecting human behaviors in the end-to-end way (Xu, Das, & Saenko, 2017).

In this structure, each feature map in the convolutional layer will be connected to multiple adjacent frames in the proceed layer so as to capture the motion information. The 3D convolution kernel can merely extract one type of features from the cube, because the weight of convolution kernel in the whole cube is as same as the shared weights.

Table 2.4.1 The brief introduction and comparsion of three mian approaches for human behavior recogntion

	Brief Introduction	**Pros**	**Cons**
iDT	Using the optical flow and SURF between video frames to obtain trajectories in the video footages, and the features are extracted along the trajectory. The SVM classifier was adopted to acquire the final results.	Highest stability and high reliability	Slow speed
Two-Stream CNN	Optical flow is calculated every two frames in the video footage to obtain temporal information. Then, CNN model is trained for the video frames (spatial) and the dense optical flow (temporal). The two branches of the network classify the categories respectively. Finally, the class scores of the two networks are fused directly (including direct average and SVM) to get the final classification results.	High precision on UCF-101 (up to 96%)	Slow speed with 20 FPS
C3D	The temporal and spatial features of video data are extracted by 3D convolution kernel. These 3D feature extractors operate in the spatial and temporal dimensions, so they can capture the motion information of the video stream. Then a 3D convolution neural network is constructed based on 3D convolution feature extractor.	Fast speed with 300 FPS	Low precision on UCF-101 (only 85%)

A 3D convolution is primarily based on the betterment of DenseNet. 2D convolution in original network is modified and applied to 3D convolution, 2D pooling is revised and supplied to 3D pooling. Moreover, a temporal transition layer (TTL) consists of several convolution kernels which have different sizes, 3D pooling layers are employed to generate the feature maps (Diba, Fayyaz, Sharma, Karami, Arzani, Yousefzadeh, & Van Gool, 2017). A deep network with temporal pyramid pooling (DTPP) is proposed. In

order to construct the representation in an end-to-end way, sparse sampling is carried out on the input video with enough frames. The time pyramid pooling layer is applied to encode frame features into a fixed size representation with multiple time scales to capture the temporal structure of human behaviors in the video (Zhu, Zhu, & Zou, 2018).

Both 2D and 3D networks have a good effect on images and videos (Luvizon, Picard, & Tabia, 2018; Zanfir, Marinoiu, & Sminchisescu, 2018; Garcia-Hernando, Yuan, Baek, & Kim, 2018; Liu, Shahroudy, Wang, Duan, & Kot, 2018), but for spatiotemporal integration, the huge memory cost is still too high. Thus, a mixed 3D/2D Convolutional Tube (MiCT) combines 2D / 3D convolution module to generate more feature maps with in-depth and rich information (Zhou, Sun, Zha, & Zeng, 2018).

Two-stream CNNs were proposed to recognize human actions, which contain both spatial stream (single frame) and temporal stream (multiframe optical flow), achieved a satisfactory performance. The Two-stream-based CNNs independently took advantage of dense optical flow from static frames and inter frames for pattern recognition; finally, it carried out the class score fusion (Simonyan & Zisserman, 2014). After the processing by using last fully connected layer, a softmax layer was ultilized to output the final result, which has two softmax-layer outputs. Each softmax has its own independent loss, the final loss is the superposition of two loss. The updated parameters are returned. Thus, the convolutional layer of this network is constant, the parameters of these two layers are shared with each other, which only modifies the final loss. Moreover, Zhu et al. proposed a novel hidden two-stream CNNs, which captures motion information between each frame implicitly, it is an end-to-end approach and does not require to compute the optical flow (Zhu, Lan, Newsam, & Hauptmann, 2018).

A Two-stream inflated 3D convolutional neural network (I3D) was proposed to overcome the disadvantages of two-stream CNNs, which shows 80.2% accuracy on HMDB-51 dataset. It expands 2D convolution in two-stream structure to 3D convolution, the input of optical flow is also added into the network. The video flow and stacked optical

flow are input into 3D convolution network respectively to get the output results, the final result is the fusion of two flows (Carreira, & Zisserman, 2017).

P3D method approximates the original 3 × 3 × 3 convolution with a 1 × 3 × 3 spatial convolution and a 3 × 1 × 1 temporal convolution. By combining three different modules, P3D ResNet is obtained. P3D ResNet optimizes C3D in terms of the number of parameters and processing speed (Qiu, Yao, & Mei, 2017). A R(2+1)D method utilizes 2D convolution to extract spatial information, 1D convolution is employed to associate the information extracted in the proceed step to get the temporal information. It is used to implement a better nonlinear operation for spatiotemporal information extraction and increase the complexity of the model, which more suits for fitting (Tran, Wang, Torresani, Ray, LeCun, & Paluri, 2018).

Classic iDT features and two-stream features are combined to propose the trajectory-pooled deep-convolutional descriptors (TDD) approach. TDD treats the pretrained two-stream networks as a fixed feature extractor. After obtained the two features, TDD takes use of spatiotemporal normalization to ensure that the numerical range of each channel is approximate to the same, and utilizes channel normalization to gurantee that the numerical range of the description vector of each spatiotemporal location is approximately the same, then applies trajectory pooling and Fisher vector to build TDD features, finally takes advantage of SVM for classification (Wang, Qiao, & Tang, 2015).

A spatiotemporal structure based on two-stream CNNs was proposed by adding the convolutional fusion layer and temporal fusion layer, which have the similar number of parameters with previous networks (Feichtenhofer, Pinz, & Zisserman, 2016). In the same year, a ResNet-based Two-Stream CNNs extend 2D ResNet into the spatiotemporal domains, a skip stream connects the temporal stream and spatial stream to recognize the input sequences (Feichtenhofer, Pinz, & Wildes, 2016).

Because adjacent frames have information redundancy, dense video sampling is not required. The temporal segment networks (TSN) adopt sparse sampling strategy associated with the information of the entire video, and utilizes dual-stream network to

obtain video clips and various scores, and finally output them through softmax layer (Wang, Xiong, Wang, Qiao, Lin, Tang, & Van Gool, 2016). The TSN is improved to achieve the temporal and relational reasoning between the video frames at different time (Zhou, Andonian, Oliva, & Torralba, 2018). Since not every frame in the video contains useful information, TSN was presented to extract local features, which are aggregated to the global features (Lan, Zhu, Hauptmann, & Newsam, 2017). The Two-Stream network was used to extract feature maps, the vector of locally aggregated descriptors (VLAD) is applied to get the video representation so as to achieve behavior recognition. It is found that the two branches of video frames (spatial) and optical flow (temporal) are best processed separately (Girdhar, Ramanan, Gupta, Sivic, & Russell, 2017).

Not all frames in a video are equally vital to the recognition. If they are treated equally, valuable frame information will be submerged in other irrelevant frames. Zhu et al. alternately optimized key frame mining and classification. The network input N video clips and output the score of each category corresponding to each clip. If the category corresponds to real tags, randomly merging is applied, otherwise, the maxout merging is adopted, the most responsive video segment will be the key frame (Zhu, Hu, Sun, Cao, & Qiao, 2016). A hierarchical and spatiotemporal pyramid structure is applied to fuse the both spatial and temporal features, which achieved 94.6% accuracy based on UCF-101 dataset (Wang, Long, Wang, & Yu, 2017).

In a video, the movement between consecutive frames is very small, by adopting the iDT algorithm to densely-sampled feature points, using optical flow to track them is able to get a better video representation. The video representation obtained by comprehensive coding, a long-term dynamic process should be taken into account, thus, a temporal and liner encoding (TLE) is embedded into the CNNs so as to fuse and encode the feature map, the final feature representation is obtained by using the element-wise multiplication of the convolution of different video frames or clips with bilinear models (Diba, Sharma, & Van Gool, 2017).

A low-rank approximation of second order pooling (attentional pooling) is to replace mean pooling or max pooling in the last pooling layer of CNN network so as to achieve the behavior recognition (Girdhar, & Ramanan, 2017). A spatiotemporal vector of locally max pooled features (ST-VLMPF) has been introduced for local deep features encoding which aggregates multiple features and solved the problem of incorrect labels that assigned to the network inputs (Duta, Ionescu, Aizawa, & Sebe, 2017).

Wu et al. proposed to train the compressed video directly based on deep neural networks, by adopting motion vectors as the input of the network, the difficulty of modeling sequential motion information is alleviated. Due to a huge size of original video and a large amount of temporal information, useful information is usually submerged with a large number of irrelevant data, through video compression, the amount of information will be reduced (Wu, Zaheer, Hu, Manmatha, Smola, & Krähenbühl, 2018).

Both human behavior recognition and human pose estimation are closely related (Newell, Yang, & Deng, 2016; Artacho, & Savakis, 2020; Huang et al., 2020; Cheng et al., 2020; Zhang et al., 2020; Wang, Tighe, & Modolo, 2020; Xu et al., 2020; Zhang, Huang, & Wang, 2020; Isogawa et al., 2020; Kocabas et al., 2020; Kundu et al., 2020; Li et al., 2020; Mitra et al., 2020), a novel model of dynamic skeletons called Spatial Temporal Graph Convolutional Networks (ST-GCN) was put forwarded in 2018 by automatically learning both spatial pattern and temporal pattern to achieve human behavior recognition (Yan et al., 2018). It performs pose estimation on videos and constructs spatiotemporal graph based on skeleton sequences. Spatiotemporal graph convolution network (ST-GCN) will gradually generate higher-level feature maps based on the graph. It will be classified by using standard softmax classifier to the corresponding category.

The dynamic skeleton models are naturally represented by time series of human positions in both 2D and 3D coordinates. Then, human behavior recognition is attained by analyzing its action pattern. The ST-GCN model consists of nine layers of spatiotemporal graph convolution units, the first three layers have 64 channels for output,

the following three layers have 128 channels for output and the rest of three layers have 256 channels for output. All of these layers have nine temporal kernel size, ResNet was applied to each unit. The proposed ST-GCN outperforms previous skeleton-based model. The combination of skeleton-based model and frame-based model further improves the performance of human behavior recognition. In addition, the ST-GCN captures motion information in dynamic skeleton sequences.

Human behavior recognition is able to be achieved by adopting skeleton-based multi-person pose estimation (Benzine, Chabot, Luvison, Pham, & Achard, 2020;Chen, Ai, Chen, Zhuang, & Liu, 2020; Fabbri, Lanzi, Calderara, Alletto, & Cucchiara, 2020;), which is mainly divided into two frameworks: Two-step framework, part-based framework. The two-step framework is to detect the pedestrian firstly, get the boundary frame, detect the key points of human body in each boundary frame, connect them into a human shape. The disadvantage is that the influence of the detection frame is too large, the missed detection, false detection, the size of IoU will affect the results. The representative method is RMPE (Fang, Xie, Tai, & Lu, 2017). The part-based framework is to detect the key parts of each human body in the whole frame, and splice the detected parts into a human shape. The disadvantage is that different parts of people will be separated, the representative method is OpenPose.

A 3D skeleton sequence is set forth to regularize LSTM network (Mahasseni, & Todorovic, 2016). Different from the previous work, an attention-enhanced graph convolutional LSTM (AGC-LSTM) was implemented for human behavior recognition, which adopts skeleton information as the input of LSTM, spatiotemporal feature maps are extracted by using AGC-LSTM, while the LSTM has a strong ability to acquire temporal features. Combined LSTM with graph structure together, the model effectively utilizes spatiotemporal information of input images (Si, Chen, Wang, Wang, & Tan, 2019).

Recurrent neural networks (RNNs) are widely utilized in action recogntions by encoding the sequence and predicting the actions frame by frame; but it is difficult for

RNNs to maintain long-term memory in practice (Ma, Sigal, & Sclaroff, 2016; Singh, Marks, Jones, Tuzel, & Shao; Yeung, Russakovsky, Mori, & Fei-Fei, 2016; Lev, Sadeh, Klein, & Wolf, 2016). RNNs based on a joint classification regression are proposed to achieve human action detection (Li, Lan, Xing, Zeng, Yuan, & Liu, 2016). 3D CNNs are proposed to learn the spatiotemporal relationships, the temporal evolution of the learned features in each timestep was considered by using an LSTM-based recurrent neural networks (RNNs) to classify each sequence (Baccouche, Mamalet, Wolf, Garcia, & Baskurt, 2011). In 2011, an unsupervised learning-based method for human behavior recognition method (Le, Zou, Yeung, & Ng, 2011) was presented, the independent subspace analysis (ISA) is extended to 3D video data, unsupervised learning algorithm is applied to model the video blocks. Moreover, the ISA algorithm based on the small input block convolutes the learned network and the larger input image, combines the responses from the convolution process as the input of the next layer, then applies the obtained description method to the video data.

A spatiotemporal long short-term memory network (ST-LSTM) has extended the traditional LSTM to spatiotemporal domain (Liu, Shahroudy, Xu, & Wang, 2016). The ST-LSTM has been improved with the global context memory, the global context attention mechanism is brought into the ST-LSTM, where the attention mechanism obtains the structure information and also eliminates the interference of noises (Liu, Wang, Hu, Duan, & Kot, 2017).

Sharma et al. took use of attention mechanism to human action recognition, which mingles soft attention model with the LSTM to cope with long sequence data and learn the key point of the movement (Sharma, Kiros, & Salakhutdinov, 2015). An end-to-end spatiotemporal attention model was accommodated for human behavior recognition (Song, Lan, Xing, Zeng, & Liu, 2017). The end-to-end recurrent pose-attention network (RPAN) was expounded by using CNN to generate feature cube, post attention mechanism shares the attention parameters through semantically-related human joints so as to attain high quality of human behavior recognition, which indicates 97.4% accuracy based on PennAction dataset (Du, Wang, & Qiao, 2017).

2D CNN based on the single frame has been employed to extract feature maps and conduct temporal information to recognize gestures (Koller, Ney, & Bowden, 2016; Wu, Ishwar, & Konrad, 2016). Moreover, 2D CNN is expanded to 3D CNN (Liu, Zhang, & Tian, 2016; Molchanov et al., 2016; Huang et al., 2015) so as to learn the motion features by adopting 3D filters in the convolutional layers, which show the positive results for recognizing hand gestures. CNN model was proposed to detect and segment hands in both unlabeled and synthetic dataset, which achieved 82% accuracy based on segmentation and detection (Neverova, Wolf, Taylor, & Nebout, 2014).

The CNN network has been well investigated to solve the image classification and recognition tasks. Moreover, it also has been investigated and implemented for sign language recognition in recent years. A CNN-based method was proposed with Gaussian skin color model and background subtraction to achieve gestures recognition from the camera images. The Gaussian skin color model controlled the influence of light on skin color, and the non-skin color of image is filtered out directly, which has 93.80% accuracy from a given dataset (Han, Chen, Li, & Chang, 2016). A two-stage CNN architecture (HGR-Net) was given, where the first stage was proposed to determine the region of interest by performing pixel-level semantic segmentation, the second stage is to recognize hand gesture (Dadashzadeh, Targhi, Tahmasbi, & Mirmehdi, 2019). Moreover, the combination of fully convolutional residual network with spatial pyramid pooling was adopted at the first stage, the result shows that proposed architecture improves 1.6% accuracy for recognition by using OUHands dataset.

A deep convolutional network was proposed with multidimensional feature learning approach (MultiD-CNN) to recognize the gestures from the RGB-D videos (Elboushaki, Hannane, Afdel, & Koutti, 2020). The method took use of 3D ResNet for training a model with both spatiotemporal features, the long short-term memory (LSTM) for processing temporal dependencies and the proposed method is outperformed compared with the previous methodsbased on different datasets. Chen et al. implemented the spatiotemporal attention with dynamic graph constructed (DG-STA) method to achieve hand gesture recognition. It took advantage of fully connected graph and self-attention mechanism to

learn the node features and edges from the hand skeleton, a novel spatiotemporal mask is applied to reduce the computational cost. According to the experimental results, DG-STA method achieved the superior performance compared with others for recognizing hand gestures (Chen, Zhao, Peng, Yuan, & Metaxas, 2019).

A deep-learning-based method was proposed by adopting two ResNet CNNs and soft attention with fully connected layer to recognize dynamic gestures. Moreover, a method was proposed to condense a digital video into a single RGB image and passed to the model for the final classification. The experimental result based on public datasets shows that the proposed method is able to improve the accuracy compared with other methods (Dos Santos, Samatelo, & Vassallo, 2020). Three representations of depth sequences are constructed, which includes dynamic depth images (DDI), dynamic depth normal images (DDNI), and dynamic depth motion normal images (DDMNI) from the depth maps to capture the spatiotemporal information by adopting the bidirectional rank pooling, the CNNs-based model is considered to achieve gesture recognition. The proposed model was evaluated based on large-scale isolated gesture recognition at the ChaLearn LAP challenge 2016 and the model was achieved the growth of 16.34% accuracy on the IsoGD dataset (Wang, Li, Liu, Gao, Tang, & Ogunbona, 2016).

Two different deep learning methods were fused to achieve gesture recognition. The convolutional two-stream consensus voting network (2SCVN) to explicitly simulate the short-term and long-term structures of RGB sequences, and 3D Depth-Saliency CNN stream (3DDSN) was used to present the motion features. The proposed methods have been evaluated based on ChaLearn IsoGD dataset with 4.47% growth of accuracy compared with other models in 2016 (Duan, Zhou, Wan, Guo, & Li, 2016). Molchanov et al. designed a dynamic hand gesture recognition method by adopting a recurrent 3D CNN model. Four kinds of visual data were fused to boost the recognition rate, which includes RGB, depth, optical flow and stereo IR. The proposed model achieved the positive accuracy rate based on ChaLearn dataset, which has 1% growth compared with other models (Molchanov, Yang, Gupta, Kim, Tyree, & Kautz, 2016). A hand gesture recognition and identification model was proposed based on the two-stream CNNs, the

depth map and optical flow as the inputs were utilized in this method. The proposed model has 18.91% accuracy improvement based on MSR Action3D dataset compared with the relevant models (Wu, Ishwar, & Konrad, 2016).

Rastgoo et al. set forth the model for hand sign language recognition by utilizing the restricted Boltzmann machine (RBM) for visual data. The model took use of RGB and depth as the input: Original image, cropped image, and noisy cropped image. The CNN is used to detect the hand in each image, three forms of the detected hand images are generated to the RGB and depth will be inputted to the RBM. The output of the RBM will be fused to recognize the sign label. As the result, the proposed model has been able to achieve significant improvement based on four different public datasets compared with the state-of-the-art models (Rastgoo, Kiani, & Escalera, 2018). After the RBM model, Rastgoo et al. proposed a deep cascaded model for sign language recognition from the videos in 2020. The model employed three spatial features: Hand features, extra spatial hand relation (ESHR), and hand pose (HP) features which were fused in the model and feed into the LSTM for temporal feature extraction. The SSD model was also adopted for hand detection. The proposed model was evaluated based on IsoGD dataset, which achieved 4.25% accuracy improvement compared with others (Rastgoo, Kiani, & Escalera, 2020).

Chapter 3
Methodology

The main content of this chapter is to clearly articulate research methods so as to satisfy the objectives of this thesis. The chapter mainly covers the processes of data preparation and data augmentation. Moreover, the details of proposed methods for human behavior recognition will be also detailed. Finally, the experimental environment will be explicated in this chapter, the implementations with the evaluation metrics will be also detailed. Moreover, the results of proposed methods will be demonstrated in the next chapter.

3.1 Data Collection

There are a plenty of public datasets for human behavior recognition which are provided by multiple research groups, such as Weizmann dataset, KTH dataset, UCF dataset, CAVIAR dataset, CASIA dataset, and BEHAVE dataset. Table 3.1.1 shows a brief description of these datasets.

Table 3.1.1 A brief description of the different datasets

Datasets	**Brief descriptions**
Weizmann	Single person behavior analysis with daily data, static camera
KTH	Single person behavior analysis with daily data, static camera with different view angles
UCF	Realistic action videos collected from the YouTube
CAVIAR	Multi-person behavior analysis
CASIA	Single/multiple person interaction data with different static camera angles
BEHAVE	Multiagent interaction data

In this project, all the experiments were based on the first two public datasets: Weizmann dataset and KTH dataset and our own datasets, the focus of our research is mainly on human behavior recognition by using surveillance videos. Weizmann dataset encapsulates ten classes, each of the classes has nine videos which were shot by using a static camera with single person behavior analysis with ordinary data, the dataset has nine participants involved in total. The resolution of the image samples is 180×144. In our experiments, we chose five classes which cover the categories: Walking, skipping, running, jacking, and jumping. Figure 3.1.1 shows the examples of Weizmann dataset.

(a) Walking (b) Skipping (c) Running (d) Jacking (e) Jumping

Weizmann Dataset

Figure 3.1.1 The examples of the Weizmann dataset

KTH dataset includes a total of 2,391 videos with six actions from 25 participants in four scenarios: Indoor, outdoor, outdoor with amplification, and outdoor with different clothing. The videos in this dataset enclose scaling changes, clothing changes, and lighting changes which were captured by using static cameras. The resolution of this dataset is 160×120. In our experiments, we chose all the classes which have the labels: Walking, running, boxing, handclapping, jogging and handwaving, with the 5,667 video frames in total. Figure 3.1.2 shows the examples of KTH dataset.

(a) Walking (b) Running (c) Boxing (d) Handclapping (e) Jogging (f) Handwaving

KTH Dataset

Figure 3.1.2 The examples of the KTH dataset

In this thesis, we also created our own datasets. The reason why we created our own datasets is firstly to test the proposed models that can be used stably and robustly in the real scenarios; secondly, most of the public datasets only have the video with low resolutions; In the end, our datasets have the higher resolution compared with the public datasets, which are more realistic and suitable at present. Moreover, higher resolution

images are more suitable for data augmentation and are able to produces better segmentation accuracy. In future, we will extend our dataset and publish it for other researches. Our datasets include two parts, our dataset I is the visual data which comprises of the samples from a total of 20 video footages with five classes of human behaviors, which were taken by using a static camera. The resolution of the samples in this dataset is 1280×720. Our dataset I consists of 3,200 frames in total and 2,000 frames were selected for model training, 1,200 frames were picked up for model testing, Figure 3.1.3 shows the example of our own dataset I. Another dataset II contains nine video footages of four classes with the tags: Hello, nice, meet, you, which includes the sign language data collected by ourself. The resolution of this dataset is 960×564. The dataset II contains 3596 frames in total, there are 2500 frames chosen for model training, 1,096 frames were selected for model testing. Figure 3.1.4 shows the samples of our dataset II.

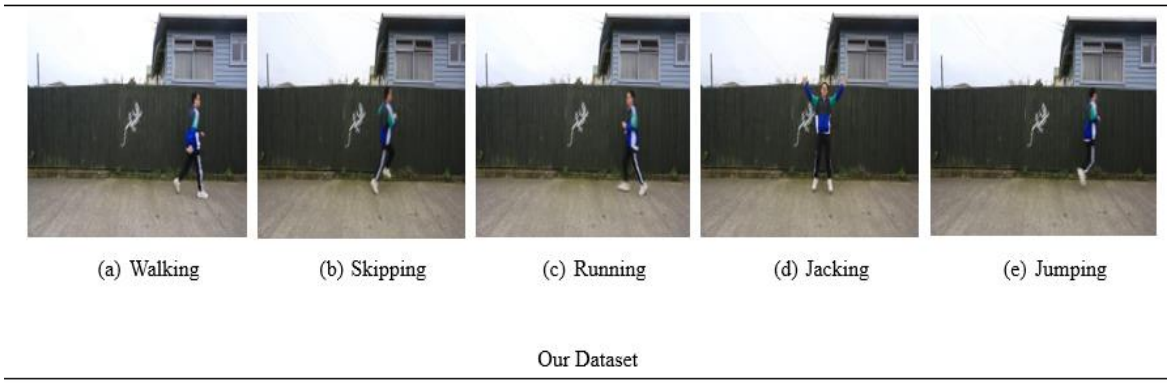


Figure 3.1.3 The examples of our dataset I

Figure 3.1.4 The examples of our own dataset II

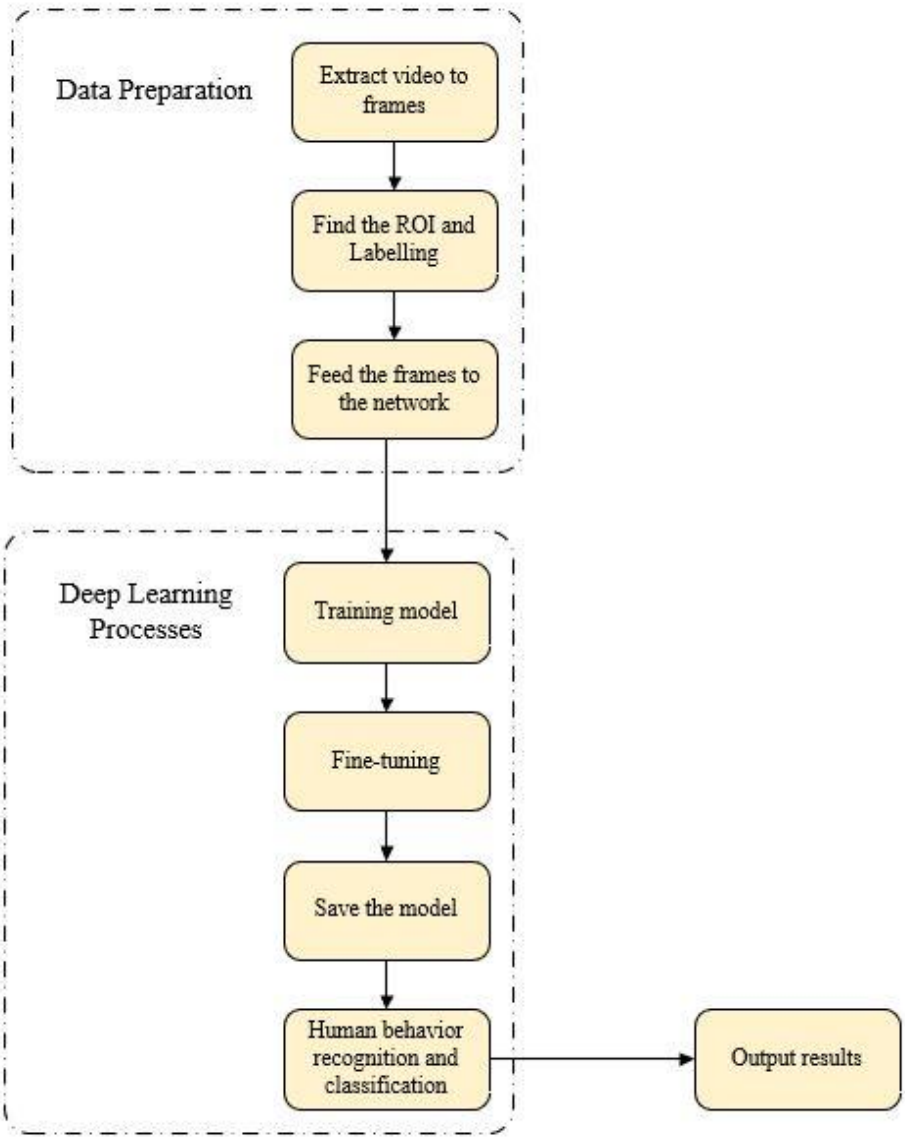


Figure 3.1.5 The steps of deep learning based human behavior recognition

Figure 3.1.5 shows the flowchart of human behavior recognition by using deep learning in our experiments. Moreover, in this research study, we adopted two public datasets to achieve our experimental results. Moreover, there are 5,667 video frames for the KTH dataset; regarding Weizmann dataset, it contains 1500 video frames. All the experiments were operated with a single GPU (RTX 2080Ti) acceleration to reduce the time consuming while training. Moreover, for training the better deep learning models to suit our experiments, the hyperparameters also play pivotal roles in each deep neural networks, such as learning rate, batch size etc. The training configuration for the hyperparameters of our experiment is listed below:

- Learning rate: 0.001
- Batch size: 64
- Momentum: 0.9
- Input image resize: 416 × 416 × 3
- Training and validation split: 80% / 20%

3.2 Data Preparation

As shown in Figure 3.1.1, the steps need to be completed before we send our data into a model of deep neural networks. The first step is to split a video into frames by using MATLAB with simple codes, the simple pseudocode for splitting a video to frames is as followed. In our experiment, we mainly adopt three datasets. Two datasets were acquired from the public datasets, another dataset was collected by our own. Moreover, there are 11,668 video frames in total.

Algorithm 3.2.1 Convert video to frames

Input: The original video

Output: The video frames

```
video = VideoReader(video_file);
frame_number=floor(video.Duration * video.FrameRate);
For i from 1 to frame_number do
    image_name=strcat('save_path',num2str(i));
    image_name=strcat(image_name,'.jpg');
    I=read(video, i);
    imwrite(I, image_name, 'jpg');
    I=[];
End for
```

The second step is to manually find the region of interest (ROI) in each frame and label the ROI with correct class. In our experiments, Weizmann dataset and our own dataset encompass five different classes, KTH dataset contains six different classes. We use a Python based toolbox for video frame labelling, which makes labelling much easier

and time efficient. Figure 3.2.1 shows the shortcut of labelling toolbox that we adopted in this thesis.

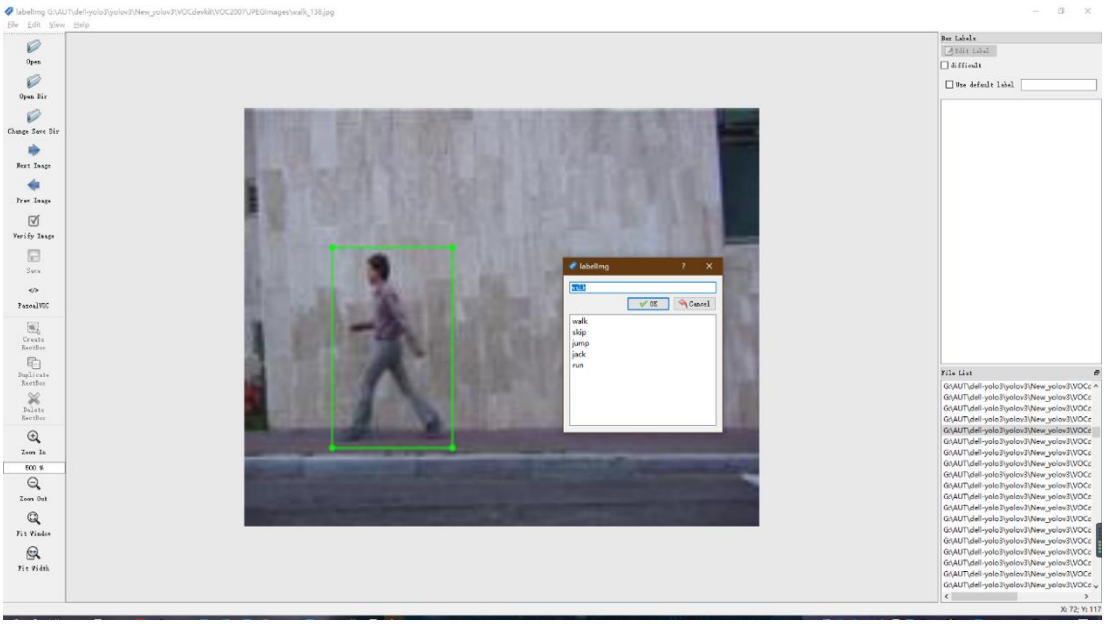


Figure 3.2.1 The shortcut of the labelling toolbox

After finding the ROI, we create a rectangle that contains the correct class, and label each frame. Moreover, the name of each class is predefined by ourself. The ROI of each frame will be stored in a .xml file separately. Figure 3.2.2 shows the example of the stored data in .xml file.

```
<annotation>
    <folder>JPEGImages</folder>
    <filename>walk_138.jpg</filename>
    <path>G:\AUT\dell-yolo3\yolov3\New_yolov3\VOCdevkit\VOC2007\JPEGImages\walk_138.jpg</path>
    <source>
        <database>Unknown</database>
    </source>
    <size>
        <width>180</width>
        <height>144</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>walk</name>
        <pose>Unspecified</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <bndbox>
            <xmin>31</xmin>
            <ymin>48</ymin>
            <xmax>71</xmax>
            <ymax>116</ymax>
        </bndbox>
    </object>
</annotation>
```

Figure 3.2.2 The example of the .xml file for labelling

where the .xml file will store the original frame, the coordinates of ROI in the frame and the class of each ROI. All the ROI information is stored into the .xml file. All ROI information is stored in the .xml files by using the PASCAL VOC format in order to complete the forthcoming training and testing.

3.3 Data Augmentation

A large scale of dataset is the premise of a successful application of deep neural networks (DNNs). Thus, data augmentation methods make a series of random changes to the training image so as to produce training samples and expand the scale of the training dataset. Normally, by increasing the depth and width of neural network, the learning ability of neural network is enhanced, which is convenient to fit the distribution of training data. In the convolution neural network, our experiments show that the depth is more important than width. However, with the increasing the depth of neural networks, the parameters that need to be trained will also increase, which will lead to overfitting. When the dataset is small, too many parameters will fit the characteristics of the dataset.

The data augmentation includes scale, rotation, crop, random noises, etc. (Yan, Zhang, Wang, Paris, & Yu, 2016). In our experiments, we adopt the data augmentation methods in order to increase our samples to achieve better performance of our models, the image brightness and contrast were adjusted in our experiments. Figure 3.3.1 shows the examples of data augmentation for our experiments.

For the brightness adjustment, in order to change the brightness of each image, all the pixel values of each image should be acquired. Then, in order to adjust the brightness of each image, we need to increase/decrease the intensity of each pixel by a constant. The simple pseudocode for the brightness adjustment is provided as follows.

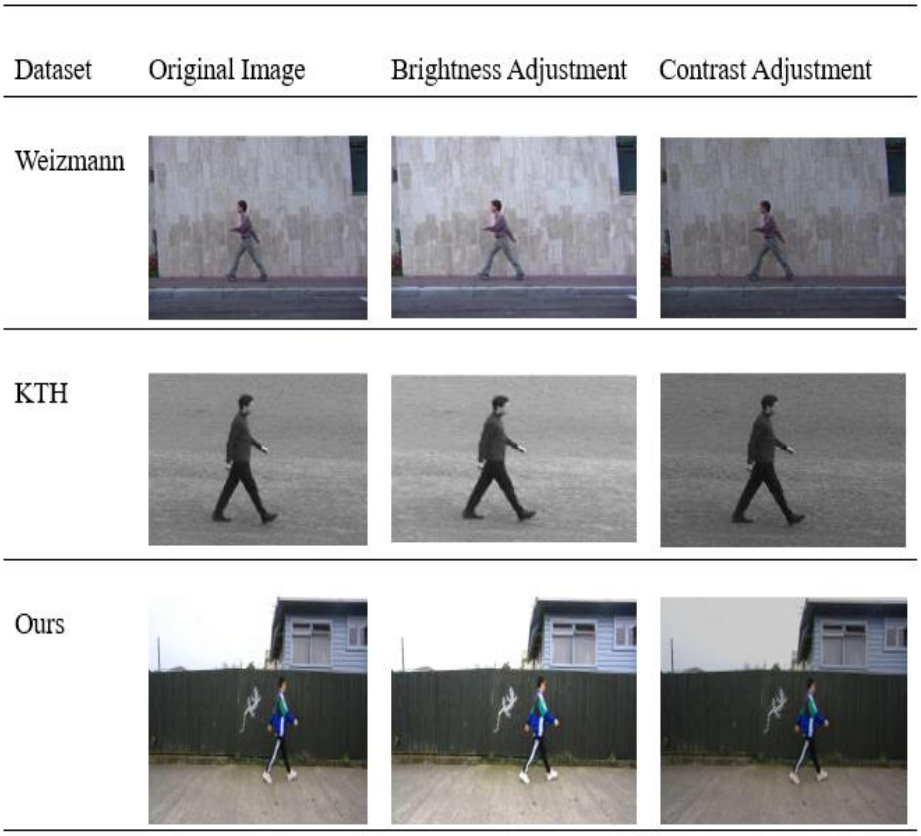


Figure 3.3.1 The examples of data augmentation

Algorithm 3.3.1 Brightness adjustment for images

Input: The original images

Output: The new images which changed by brightness adjustment

orgImage = getPixelValue(r, g, b)

newR = r(orgImage) + brightness

newG = g(orgImage) + brightness

newB = b(orgImage) + brightness

changedPixelValue = RGB(newR, newG, newB)

In Algorithm 3.3.1, given the original three-dimensional image $orgImage$, each pixel is denoted as $\text{getPixelValue}(r, g, b)$, the brightness as a constant value achieves the adjustment, the adjusting range for brightness is within the interval $(0,255)$. In our experiments, we set the value as 1.2, which means that the pixel intensity of input the image will be brighter as 120% of the original image.

The contrast is the difference between the intensity of both the maximum pixel and the minimum pixel in an image. The contrast adjustment is implemented by changing the maximum and minimum intensities of the given pixels, the range for contrast adjustment is within the interval $(-255,255)$. The negative value will lessen the contrast of the image; on the contrary, positive value will raise the contrast of the image. The simple pseudocode for the contrast adjustment is shown as follow.

Algorithm 3.3.2 Contrast adjustment for images

Input: The original images

Output: The new images which changed by contrast adjustment

---

orgImage = getPixelValue(r, g, b)

newR = $F_{contrast} \times (\text{r}(\text{orgImage}) - \text{valueContrast}) + \text{valueContrast}$

newG = $F_{contrast} \times (\text{g}(\text{orgImage}) - \text{valueContrast}) + \text{valueContrast}$

newB = $F_{contrast} \times (b(\text{orgImage}) - \text{valueContrast}) + \text{valueContrast}$

changedPixelValue = RGB(newR, newG, newB)

---

where the $F_{contrast}$ denotes the correction factor, valueContrast refers to the contrast range.

3.4 The Proposed Method

In our experiment, we adjust the layers of deep neural networks. YOLOv3 contains 53 layers in total. Moreover, we adjust YOLOv3 layers by decreasing its convolutional layers and reducing GPU processing time.

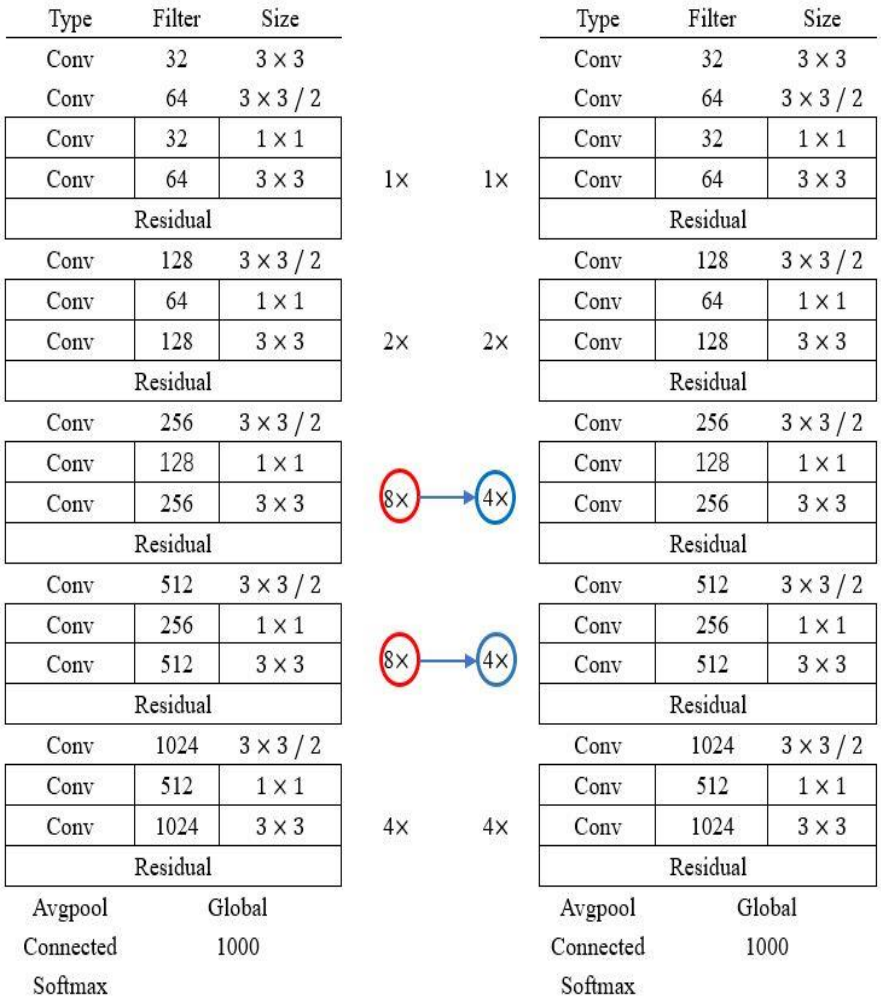


Figure 3.4.1 The revised YOLOv3 network

For our deep neural network, we only have 45 layers in total which are shown in Figure 3.4.1, the left side is the original YOLOv3 network, the right side is the modified YOLOv3 network, we merely decrease half of the 8 layers. In our experiment, we adjust the network to train our model and also utilize it to recognize the simple human behaviors

such as walk, run, skip, jack and jump. Moreover, Figure 3.4.2 shows the example of feature visualization for the revised YOLOv3 network.

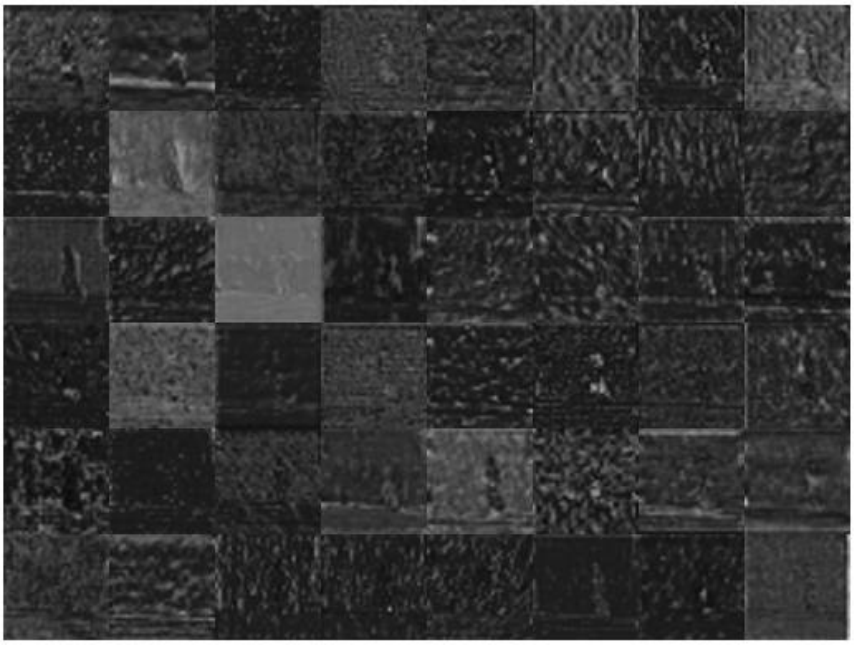


Figure 3.4.2 The example of feature visualization for the revised YOLOv3 network

In this research project, we spell out a spatial attention-based model SKNet which explicates more positive results than previous models in human behavior recognition. Pertaining to the spatial attention module, its focus is on where an informative part is. Figure 3.4.3 shows the structure of spatial attention module. The spatial attention is calculated as

$$M_s(F) = \sigma\left(f^{7\times7}([AvgPool(F); MaxPool(F)])\right) \quad (3.4.1)$$

where $\sigma(\cdot)$ is the sigmoid function, $f^{7\times7}(\cdot)$ is a convolution function with the filter size of 7×7. The spatial attention module applies average-pooling and max-pooling operations along the channel axis and concatenates them to generate an efficient feature descriptor. Consequently, a convolution operation with a 7×7 filter is applied to produce

the feature maps, a sigmoid function for normalization is offered to yield the final feature maps.

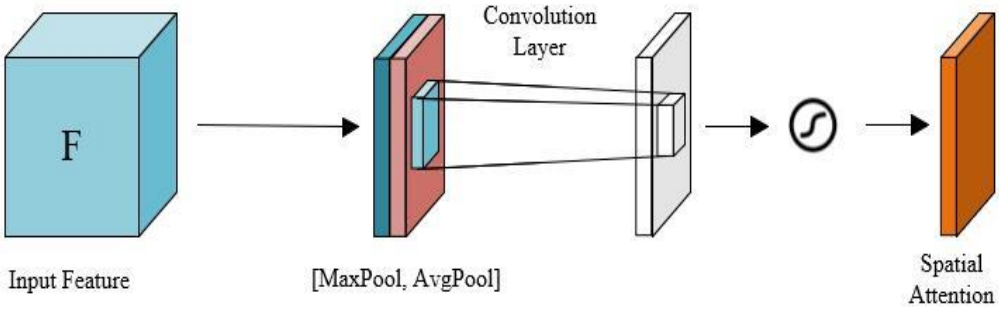


Figure 3.4.3 The structure of spatial attention module

The Selective Kernel Networks (SKNet) (Li et al., 2019) was utilized into this research work to attain human behavior recognition, the SKNet contains multiple selective kernel units which were stacked to a deep neural network.

For split operator, through any given feature map $X \in \mathbf{R}^{H' \times W' \times C'}$, we firstly perform two transformations $\tilde{F}: X \to \tilde{U} \in \mathbf{R}^{H \times W \times C}$ and $\hat{F}: X \to \hat{U} \in \mathbf{R}^{H \times W \times C}$ with convolution sizes of 3 and 5 respectively, where $\tilde{F}$ and $\hat{F}$ are to conduct two transformations which is composed of efficiently grouped convolutions, batch normalization, ReLU function in sequence, and get the output. Moreover, in order to further betterment the efficiency, the conventional convolution of 5 × 5 convolution kernel is replaced by 3 × 3 convolution kernel and dilation size 2.

Fuse operator is to filter the results from multiple branches via an element-wise summation, then embed the global information by simply using global average pooling to generate channel-wise statistics. Furthermore, a compact feature is to enable the guidance for adaptive selections by using a simple fully connected layer, with the reduction of dimensionality for better efficiency.

The select operator utilizes softmax function to apply on the channel-wise digits. where ***a*** and ***b*** denote the soft attention vector for the previous corresponding output, the final feature map **V** is acquired through the attention weights based on various kernels.

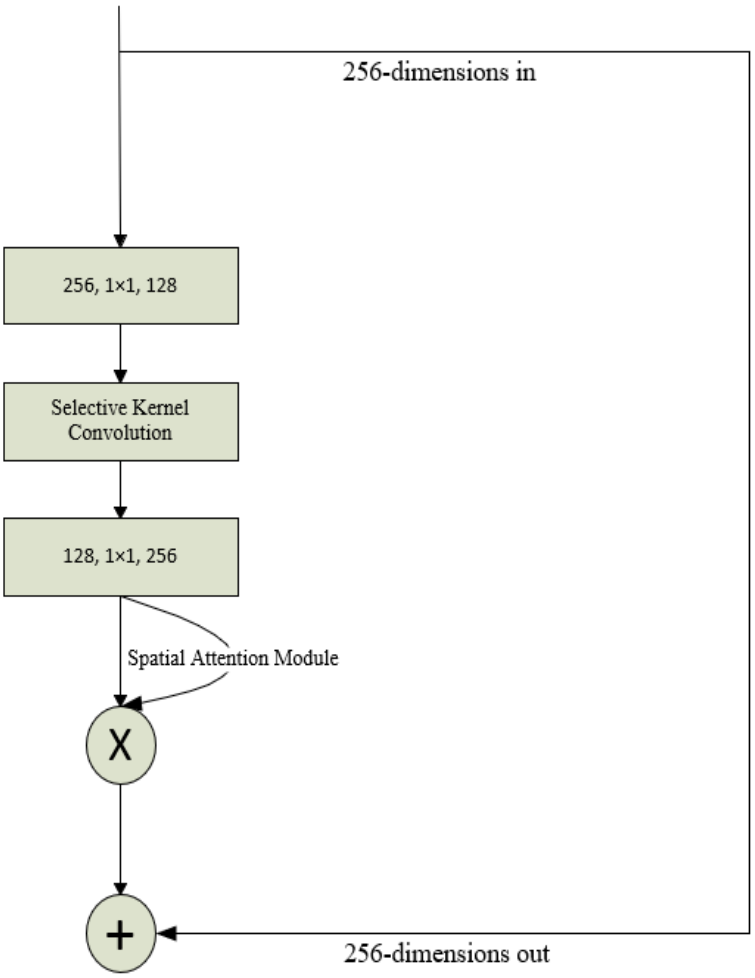


Figure 3.4.4 The structure of a block of SKNet with the spatial attention module

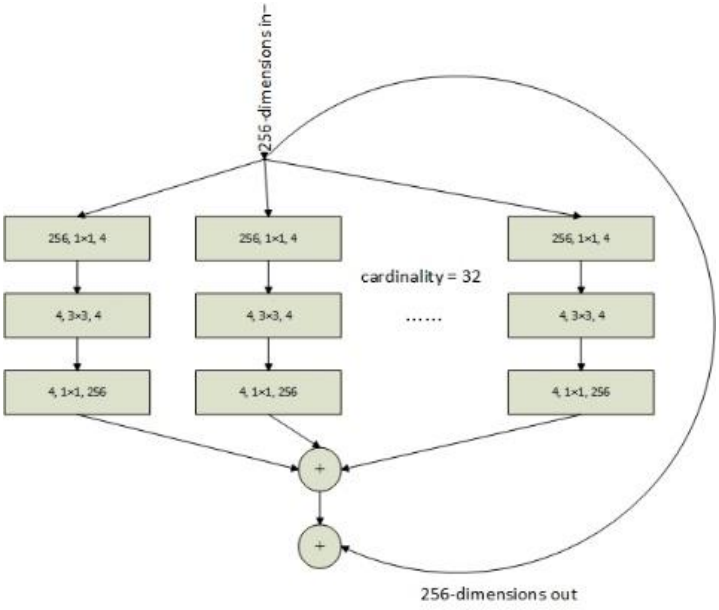


Figure 3.4.5 The structure of single block of ResNeXt model

In this research project, we combine the spatial attention module along with the SKNet to achieve human behavior recognition. Figure 3.4.4 shows a block of SKNet with the spatial attention module. For each block, it is similar to ResNeXt block, but it contains a convolution operation with the filter size 1×1, selective kernel convolution and spatial attention module. Figure 3.4.5 shows the structure of single block of ResNeXt model. ResNeXt adopted the group convolution, which controls the number of groups by cardinality, and each branch adopts the same topology. Thus, ResNeXt model not only improved the accuracy without increasing the complexity of the parameters, but also reduced the computation cost.

Table 3.4.1 The structure of SKNet with attention module

Output	ResNeXt	SKNet	SKNet + Attention
112×112	7×7, 64, stride 2		
56×56	3×3 max pool, stride 2		
56×56	$\begin{bmatrix} 1\times1, & 128 \\ 3\times3, & 128, C=32 \\ 1\times1, & 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, & 128 \\ SK\ unit, & 128 \\ 1\times1, & 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, & 128 \\ SK\ unit, & 128 \\ 1\times1, & 256 \\ Spatial\ Attention \end{bmatrix}\times3$
28×28	$\begin{bmatrix} 1\times1, & 256 \\ 3\times3, & 256, C=32 \\ 1\times1, & 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, & 256 \\ SK\ unit, & 256 \\ 1\times1, & 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, & 256 \\ SK\ unit, & 256 \\ 1\times1, & 512 \\ Spatial\ Attention \end{bmatrix}\times4$
14×14	$\begin{bmatrix} 1\times1, & 512 \\ 3\times3, & 512, C=32 \\ 1\times1, & 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, & 512 \\ SK\ unit, & 512 \\ 1\times1, & 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, & 512 \\ SK\ unit, & 512 \\ 1\times1, & 1024 \\ Spatial\ Attention \end{bmatrix}\times6$
7×7	$\begin{bmatrix} 1\times1, & 1024 \\ 3\times3, & 1024, C=32 \\ 1\times1, & 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, & 1024 \\ SK\ unit, & 1024 \\ 1\times1, & 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, & 1024 \\ SK\ unit, & 1024 \\ 1\times1, & 2048 \\ Spatial\ Attention \end{bmatrix}\times3$
1×1	7×7, global average pool, softmax, 5-d		

Table 3.4.1 shows the structure of SKNet with attention module, it has four stages with {3, 4, 6, 3} SK units. For each SK unit, the parameters will be set as

$$SK\ unit = [M = 2, G = 32, r = 16] \quad (3.4.2)$$

where M is the number of paths which is set as two, that means, two different kernels are aggregated. The group number G is set to 32 that controls the cardinality of each path. r is the reduction ratio which is set to 16 to control the numbers of parameters in the fuse operator.

In this research project, we proposed an ensemble learning-based method to recognize human behavior by using Weka 3 to combine our models together, we gained much better results. Ensemble learning is able to complete the learning tasks together by integrating the multiple learning algorithms to get the better performance of the results. The ensemble learning is to combine multiple weak classifiers in order to get a better and more comprehensive strong classifier. The potential idea of ensemble learning is that even if a weak classifier gets a wrong prediction, other weak classifiers can correct the error back. Ensemble learning is simply divided into two types, the homogeneous ensemble learning and heterogeneous ensemble learning. When all the individual learners are the same, we see them as homogeneous ensemble learning; on the contrary, when all the individual learners are different, we treat them as heterogeneous ensemble learning. In our experiments, we adopt four learners, which include AdaBoost (Freund, & Schapire, 1995; Norvig, & Russell, 2016; Ertel, 2018), Random Forest (Breiman, 2001; Kontschieder, Fiterau, Criminisi, & Bulo, 2015), Bagging (Breiman, 1996), and Naïve Bayes.

The general structure of ensemble learning is to produce a group of individual learners and combine them together. Figure 3.4.6 shows the basic structure of ensemble learning.

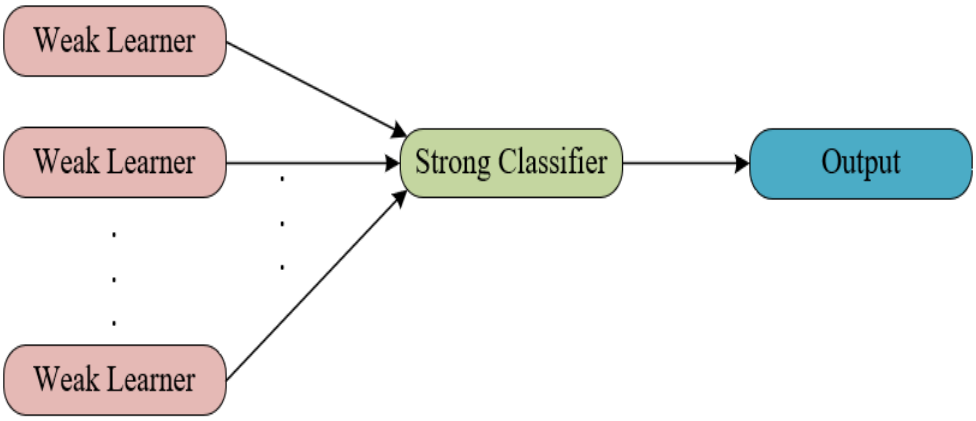


Figure 3.4.6 The basic structure of ensemble learning

Boosting method assists us to get a strong learner by combining a series of weak learners and integrating their learning ability. The adaptive boost (AdaBoost) adjusts the weight of samples base on the basis of previous learners, increases the proportion of samples that have been incorrectly classified, reduce the proportion of samples that have been correctly classified. The learners will focus on those samples that have been incorrectly classified. Finally, these learners are combined into a strong learner by weighting. Specifically, learners with high classification accuracy have higher weights, while learners with low classification accuracy have lower weights.

Decision tree as the component of random forest, decision tree is a rapid and effective method with tree structure, in which each internal node represents a test on an attribute, each branch represents a test output, and each leaf node represents a category. Figure 3.4.7 shows the basic structure of a decision tree.

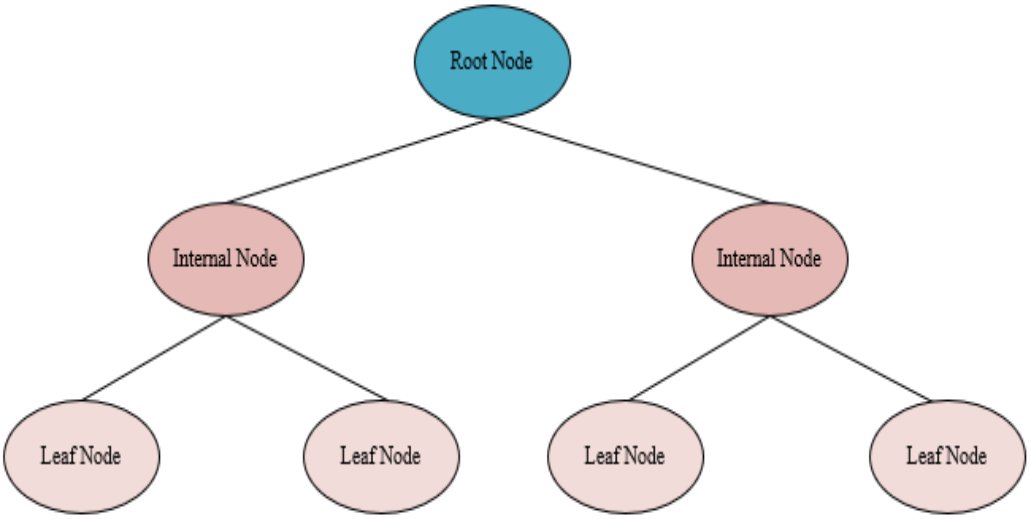


Figure 3.4.7 The basic structure of the Decision Tree

Random forest is a classifier that takes use of multiple decision trees to train and predict samples, the output is determined by using mode of the output category of the individual tree. Random forest has the advantages with the high accuracy, it is not easy to overfit, Random forest algorithm is able to handle with both discrete and continuous data.

Bootstrap aggregating algorithm (Bagging algorithm) construct multiple weak learners separately. The multiple weak learners are in parallel relationship with each other that can be trained simultaneously. Multiple weak learners are combined together. The Bagging algorithm is combined with other classification and regression algorithms to improve its accuracy and stability, while reducing the variance of the results to avoid overfitting.

Naïve Bayes method is based on Bayes algorithm, which assumes that given the target value, the attributes are mutually independent on each other. That is to say, no attribute variable will occupy a larger proportion in decision making, and also no attribute variable will occupy a smaller proportion in decision making.

In human behavior recognition, most of traditional machine learning methods are based on feature extraction techniques, most of feature extraction techniques are based on spatial information, which may be affected by external environments. As an effective machine learning method, iDT has accomplished contributions in the field of human behavior recognition. In deep learning, most of the research work explicates that both spatial and temporal information is vital to motion features. Thus, LSTM is taken into account in our research in order to extract the temporal information for each video frames. Figure 3.4.8 shows the basic LSTM architecture for our study.

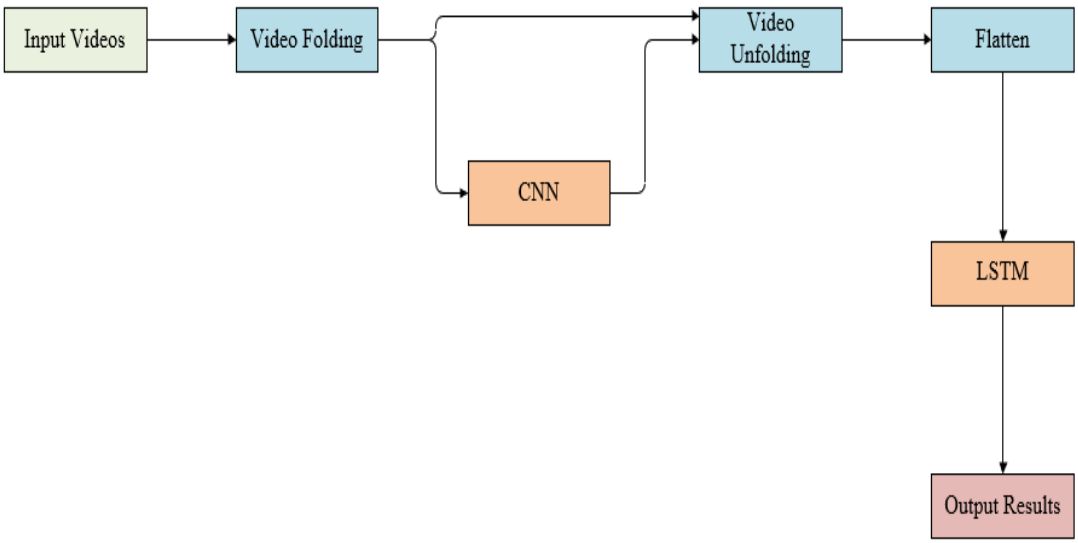


Figure 3.4.8 The basic LSTM architecture for human behavior recognition

In Figure 3.4.7, we convert a video to the sequence of feature vectors so as to accurately present the features from each video frame. Consequently, LSTM net is applied to predict human behaviors. After combined CNN and LSTM, the network achieves extremely high accuracy of human behavior recognition. The reason is that our convolutional operator is able to deal with each video frame independently. The model has the capability to restore the structure of each sequence and reshape the output to a vector sequence. CNNs encompass the feature extractor, the output feature maps are generated from activation functions and relevant pooling layers, the feature maps

exported from the CNNs will be imported as the input of the LSTM network. Figure 3.4.9 shows the CNN+LSTM network structure.

Figure 3.4.9 The network structure of CNN+ LSTM

3.5 Evaluation Metrics

To evaluate different models, cross validation was utilized in order to measure the performance of human behavior recognition. The primary purpose of this confusion matrix is to compare the ground truth with the classification results. Furthermore, the confusion matrix will not only provide the classification accuracy, but also show the relevant relationship of results and predicted classes. Table 3.5.1 shows the confusion matrix.

Table 3.5.1. The confusion matrix

	The predicted classes		
The actual classes		**Positive**	**Negative**
	Positive	TP	FN
	Negative	FP	TN

where:

- True Positive (TP): positive samples are correctly predicted as positive samples by using the classifiers.
- True Negative (TN): negative samples are correctly predicted as negative samples by using the classifiers.
- False Positive (FP): negative samples are incorrectly predicted as positive samples by using the classifiers.
- False Negative (FN): positive samples are incorrectly predicted as negative samples by using the classifiers.

Accuracy (ACC) refers to the proportion of the total number, which is correctly predicted, it is the most common evaluation method.

$$\mathrm{ACC} = \frac{(TP+TN)}{(TP+TN+FP+FN)}. \tag{3.5.1}$$

Recall (R) indicates the ratio of the positive samples are correctly classified, and the equation is presented as follow

$$R = \frac{TP}{(TP+FN)}. \tag{3.5.2}$$

Precision (P) means the proportion of actual positive instance in the samples which are classified as positive samples,

$$P = \frac{TP}{(TP+FP)} \tag{3.5.3}$$

Intersection over Union (IoU) is a standard performance measure for the object segmentation problem. IoU represents the overlap between the candidate bounding box and the ground truth bounding box, namely, the ratio of their intersection and union. The closer the correlation, the greater the value. The best situation is both candidate bounding box and the ground truth bounding box are completely overlapped, that means, the IoU ratio is one. Figure 3.5.1 shows the basic idea of IoU.

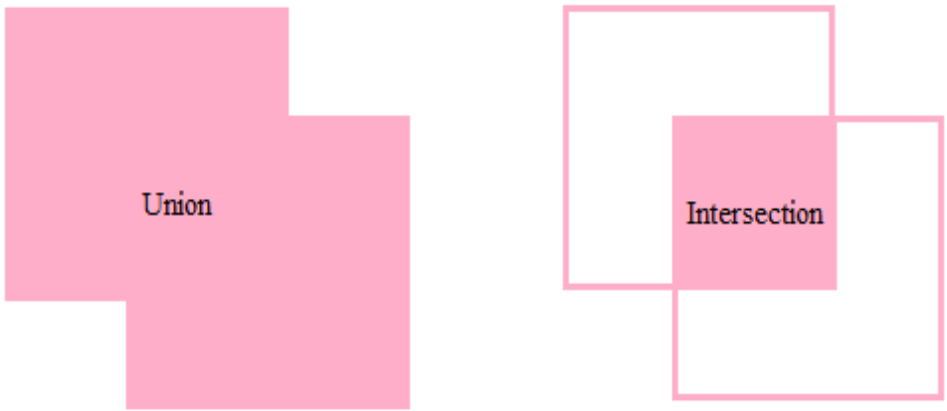


Figure 3.5.1 The basic idea of IoU

Given a set of images, the IoU measurement gives the similarity between the predicted area of the objects presented in the set of images and the ground truth area, which is defined by eq.(3.5.4)

$$IoU = \frac{TP}{(TP+FN+FP)} \quad or \quad \frac{\text{area of overlap}}{\text{area of union}}. \tag{3.5.4}$$

Mean average precision (*mAP*) for a set of classes is the mean of average precision, where N denotes as the number of classes and C means as the class,

$$mAP = \frac{\sum AveragePrecision_C}{N(Classes)}. \tag{3.5.5}$$

To verify the time cost of our proposed methods, the frame per second (FPS) was calculated at the backstage. FPS calculation is presented as follow:

$$\text{FPS} = \frac{frameNum}{elapsedTime} \tag{3.5.6}$$

Chapter 4
Results

The main content of this chapter is to introduce the schema of method and implementation of human behavior recognition based on deep learning. The experimental results and outcomes will be detailed under the support of tables and figures. Moreover, the limitations of this thesis will be pointed out at the end of this chapter. Based on the results, the discussions and analysis will be summarized in the next chapter.

4.1 Experimental Results

Our focus of this thesis is chiefly on the proposed deep learning methods and how they affect our outcomes. Two public datasets were selected, we also create our own datasets for this research work. We adopted four models for our experiments which include YOLOv3, YOLOv2, ResNet, and DenseNet to gain the results of human behavior recognition. Our focus was on these methods how it affects the outcomes. Moreover, an attention mechanism based on the deep neural networks (ResNeXt and SKNet) were also investigated in this research project. Figure 4.1.1 and Figure 4.1.2 show the results on the video frames utilizing two datasets. Figure 4.1.3 and Figure 4.1.4 show the results based on our own dataset I & II.

Figure 4.1.1 The results of video frames for the Weizmann dataset

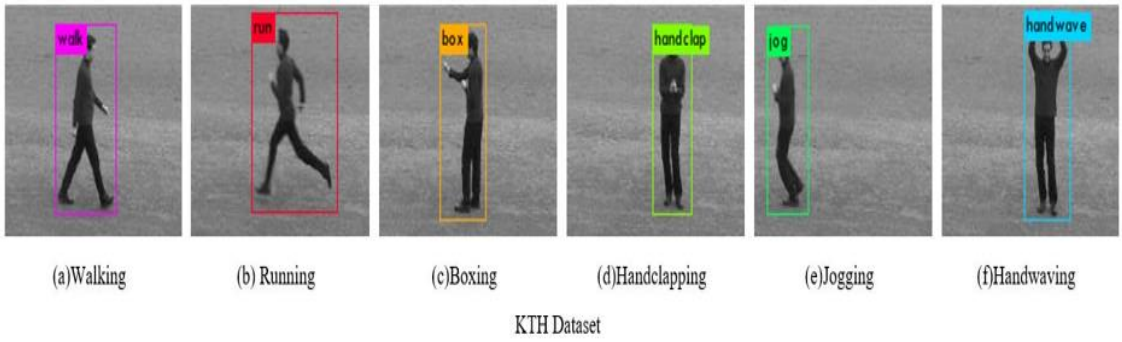


Figure 4.1.2 The results of video frames for the KTH dataset

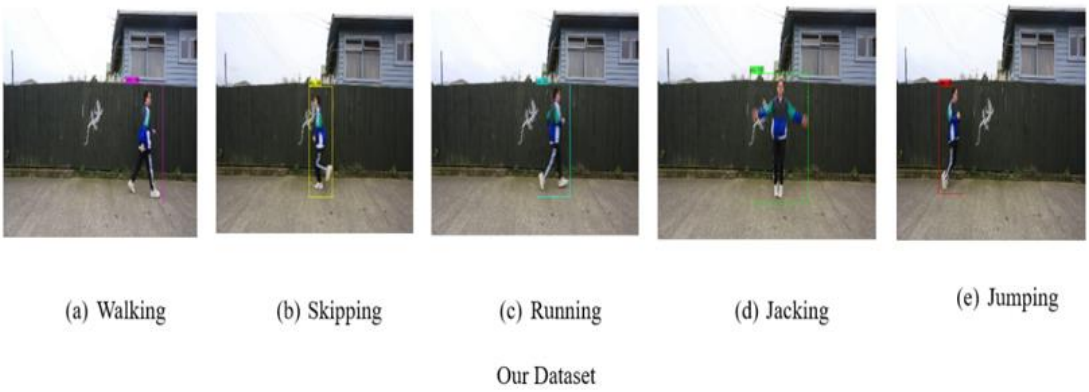


Figure 4.1.3 The results of video frames for our own dataset I

Figure 4.1.4 The results of video frames for our own dataset II

4.2 Weizmann Dataset

Four deep learning models were adopted in this project, by adopting these models, accuracy is achieved up to 90%. YOLOv3 shows the highest accuracy 96.29% and DenseNet has the lowest accuracy 92.62%. Figure 4.2.1 shows the training loss of each proposed models, YOLOv2 achieves the lowest training loss 0.96%, YOLOv3 has the highest training loss 1.81%.

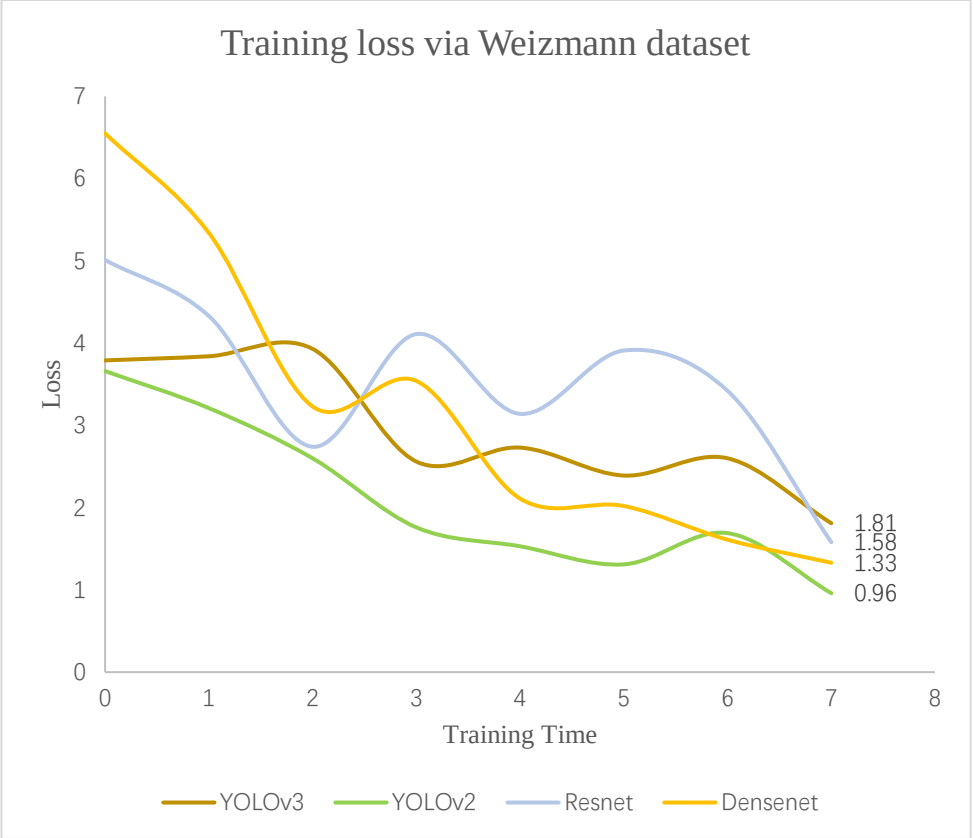
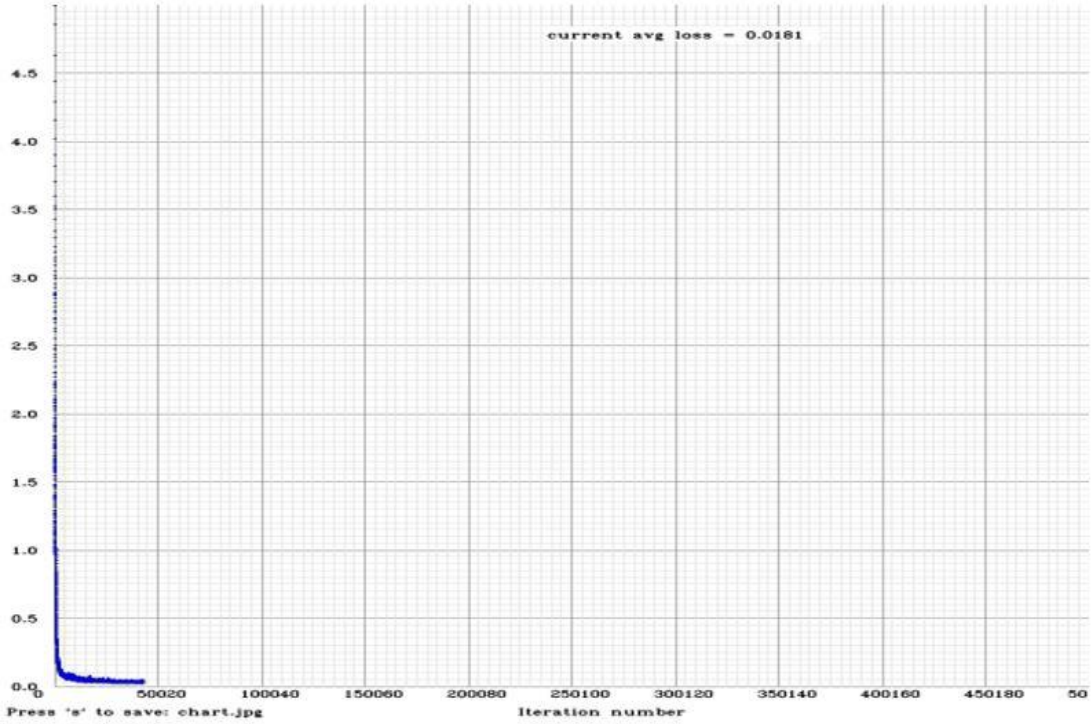


Figure 4.2.1 The training loss via Weizmann dataset by adopting proposed models

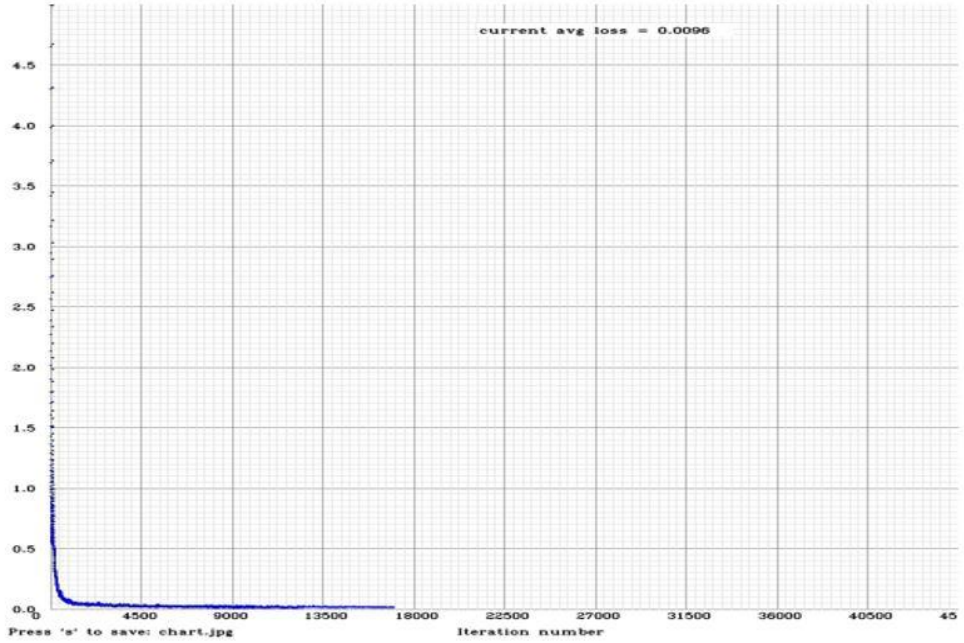
Pertaining to Weizmann dataset, YOLOv3 gains the highest accuracy 96.29%, the overall accuracy of ResNet gets up to 91.40%. Figure 4.2.2 shows the result of the trained YOLOv3, the training loss is around 0.0181 after computed 44,200 epochs, where the x-axis represents the number of iterations during the training process, y-axis indicates the training loss for each iteration. The test results of using YOLOv3 have 0.946 mAP. Jacking has the highest precision 100% and the test FPS is up to 22.7. Moreover, the total accuracy gains 96.29%. In this experiment, by adopting YOLOv3 in Weizmann dataset, Jacking has the highest accuracy (100%), Jumping has the accuracy 91.74%, which is the lowest accuracy compared with other classes.



YOLOv3	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9916	0.9116	0.9066	1	0.9174	0.946 (mAP)
Accuracy	0.9674	0.9571	0.9327	1	0.9174	0.9629 (total accuracy)

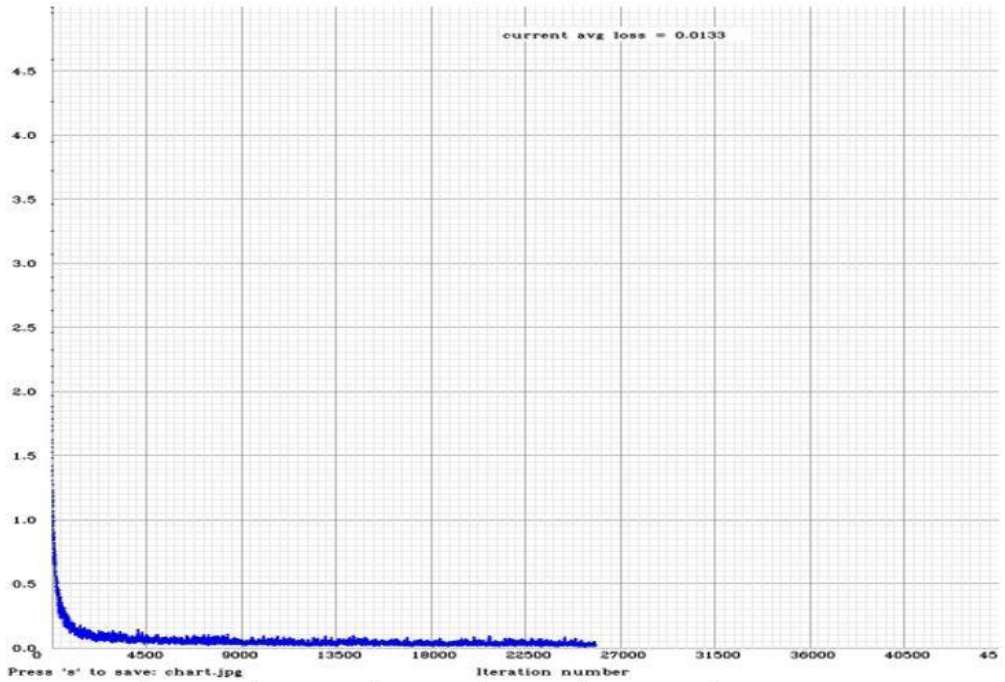
Figure 4.2.2 The training results by using YOLOv3 and the Weizmann dataset

By adopting the YOLOv2 in Weizmann dataset, the overall accuracy is up to 92.71%. In this experiment, Jacking individually reaches the highest accuracy (100%), Jumping has the lowest accuracy 72.79%. Figure 4.2.3 shows the results of training processing, the training loss achieves 0.0096 after 17,000 epochs, where x-axis shows the iteration steps during the training, and y-axis represents the training loss of each iteration step. The test results by using YOLOv2 have 0.911 mAP; Jacking has the highest recognition rate 100%, the running behavior only achieves the lowest recognition rate 65.33%; the experiment carried out at the frame rate 31.3 per second.



YOLOv2	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9878	0.9699	0.6533	1	0.9428	0.911 (mAP)
Accuracy	0.9680	0.9149	0.9423	0.9923	0.7279	0.9271(total accuracy)

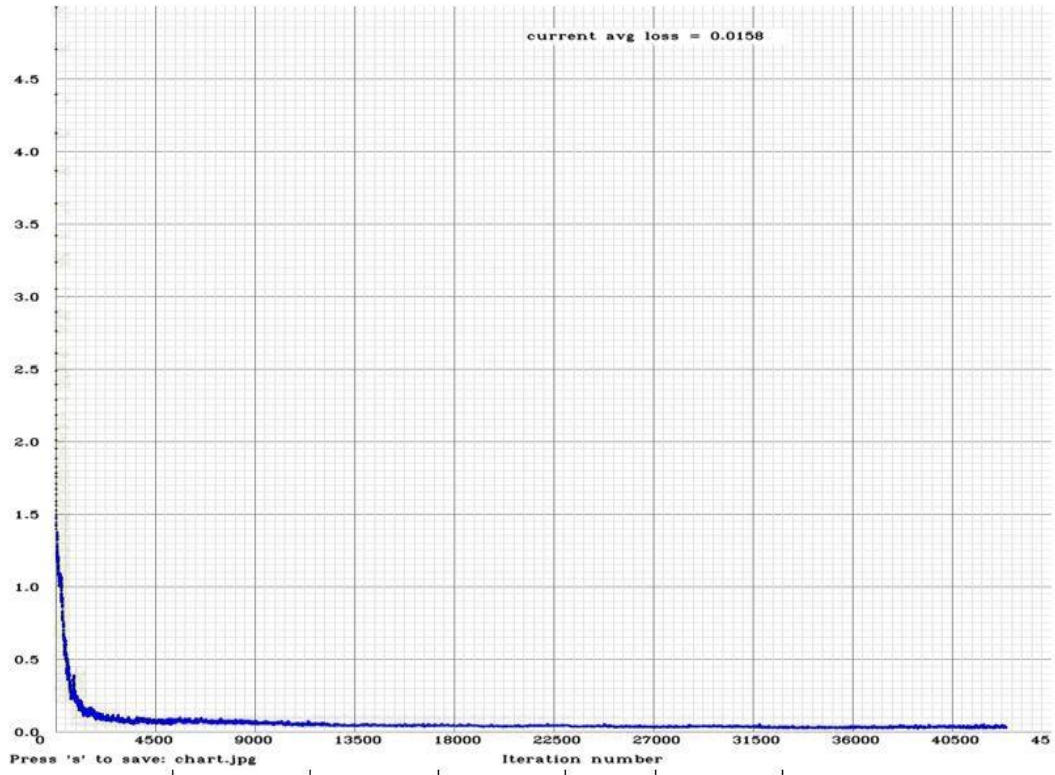
Figure 4.2.3 The Weizmann training result by using YOLOv2



Densenet	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9758	0.9552	0.9083	1	0.9250	0.953 (mAP)
Accuracy	0.9798	0.9275	0.9167	1	0.6991	0.9262 (total accuracy)

Figure 4.2.4 The Weizmann training results by using DenseNet

DenseNet was also utilized in the experiments with Weizmann dataset, the total accuracy of human behavior recognition is 92.62%, Jacking has the highest accuracy of 100%, and Jumping achieved the lowest accuracy (69.91%) compared with other classes. Figure 4.2.4 shows the result of trained DenseNet, the training loss achieved 0.0133 after 24,900 epochs. The test result by adopting DenseNet is 0.953 (mAP), Jacking has the highest precision 100%, Running behavior only achieves the lowest precision 90.83%, the FPS is up to 30.3 frames per second.

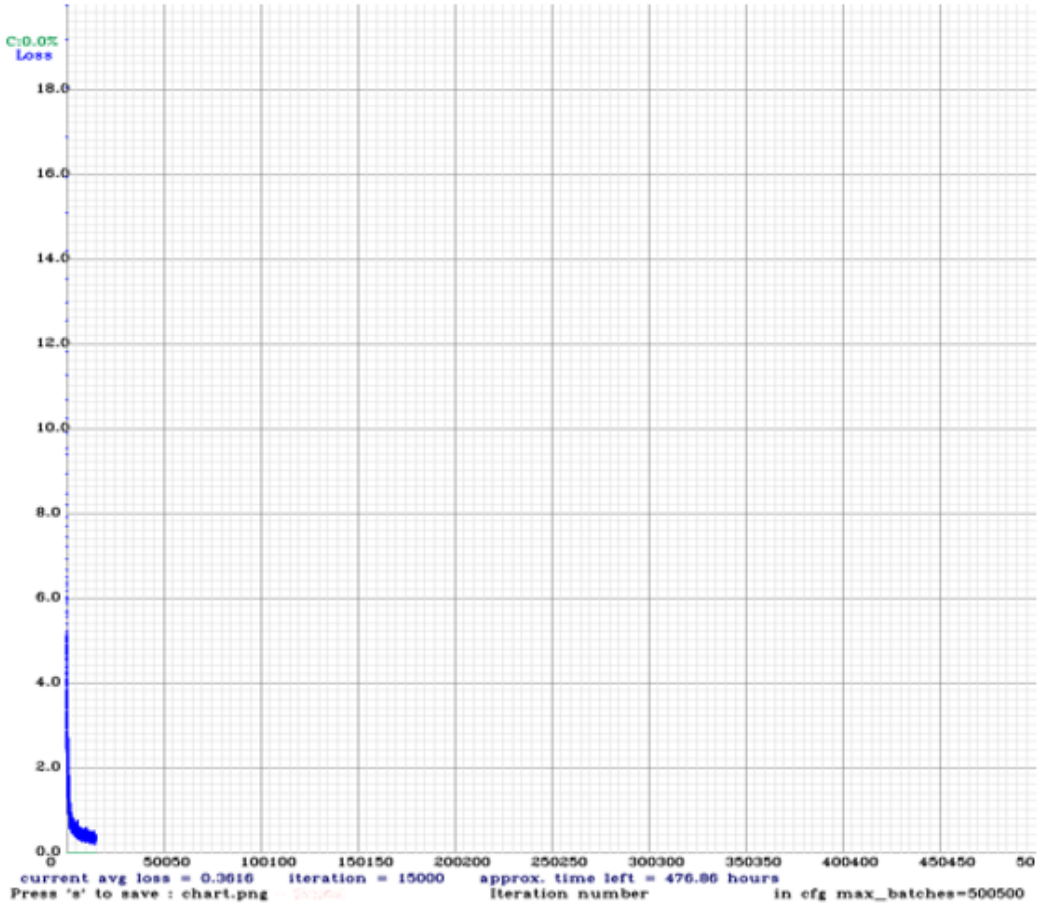


Resnet	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9313	0.8377	0.8205	1	0.8508	0.888 (mAP)
Accuracy	0.9879	0.9348	0.9057	1	0.5229	0.9140 (total accuracy)

Figure 4.2.5 The Weizmann training result by using ResNet

The experiment using Weizmann dataset and ResNet has achieved the accuracy up to 91.40%, where Jacking has the highest accuracy (100%), and Jumping only achieved 52.29% accuracy. Figure 4.2.5 shows the result of trained ResNet, the training loss achieved 0.0158 after 43,100 epochs. The test results of loss by adopting ResNet have the lowest 0.888 (mAP) compared with other three networks. Jacking has the highest

precision (100%) under this experimental condition; while Running behavior only has the precision 82.05%. The frame rate of human behavior recognition is up to 17.1 FPS.



YOLOv4	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9752	0.9379	0.9466	1	0.9503	0.962 (mAP)
Accuracy	0.9683	0.9544	0.9839	1	0.9614	0.9736 (total accuracy)

Figure 4.2.6 The Weizmann training result by using YOLOv4

In the end, the experiment using YOLOv4 in Weizmann dataset has carried out the accuracy up to 97.36%, Jacking individually reaches 100% accuracy, and Jumping has the lowest accuracy (96.14%) compared with other classes. Figure 4.2.6 shows the result of trained YOLOv4, the training loss attained 0.3616 after 15,000 epochs. The test results of loss by adopting YOLOv4 have the highest 0.962 (mAP). Jacking has the highest precision (100%) under this experimental condition; while Skipping only has the precision 93.79%. The frame rate of human behavior recognition is up to 61.3 FPS.

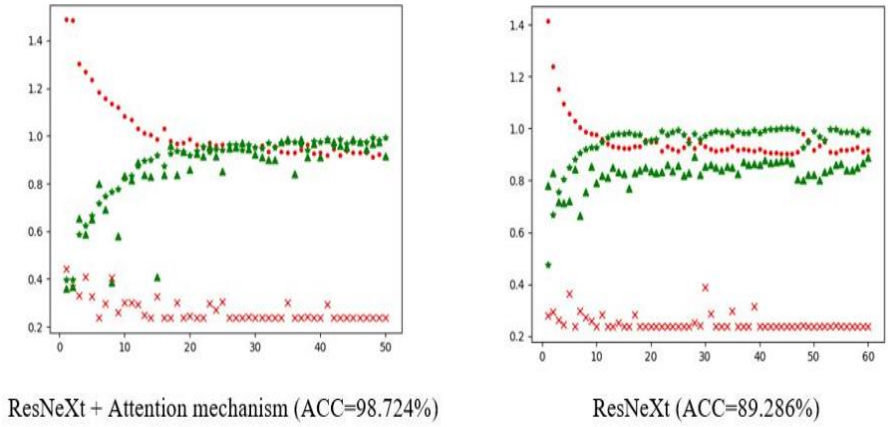


Figure 4.2.7 ResNeXt and Attention mechanism training and validation loss during the training process by using Weizmann dataset

Figure 4.2.7 exhibits the training and validation loss during the training process by using Weizmann dataset with ResNeXt and attention mechanism. The *x*-axis represents the training epoch and *y*-axis stands for the loss. From Figure 4.2.7, the deep learning models are able to achieve 89.286% total accuracy. Moreover, by adopting the attention mechanism into the deep learning models, which not only improve the accuracy without increasing the complexity of the parameters, but also reduce the number of hyperparameters. By adopting the attention mechanism, the results were becoming more robust. By combining the ResNeXt model with attention mechanism together, the accuracy reaches up to 98.724%.

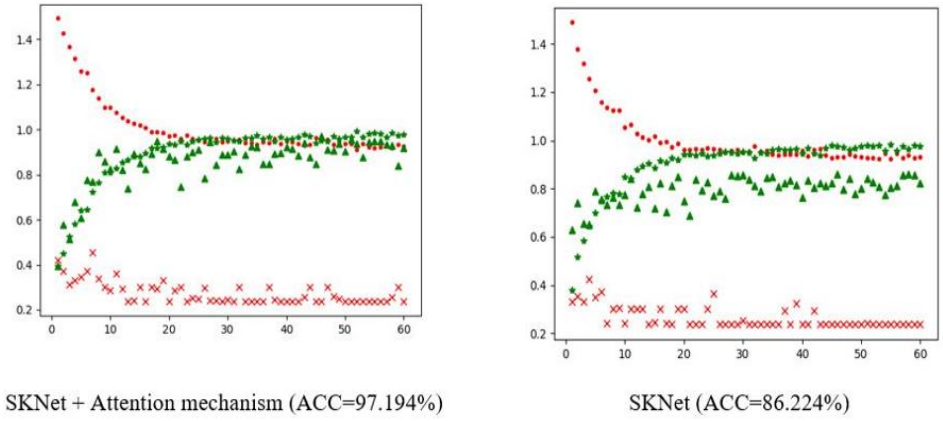


Figure 4.2.8 SKNet and Attention mechanism training and validation loss during the training process by using Weizmann dataset

Figure 4.2.8 shows the training and validation loss during the training process by using Weizmann dataset with the SKNet and attention mechanism. In Figure. 4.2.8, we witness that SKNet models are able to achieve 86.224% accuracy. Moreover, by combining the attention mechanism with the SKNet model, our results were becoming more robust and the accuracy and reach up to 97.194%.

In these experiments, all our experiments necessitate large amount of computations, we chose batch size of 8 and learning rate 0.001. Moreover, the number of the epoch is set to 60. In Figure 4.2.8, the green dotes represent the training and validation accuracy, and the red dotes stand for the training and validation loss.

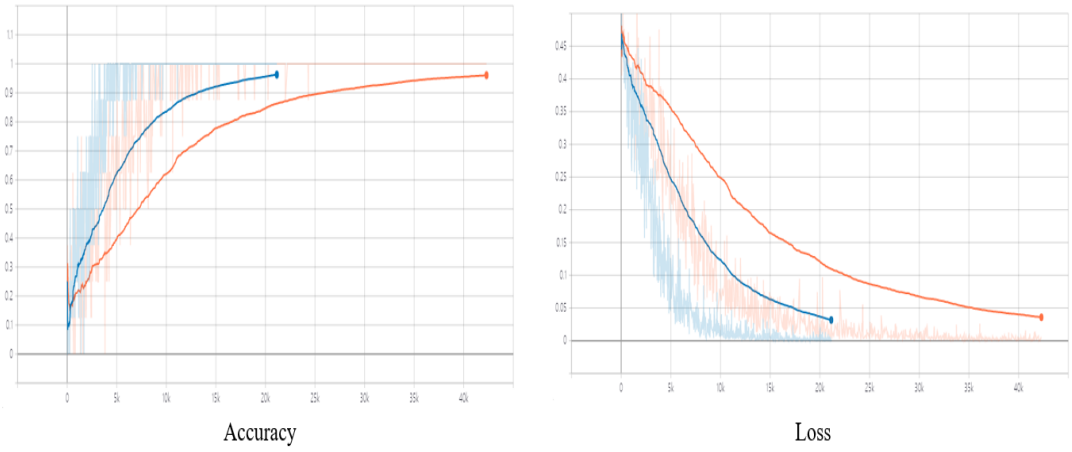


Figure 4.2.9 The training and testing accuracy and loss by using CapsNet on Weizmann dataset

Figure 4.2.9 shows the training/testing accuracy and loss during the training process by using Weizmann dataset with the CapsNet. In Figure. 4.2.9, we witness that CapsNet model is able to achieve 96.86% accuracy. Moreover, the orange line represents the training set; the blue line stands for the testing set; where the *x*-axis represents the training/testing steps, the *y*-axis shows the training/testing outcomes. In our experiment, the number of iterations is set to 10,000, the batch size is 8, and the learning rate is 0.001.

4.3 KTH Dataset

These four deep neural models were utilized in KTH dataset. YOLOv3 achieves the highest mAP at 0.8458; YOLOv2 only having 0.8059 mAP is the lowest one compared to others. Figure 4.3.1 shows the training loss of each proposed models and DenseNet achieves the lowest training loss 1.85%. ResNet has the highest training loss 3.92%.

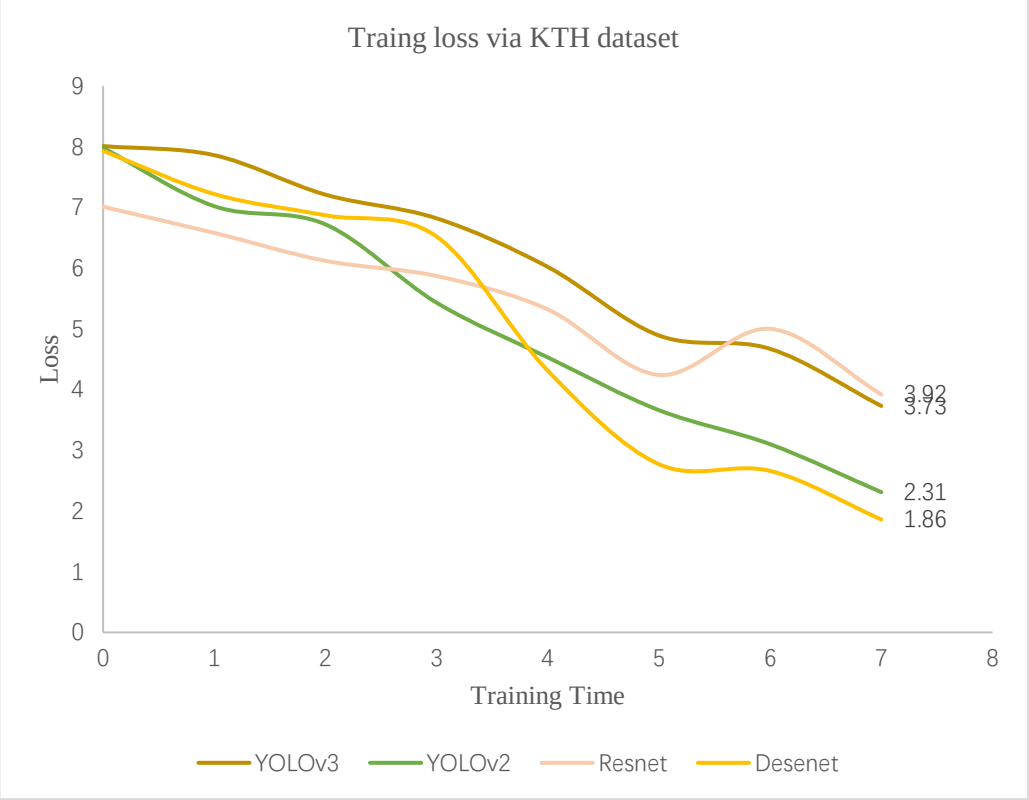
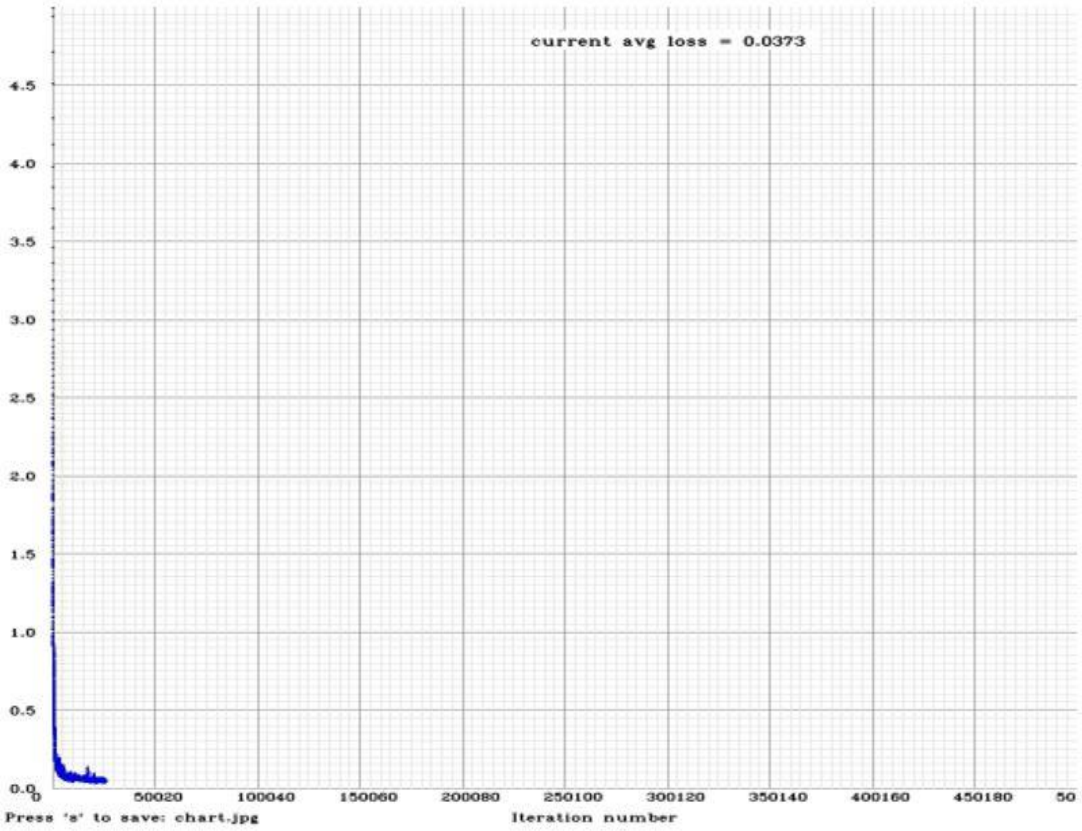


Figure 4.3.1 The training loss via KTH dataset by adopting proposed models

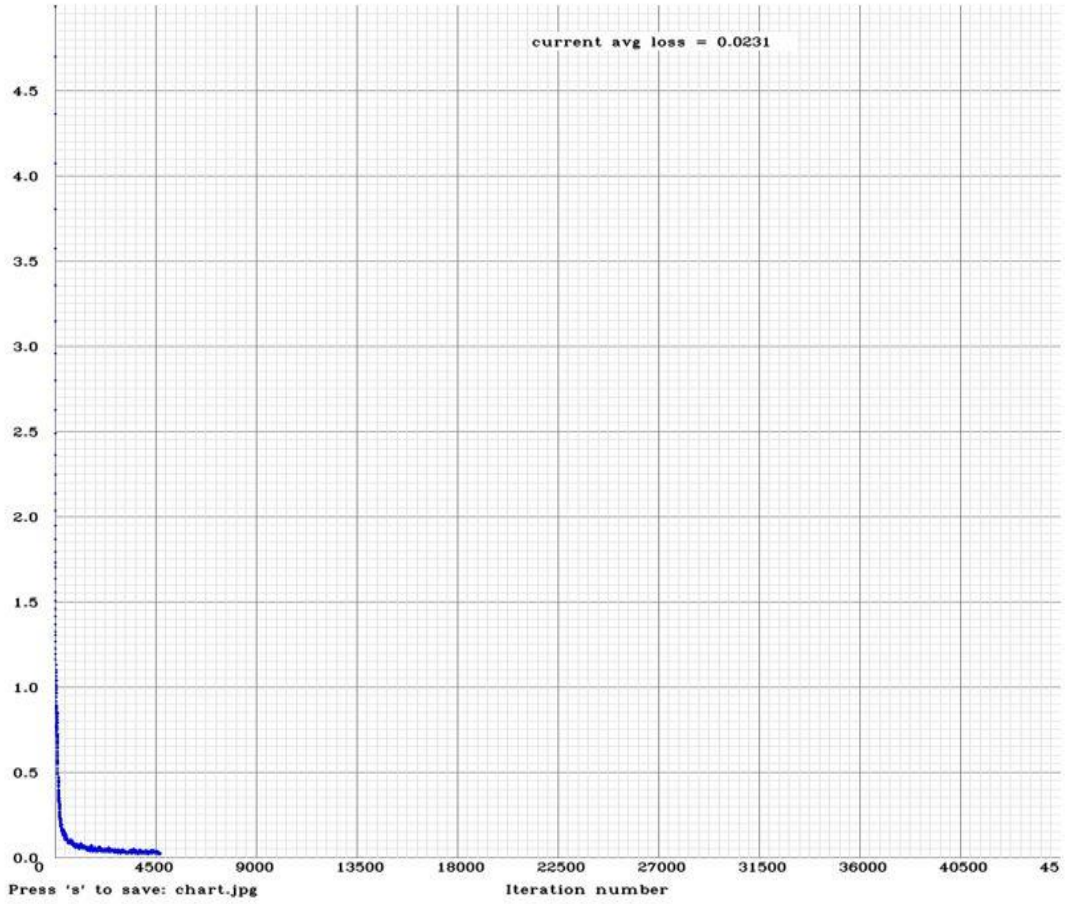
By using YOLOv3 in KTH dataset, it has 84% total accuracy; Walking individually achieved 97.14% accuracy, and Jogging can only achieve 42% accuracy. Figure 4.3.2 shows the result of trained YOLOv3 in KTH dataset. The training loss achieves 0.0373 after 28,800 epochs, where *x*-axis represents the iteration steps and y-axis shows the training loss of each step. By utilizing YOLOv3, the test results reach up to 0.8458 (mAP), the FPS is up to 22.7 frames per second. Boxing has the highest precision rate at 100%, meanwhile, Running has the precision 54% compared to other behaviors.



YOLOv3	Walk	Run	Box	Handclap	Jog	Handwave	Results
Precision	0.9967	0.5400	1	0.8590	0.7323	0.9467	0.8458 (mAP)
Accuracy	0.9714	0.9589	0.9282	0.8851	0.4200	0.8724	0.8400 (total accuracy)

Figure 4.3.2 The KTH training result by using YOLOv3

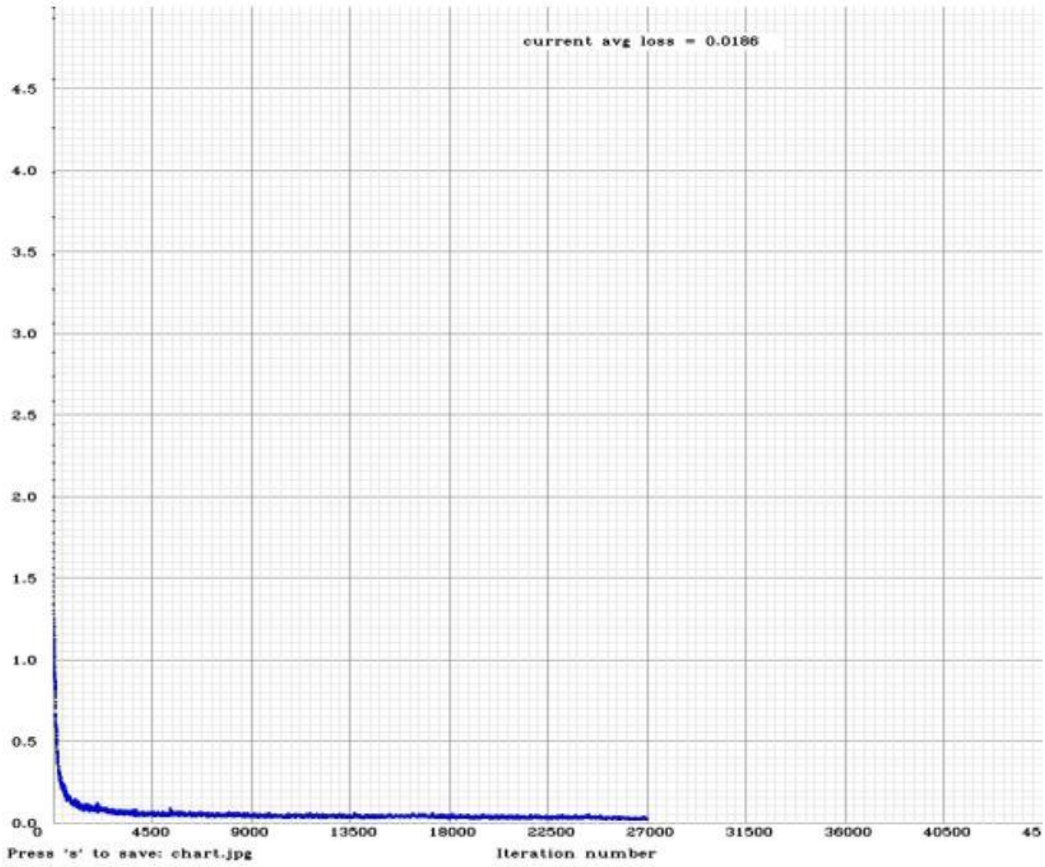
By utilizing YOLOv2 in KTH dataset, the total accuracy is 82.63%. Figure 4.3.3 shows the result of trained YOLOv2 in KTH dataset, the training loss is 0.0231 after 4,700 epochs. The testing result by using YOLOv2 only generated 0.8059 (mAP) which is the lowest one compared to other neural networks, the *x*-axis in Figure 4.3.3 indicates the iteration steps during the training, and *y*-axis shows the training loss for each step. Moreover, the FPS is up to 31.2 frames per second. Boxing has the highest precision (100%), but Running has the lowest one (44.60%) compared to other classes of behaviors. In this experiment, Boxing individually reaches 100% accuracy, and Jogging has the lowest accuracy of 66.39%



YOLOv2	Walk	Run	Box	Handclap	Jog	Handwave	Results
Precision	0.9727	0.4460	1	0.8280	0.6385	0.9500	0. 8059(mAP)
Accuracy	0.9602	0.8855	0.8166	0.7895	0.6639	0.8421	0. 8263(total accuracy)

Figure 4.3.3 The KTH training result by using YOLOv2

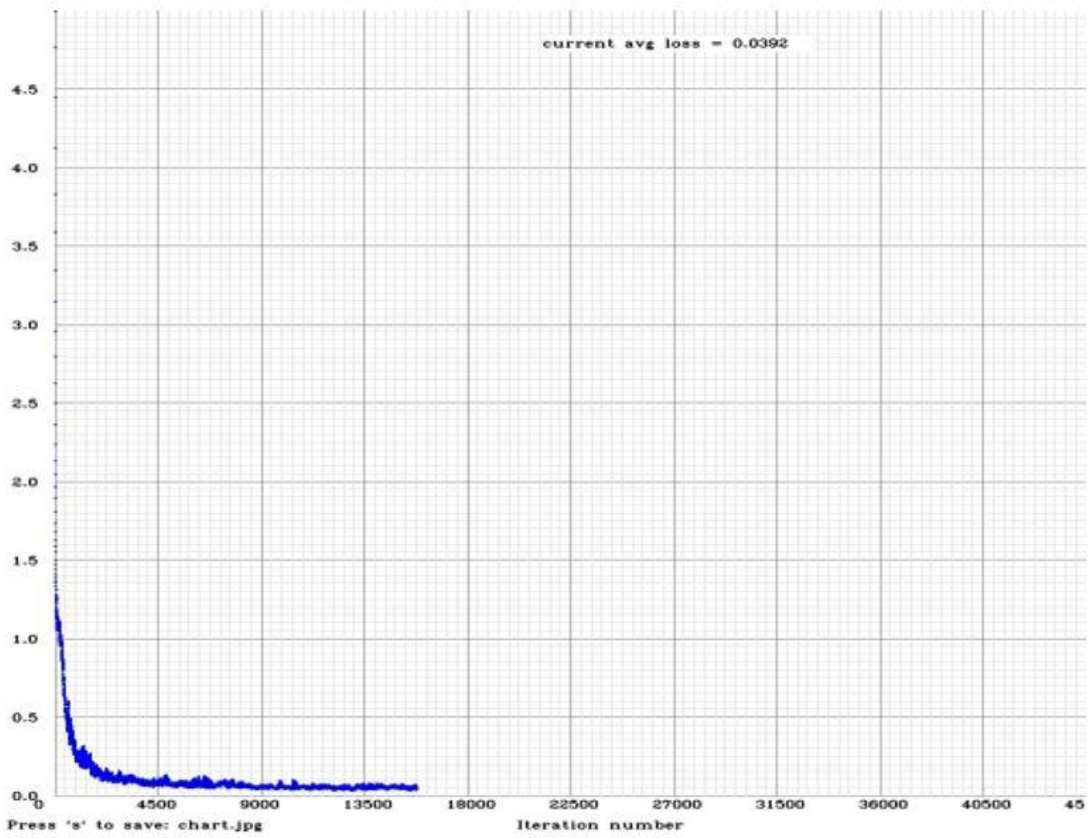
By adopting DenseNet in KTH dataset, it has the accuracy 86.63%, Walking individually reaches the accuracy of 97.96%; Handwaving has the lowest accuracy of 84.77%. Figure 4.3.4 shows the result of trained DenseNet in KTH dataset, the training loss is 0.0186 after 27,100 epochs, *x*-axis shows the iteration number in this experiment, and *y*-axis indicates the training loss. By adopting the DenseNet for this experiment, the result achieved 0.8446 (mAP) and the FPS is up to 32.3. Boxing has the highest precision (100%), while Running has the lowest one (63.46%).



Densenet	Walk	Run	Box	Handclap	Jog	Handwave	Results
Precision	0.9703	0.6346	1	0.8489	0.6220	0.9919	0. 8446(mAP)
Accuracy	0.9796	0.9206	0.8481	0.8887	0.7136	0.8477	0. 8663(total accuracy)

Figure 4.3.4 The KTH training result by using DenseNet

The experiment of KTH dataset by adopting the Resnet has the total accuracy 85.45%, Walking has the highest accuracy 98.41%, and Jogging only achieved 74.74% accuracy. Figure 4.3.5 shows the result of trained ResNet in KTH dataset, the training loss attains 0.0392 after 15,700 epochs. By adopting the ResNet for this experiment, the testing result achieves 0.8383 (mAP), the FPS is only 16.4. In this experiment, *x*-axis represents the iteration steps during the training, *y*-axis shows the training loss in each iteration step. Boxing has the highest precision 100%, Running has the lowest one (64.38%) compared to other classes.



Resnet	Walk	Run	Box	Handclap	Jog	Handwave	Results
Precision	0.9773	0.6438	1	0.8711	0.6628	0.8748	0. 8383(mAP)
Accuracy	0.9841	0.8674	0.8511	0.8375	0.7474	0.8393	0. 8545(total accuracy)

Figure 4.3.5 The KTH training result by using Resnet

The experiment of KTH dataset by adopting the CapsNet has the total accuracy 97.67%. Figure 4.3.6 shows the training/validation accuracy and loss by adopting KTH dataset, the training loss attains 0.0192 after 10,750 epochs. By adopting the CapsNet for this experiment, Boxing has the highest accuracy 98.21%, Running has the lowest one (95.28%) compared to other classes. In this experiment, we chose the batch size 8 and the learning rate 0.001. Moreover, the orange line represents the training set; the blue line stands for the testing set; where *x*-axis represents the training/testing steps, *y*-axis represents the training/testing values.

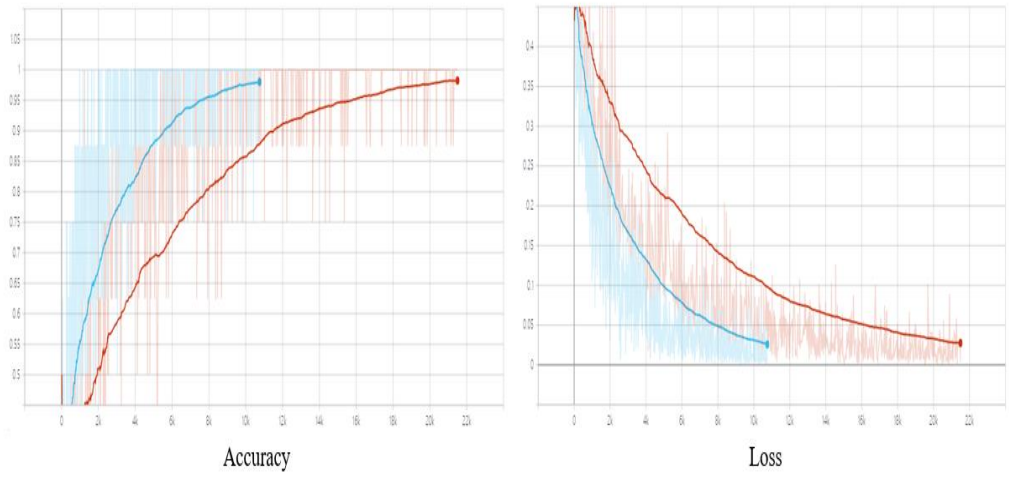
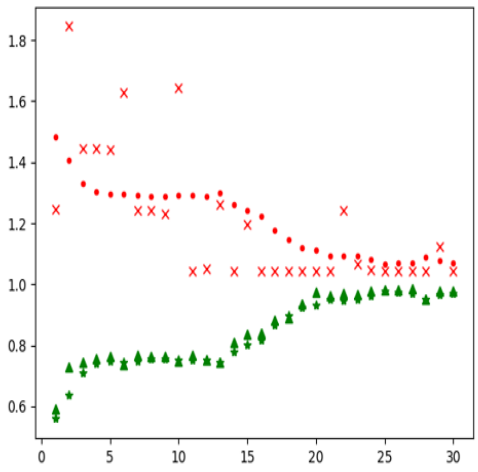


Figure 4.3.6 The training/validation accuracy and loss during the training process by using KTH dataset

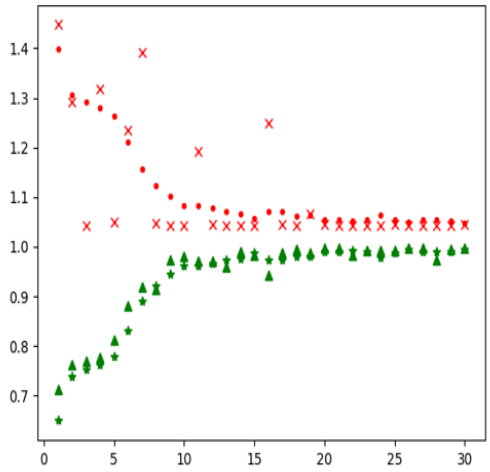
The experiment of KTH dataset by adopting the SKNet has the total accuracy 98.64%. The SKNet after combined with attention mechanism showed the positive result which is able to earn 99.79% accuracy. Compared with the two models, the accuracy grows 1.15%. Figure 4.3.7 shows the SKNet and Attention mechanism training and validation loss during the training process by using KTH dataset, the training loss attains 1.048 after 16,830 epochs. By adopting the SKNet with attention module, Boxing has the highest accuracy 99.89%, Jogging has the lowest one (99.68%) compared to other classes. By adopting the ResNeXt model in KTH dataset, the total accuracy reaches 98.95%.

The ResNeXt after combined with attention mechanism showed the positive result which is able to earn 99.86% accuracy. Compared with the two models, the accuracy grows 0.91%. Figure 4.3.8 shows the ResNeXt and Attention mechanism training and validation loss during the training process by using KTH dataset, the training loss attains 1.049 after 16,830 epochs. By adopting the ResNeXt with attention module, Boxing has the highest accuracy 100%, Running has the lowest one (99.06%) compared to other classes. In these experiments, we chose the batch size 8 and the learning rate 0.001. Moreover, the number of the epoch is set to 30. In these experiments, we chose the batch size 8 and the learning rate 0.001. Moreover, the number of the epoch is set to 30. In Figure 4.3.7 and Figure 4.3.8, the green dots represent the training and validation

accuracy, the red dots stand for the training and validation loss, and *x*-axis denotes the number of epochs, *y*-axis represents the accuracy/loss values.

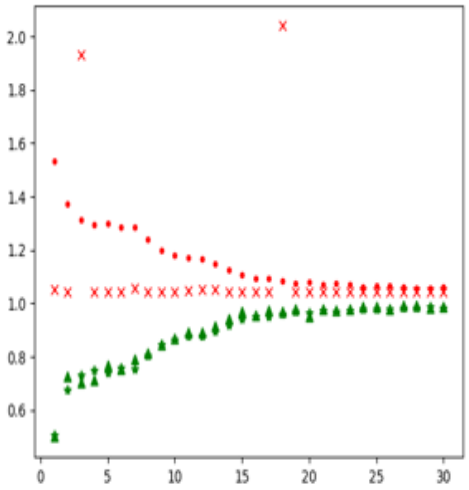


SKNet

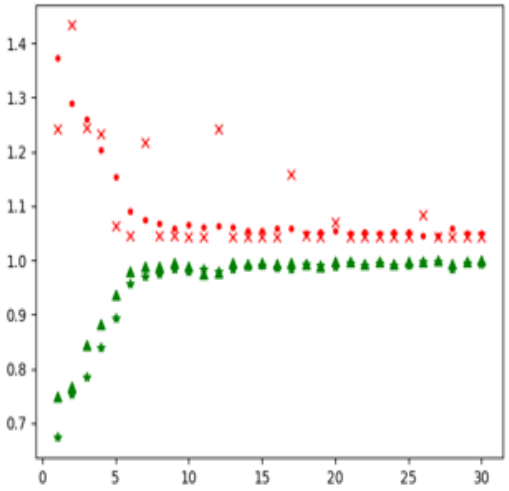


SKNet + Attention

Figure 4.3.7 SKNet and attention mechanism training and validation loss during the training process by using KTH dataset



ResNeXt



ResNeXt + Attention

Figure 4.3.8 ResNeXt and attention mechanism training and validation loss during the training process by using KTH dataset

4.4 Our Own Dataset

From the experiments, ResNeXt achieved the total accuracy 98.697% on our own dataset I, the ResNeXt with attention mechanism is able to gain 99.846% total accuracy which grows 1.149% compared with the ResNeXt model. Figure 4.4.1 shows the training and validation loss during the training process by adopting our own dataset I with state-of-the-art deep learning methods.

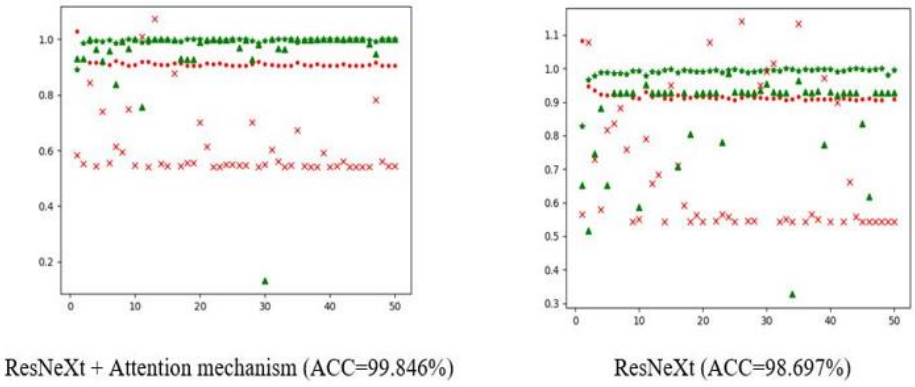


Figure 4.4.1 ResNeXt and attention mechanism training and validation loss during the training process by using our own dataset I

The SKNet model individually achieved 96.982% accuracy with the assistance of our own dataset I; the SKNet after combined with attention mechanism shows the positive result which is able to earn 98.644% accuracy. Compared with the two models, the accuracy grows 1.662%.

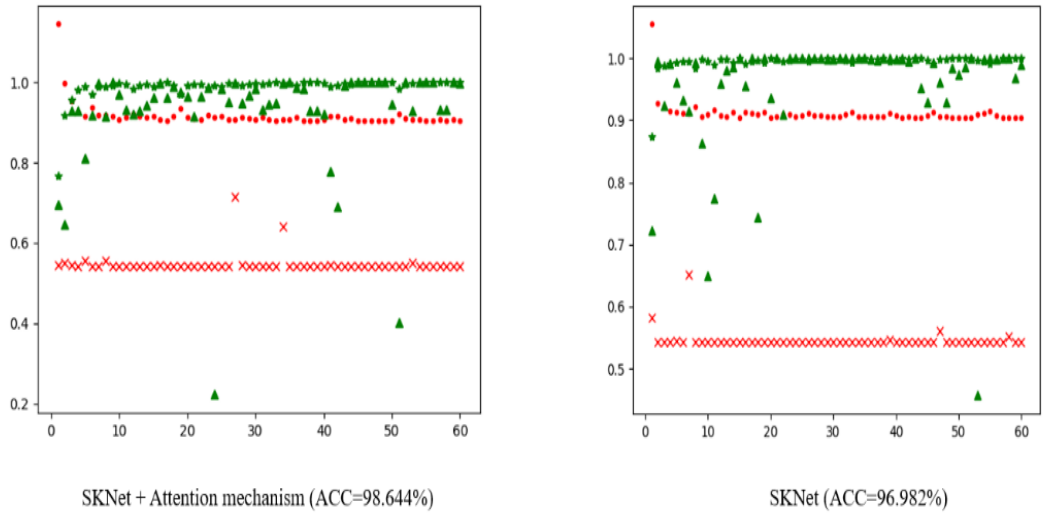
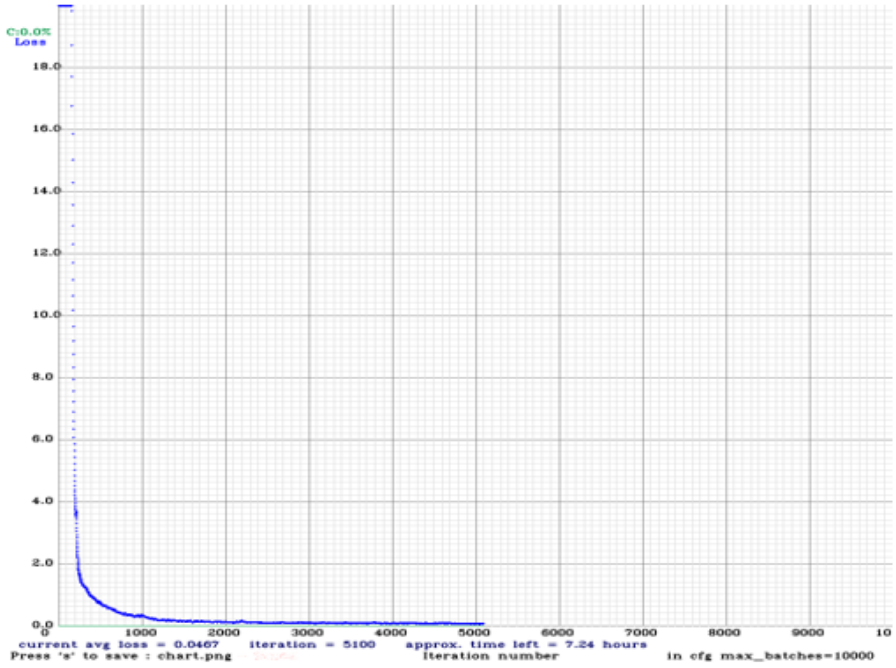


Figure 4.4.2 SKNet and attention mechanism training and validation loss during the training process by using our own dataset I

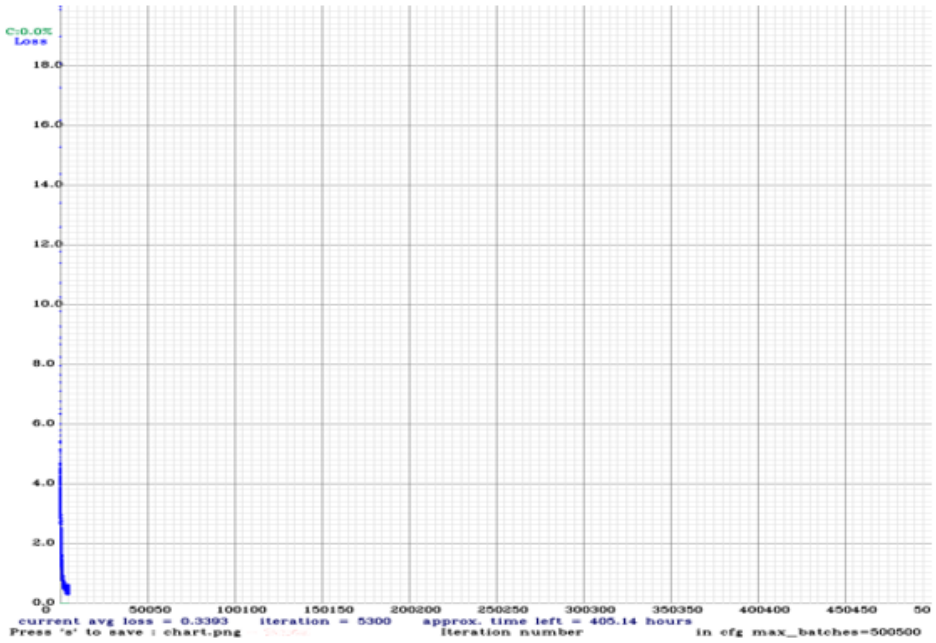
Figure 4.4.2 shows the training and validation loss during the training process by adopting our own dataset I with SKNet and SKNet with an attention mechanism. For better comparing the models based on various datasets, we selected Weizmann dataset and our own dataset I. For both datasets, the number of classes is the same. The both datasets subsume the same static video frames. In our experiments, the number of epochs is 60, batch size is 8, and the learning rate is 0.001.

The experiment of our dataset I by adopting the YOLOv3 has the total accuracy 96.37%, Jacking has the highest accuracy (100%), and Jumping only achieved 95.33% accuracy in this experiment. Figure 4.4.3 shows the result of trained YOLOv3 in our dataset, the training loss attains 0.0467 after 5,100 epochs. For this experiment, the test result achieves 0.952 (mAP), and the FPS is up to 34.1 frames per second. Jacking has the highest precision 100%, Skipping has the lowest one (91.61%) compared to other classes.



YOLOv3	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9566	0.9161	0.9659	1	0.9214	0.952 (mAP)
Accuracy	0.9655	0.9215	0.9782	1	0.9533	0.9637 (total accuracy)

Figure 4.4.3 Our dataset I training result by using YOLOv3



YOLOv4	Walk	Skip	Run	Jack	Jump	Results
Precision	0.9796	0.9537	0.9648	1	0.9769	0.975 (mAP)
Accuracy	0.9753	0.9569	0.9779	1	0.9579	0.9736 (total accuracy)

Figure 4.4.4 Our dataset I training result by using YOLOv4

In this research, YOLOv4 in our dataset has the accuracy 97.36%, Jacking individually reaches 100% accuracy, and Skipping has the lowest accuracy (95.69%). Figure 4.4.4 shows the result of trained YOLOv4 in our dataset, the training loss is 0.3393 after 5,300 epochs. By adopting the YOLOv4 for this experiment, the result achieved 0.975 (mAP) and the FPS is up to 62.3. Jacking has the highest precision (100%), while Skipping has the lowest one (96.48%).

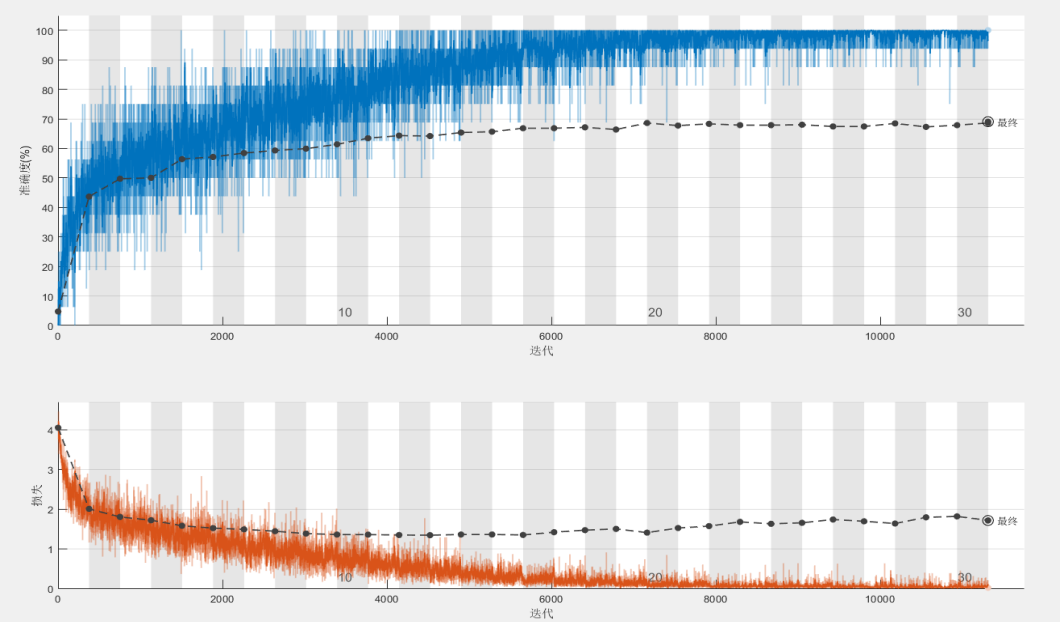


Figure 4.4.5 CNN+LSTM training and validation losses during the training process by using our own dataset I

Figure 4.4.5 shows the training and validation losses based on our own dataset I by using the model CNN+LSTM. Regarding CNN+LSTM, it got 98.53% accuracy based on our own dataset. In this experiment, the proportion between training and validation sets was set to 90:10. The number of epochs is assigned as 30 with 11,310 iterations, the batch size is 16, the learning rate is 0.0001. In Figure 4.4.5, *x*-axis represents the training iteration and *y*-axis stands for the accuracy; the blue dotes represent the training accuracy in each iteration, and black dotes show the validation accuracy in the top plot. For the bottom plot, *x*-axis represents the training iteration and *y*-axis stands for the loss; the orange dotes and black dotes refer to the training and validation loss.

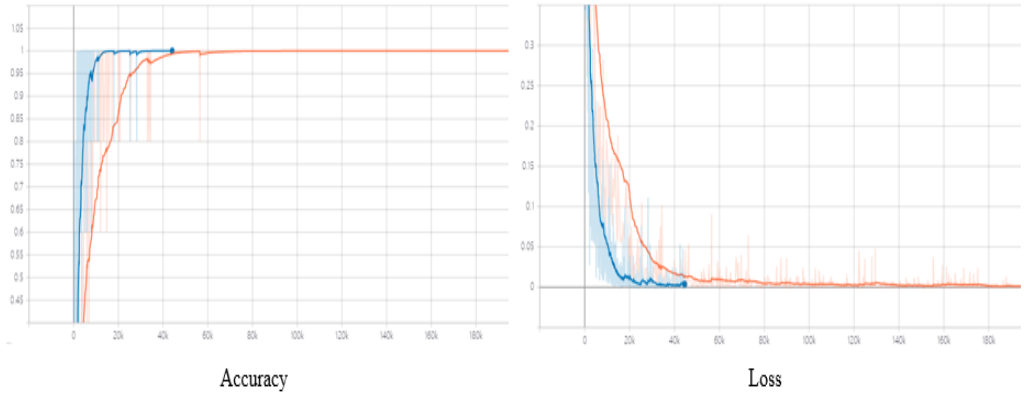


Figure 4.4.6 The training/testing accuracy and loss during the training process by using CapsNet on our own dataset I

Figure 4.4.6 exhibits the training/testing accuracy and loss during the training process by adopting CapsNet on our dataset I. From Figure 4.4.6, CapsNet individually achieved 97.52% accuracy on our dataset I. In this experiment, we chose the batch size 8 and the learning rate 0.001. Moreover, the orange line represents the training set; the blue line stands for the testing set; where the x-axis represents the training/testing steps, the y-axis represents the training/testing values.

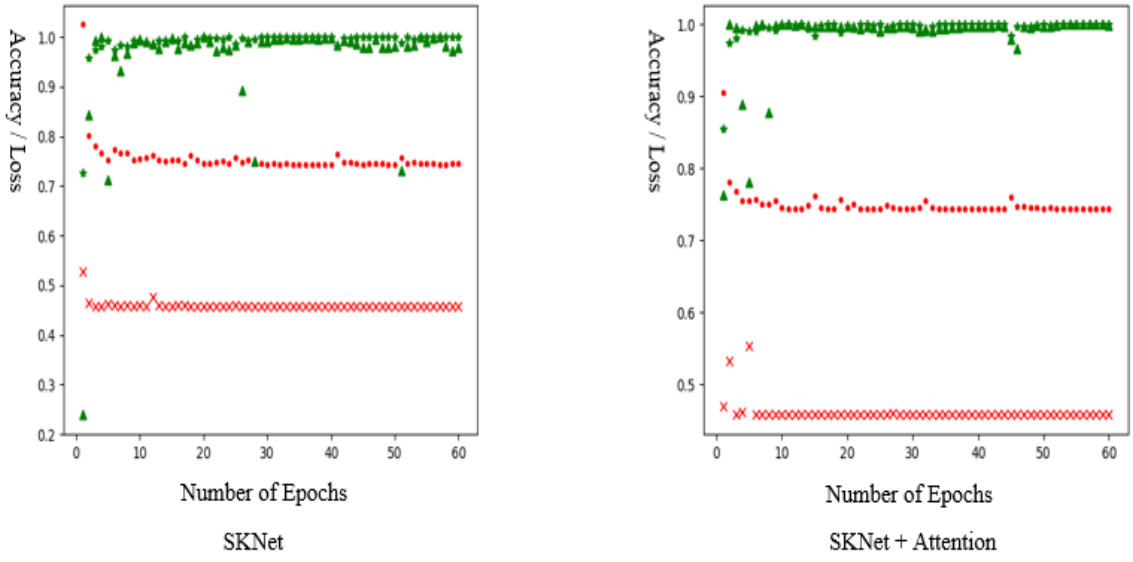


Figure 4.4.7 SKNet and attention mechanism training and validation losses during the training process by using our own dataset II

Figure 4.4.7 shows the training and validation losses by adopting our own dataset II with SKNet and attention mechanism. From Figure 4.4.6, we see that SKNet models are able to achieve 97.95% accuracy. Moreover, by combining the attention model with the SKNet net, the accuracy reaches to 98.88%. Compared with these two models, the accuracy grows 0.93%.

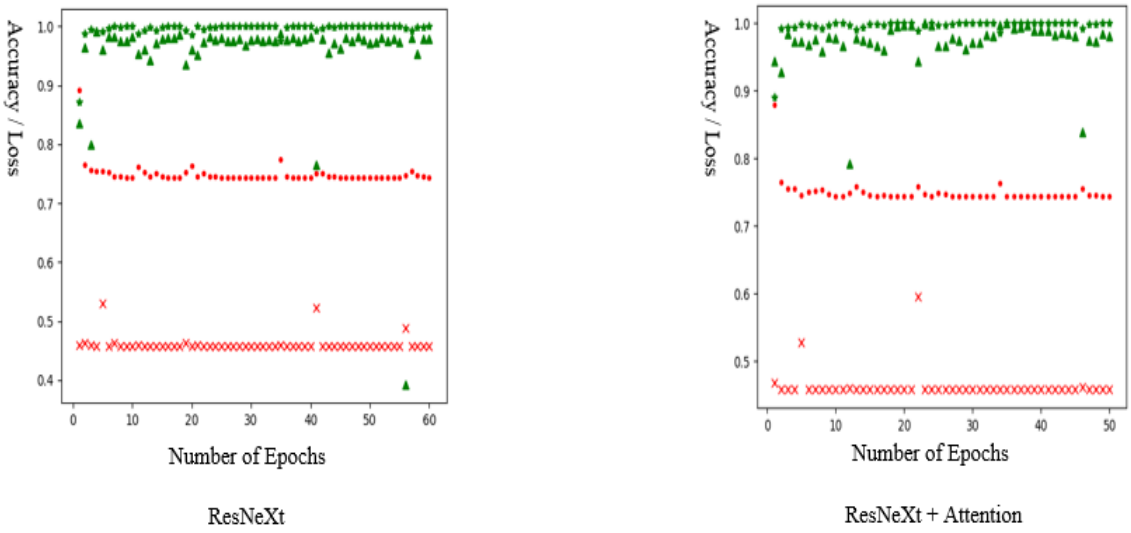


Figure 4.4.8 ResNeXt and attention mechanism training and validation losses during the training process by using our own dataset II

Figure 4.4.8 shows the training and validation losses by adopting our own dataset II with ResNeXt and attention mechanism. In Figure 4.4.7, ResNeXt individually achieved 97.82% accuracy with the assistance of our dataset II; ResNeXt after combined with attention mechanism is able to earn 98.19%. In these experiments, it requires large amount of computations, we chose the batch size 8 and learning rate 0.001. Moreover, the number of the epoch is set to 60. In both Figure 4.4.7 and 4.4.8, the green dots represent the training and validation accuracy, the red dots stand for the training and validation loss, *x*-axis denotes the number of epochs, *y*-axis represents the accuracy/loss values.

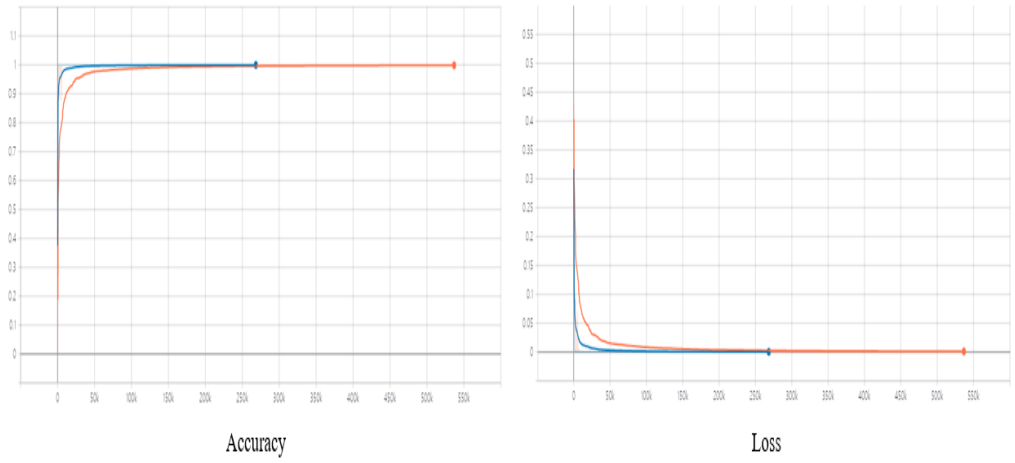


Figure 4.4.9 The training/testing accuracy and loss by using CapsNet on our own dataset II

Figure 4.4.9 shows the training/testing accuracy and loss by using CapsNet on our own dataset II. The CapsNet individually achieved 98.72% accuracy with the assistance of our dataset II. Compared with the previous SKNet method, the accuracy grows 0.77%. Moreover, the orange line represents the training set; the blue line stands for the testing set; where x-axis represents the training/testing steps, y-axis shows the training/testing values. In our experiment, the number of iterations is set to 10,000, the batch size is 8, and the learning rate is 0.001.

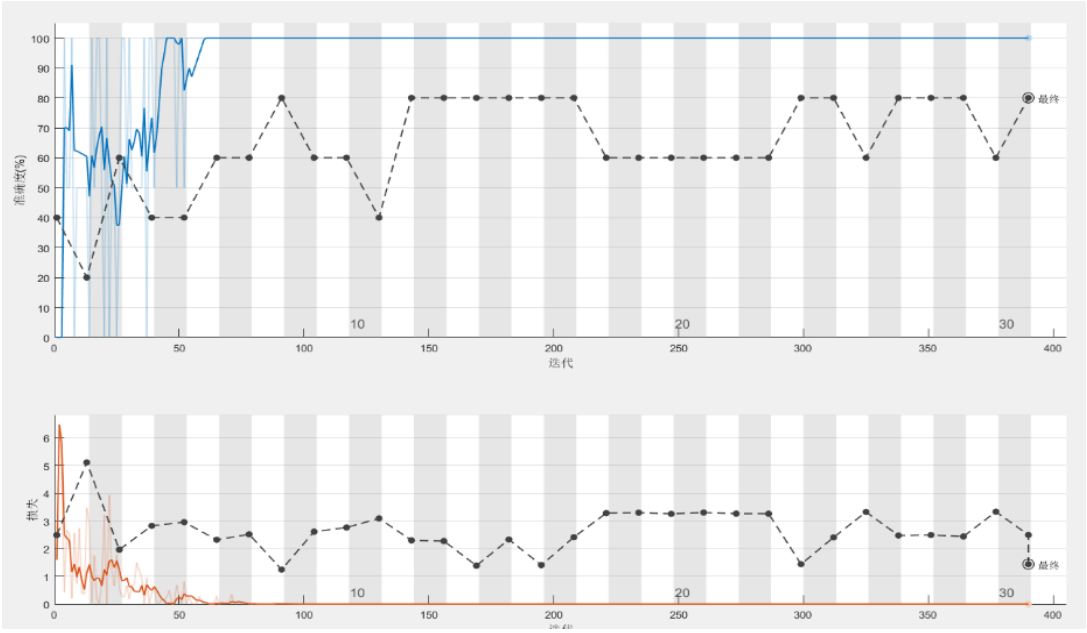


Figure 4.4.10 The training/validation accuracy and loss by using LSTM on our own dataset II

Figure 4.4.10 exhibits the training/validation accuracy and loss by using LSTM on our own dataset II, where the blue line denotes the training accuracy; the orange line shows the training loss; the black dots denotes the training/validation accuracy and loss; x-axis represents the number of iterations and the y-axis stands for the values of the accuracy and loss. The LSTM attained 99.56% accuracy with the assistance of our dataset. In this experiment, due to the small dataset we use, the number of iterations is set to 380, the batch size is 4, and the learning rate is 0.001. We adopted CapsNet + LSTM model with class score fusion.

Chapter 5
Discussions and Analysis

In this chapter, discussion and analysis with respect to the outcomes of the experiments are clearly demonstrated and presented. Moreover, the comparisons between related work and our proposed methods will be discussed in this chapter. The demonstration of human behavior recognition based on the outcomes of deep learning methods will be addressed. Finally, the significance of this thesis will be also identified through analyzing the outcomes. According to the outcomes, the conclusion and future work will be addressed in the next chapter.

5.1 Discussions

Throughout our experiments, we took use of two public datasets and four deep neural networks to compare the outcomes of our deep learning approaches. The previous work was implemented by using these two public datasets for human behavior recognition. From the previous work, visual features were extracted from various video frames, which were employed to represent human behaviors. However, traditional approaches mainly were implemented by using machine learning (ML) approaches so as to reduce the irrelevant or redundant features before classification. Typically, traditional approaches based on ML techniques cannot attain the recognition in real time and the processing is also time consuming.

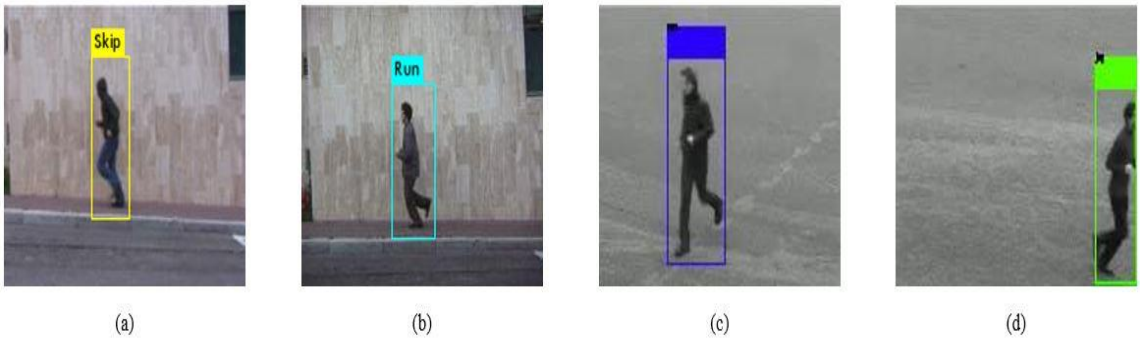


Figure 5.1.1 The examples of incorrect classification, the correct labels are (a) Running in Weizmann dataset, (b) Skipping in Weizmann dataset, (c) Jogging in KTH dataset, and (d) Running in KTH dataset.

Throughout our experiments, DenseNet has the highest accuracy of 0.953 (mAP), YOLOv3 has the highest accuracy of 96.29% by using Weizmann dataset, YOLOv3 has the highest mAP (0.8458) with the KTH dataset. By using Weizmann dataset, Jacking has the highest precision and accuracy amongst all experiments. Moreover, Boxing using KTH dataset has the highest precision within all experiments. However, if using Weizmann dataset, the precision and accuracy of Running and Skipping may not be robust, so do Running and Jogging in KTH dataset. The reason why we obtain the low precision is that we are using video frames as the training dataset, these behaviors may have similarities. Thus, during the tests, the results get lower than the expected. Figure

5.1.1 shows the examples of various behaviors which are incorrectly classified in two adopted datasets.

As we know, deep learning shows its possibility to resolve the problem of human behavior recognition in real time. Table 5.1.1 compares traditional machine learning methods and deep learning methods for human behavior recognition in Weizmann dataset. Regarding YOLOv3, we achieved the highest accuracy 96.29%, the nearest neighbor classifier (NNC) has shown the accuracy 95.6% by using the global spatiotemporal distributions of interest points (Bregonzio, Gong, & Xiang, 2009). Moreover, pertaining to deep learning methods, they achieved the real-time recognition; meanwhile, traditional methods require human behavior recognition frame by frame.

Table 5.1.1 The comparisons of machine learning and deep learning methods using Weizmann dataset

		Walk	**Skip**	**Run**	**Jack**	**Jump**	**mAP**	**Accuracy**	**fps**
Deep Learning Methods	YOLOv3	0.9916	0.9116	0.9066	1	0.9174	0.946	**0.9629**	23.3
	YOLOv2	0.9878	0.9699	0.6533	1	0.9428	0.911	0.9271	32.3
	DenseNet	0.9758	0.9552	0.9083	1	0.9250	**0.953**	0.9262	31.3
	ResNet	0.9313	0.8377	0.8205	1	0.8508	0.888	0.9140	17.0
	Modified YOLOv3	0.94	0.94	0.97	1	0.97	**0.964**	0.953	24.7
Machine Learning Methods	ANN	0.904	0.942	0.923	0.962	0.904	0.927	-	-
	Decision Tree	0.9615	0.8667	0.9808	0.9808	0.9231	**0.942**	-	-
	NNC	1	0.78	1	1	1	-	**0.956**	-
	SVM	1	0.67	1	1	1	-	0.934	-

Table 5.1.2 The comparisons of different methods in KTH dataset

		Walk	Run	Box	Handclap	Jog	Handwave	mAP	Accuracy	FPS
Deep Learning Methods	YOLOv3	0.9967	0.5400	1	0.8590	0.7323	0.9467	**0.8458**	0.8400	22.7
	YOLOv2	0.9727	0.4460	1	0.8280	0.6385	0.9500	0.8059	0. 8263	31.2
	DenseNet	0.9703	0.6346	1	0.8489	0.6220	0.9919	0.8446	**0. 8663**	32.3
	ResNet	0.9773	0.6438	1	0.8711	0.6628	0.8748	0.8383	0. 8545	16.4
	3D-CNN	0.97	0.79	0.9	0.94	0.84	0.97	-	**0.902**	-
Machine Learning Methods	LF+SVM	0.838	0.549	0.979	0.597	0.604	0.736	-	0.717	-
	Linear SVM+LTP	0.90	0.86	0.98	0.95	0.76	0.96	-	0.742	-
	Histograms of spatio-temporal gradients	0.969	0.969	0.938	1	0.781	1	-	**0.942**	-
	SVM+HOG	0.924	0.930	0.941	0.910	0.914	0.930	-	0.9248	-
	AdaBoost + C.45	0.850	0.780	0.960	0.838	0.820	0.690	-	0.823	-

KTH as the most popular and active public dataset has been well investigated. In Table 5.1.2, we compare the traditional machine learning methods and deep learning methods for human behavior recognition using KTH dataset. In our experiments, DenseNet achieved the accuracy 86.63%. 3D-CNNs extracts spatiotemporal features by conducting 3D convolutions which have the highest accuracy (90.2%) (Ji, Xu, Yang, & Yu, 2013). Moreover, Ivan *et al*. proposed a histogram of gradients with a greedy matching method to recognize human behavior which achieved accuracy 94.2%. Because human behaviors have the similar local spatiotemporal events, which lead to misclassifications (Laptev & Lindeberg, 2004). Moreover, human behavior recognition

is implemented with the accuracy 92.48% by adopting SVM with HOG features (Jagadeesh & Patil, 2019).

The results show that both traditional machine learning methods and deep learning methods have succeeded in human behavior recognition. However, through literature review, the traditional machine learning methods may have the best accuracy on human behavior recognition, but it requires preprocessing, segmentation, and feature extraction; it cannot achieve the recognition in real time. 3D-CNN model was proffered which has the highest accuracy (90.2%), the model takes use of not only the spatial information of video frames, but also the temporal information which was also captured by using multiple consecutive frames. The 3D-CNN model implements real-time recognition which shows the performance in the real scenes.

5.2 Contributions

For the past decades, human behavior recognition dominantly took use of traditional machine learning methods. Our contribution of this research work is to implement the modern methods by using end-to-end models, we achieve real-time human behavior recognition without using frame-by-frame way. Moreover, a powerful GPU is configured to accelerate the processing so as to achieve time efficiency. During the experiment, YOLOv3 gained the highest accuracy based on the public dataset. Meanwhile, we modify YOLOv3 (Lu, Yan & Nguyen, 2018) by decreasing the convolutional layers to increase the processing speed which also has 95.3% accuracy.

In these experiments, by harnessing the single deep learning model to recognize the human behavior, YOLOv3 got the highest accuracy 96.29% with Weizmann dataset; DenseNet gained 86.63% accuracy. However, for the single behavior recognition by using KTH dataset was low, the reason is that a number of behaviors may have similarities.

In this research project, we implement four deep learning models with positive results. During the experiment, YOLOv3 reached the highest accuracy based on the public dataset. Meanwhile, we modify YOLOv3 (Lu, Yan & Nguyen, 2018) by

decreasing the convolutional layers so as to increase the speed, it has the accuracy 95.3%. The proposed method aims to recognize human behavior in indoor environment for public security in real time.

Table 5.2.1 The comparisons of deep learning methods and ensemble learning method in Weizmann dataset

Weizmann Dataset		**Walk**	**Skip**	**Run**	**Jack**	**Jump**	**Accuracy**
Deep Learning Methods	YOLOv3	0.9674	0.9571	0.9327	1	0.9174	**0.9629**
	YOLOv2	0.9680	0.9149	0.9423	0.9923	0.7279	0.9271
	DenseNet	0.9798	0.9275	0.9167	1	0.6991	0.9262
	ResNet	0.9879	0.9348	0.9057	1	0.5229	0.9140
	Modified YOLO3	0.9750	0.9330	0.9240	0.9980	0.7160	0.9530
	CapsNet	0.9697	0.9539	0.9695	1	0.9499	0.9686
Ensemble Learning Methods	AdaBoost + Naïve Bayes	1	0.956	0.927	1	0.895	0.9713
	AdaBoost + Random Forest	1	0.978	0.951	1	0.895	**0.9776**
	Bagging + Naïve Bayes	1	0.956	0.902	1	0.842	0.9617
	Bagging + Random Forest	1	0.911	0.951	1	0.978	**0.9776**

To the best of our knowledge, it is the first time we proposed an ensemble learning-based method to recognize human behavior by using Weka Ver.3 to combine these four models together, which gained much better results. In our experiments, we took use of two different ensemble learners to evaluate the results. Table 5.2.1 and Table 5.2.2 show

the comparisons between the deep learning methods and ensemble learning methods based on public datasets. Bagging classifier with random forest got 98.8% accuracy is 12.17% higher than single model recognition. AdaBoost classifier with random forest was able to get 98.71% accuracy with KTH dataset. By combining Bagging learners together with random forest based on Weizmann dataset, the final accuracy has been achieved 97.76%.

In these experiments, various deep learning methods are applied to recognize human behaviors, which achieved promising results. YOLOv3 achieved 84% accuracy in KTH dataset, walking has the highest accuracy of 97.14%; YOLOv2 was able to get 82.63% accuracy; By adopting CapsNet in KTH dataset, the accuracy reaches 97.67%, handwaving has the highest accuracy of 98.86%. The combination of ResNeXt with attention mechanism reaches highest accuracy (99.86%) by using KTH dataset, which has 0.91% growth compared with only adopt ResNeXt model. By combining SKNet with attention mechanism, the recognition accuracy reaches 99.79%, which has 1.15% growth compared with SKNet.

Table 5.2.2 The comparisons of deep learning methods and ensemble learning method in KTH dataset

KTH Dataset		**Walk**	**Run**	**Box**	**Handclap**	**Jog**	**Handwave**	**Accuracy**
Deep Learning Methods	YOLOv3	0.9714	0.9589	0.9282	0.8851	0.4200	0.8724	0.8400
	YOLOv2	0.9602	0.8855	0.8166	0.7895	0.6639	0.8421	0. 8263
	DenseNet	0.9796	0.9206	0.8481	0.8887	0.7136	0.8477	**0. 8663**
	ResNet	0.9841	0.8674	0.8511	0.8375	0.7474	0.8393	0. 8545
	CapsNet	0.9854	0.9628	0.9821	0.9739	0.9674	0.9886	0.9767
	SKNet	0.9887	0.9806	0.9968	0.9869	0.9883	0.9771	0.9864

	SKNet + Attention	0.9974	0.9979	0.9989	0.9979	0.9968	0.9985	0.9979
	ResNeXt	0.9964	0.9878	0.9906	0.9866	0.9899	0.9857	0.9895
	ResNeXt + Attention	0.9987	0.9966	1	0.9985	0.9986	0.9992	0.9986
Ensemble Learning Methods	AdaBoost + Naïve Bayes	0.9980	0.8820	1	0.9920	0.9480	1	0.9834
	AdaBoost + Random Forest	0.9980	0.8850	1	1	0.9700	1	0.9871
	Bagging + Naïve Bayes	0.9980	0.8820	1	0.9920	0.9810	0.9840	0.9861
	Bagging + Random Forest	0.9980	0.8550	1	1	0.9750	1	**0.9880**

Compared with the previous work, the state-of-the-art models in deep learning with attention mechanism show much stable and robust in human behavior recognition. Table 5.2.3 shows the comparison of different deep learning models in human behavior recognition by using a different dataset.

YOLOv3 for human behavior recognition was implemented which achieved 96.29% accuracy which was the highest result in our research study. In Table 5.2.3, the state-of-the-art models in deep learning with attention mechanism have the most positive results for human behavior recognition, ResNeXt with attention mechanism is able to achieve 98.724% total accuracy which is 2.434% higher than previous YOLOv3 model, SKNet with attention mechanism achieved up to 97.194% total accuracy and it has 0.904% growth of the total accuracy compared with the previous YOLOv3 model. YOLOv4 for human behavior recognition reaches the highest result based on Weizmann dataset which has 97.36% accuracy; based on our own dataset, it has up to 97.36% accuracy.

Table 5.2.3 The comparison of different deep learning models in human behavior recognition

Methods		ResNeXt + Attention	ResNeXt	SKNet + Attention	SKNet	ResNet	DenseNet	YOLOv2	YOLOv3	YOLOv4
Weizmann Dataset	Total Accuracy	**98.724 %**	89.286 %	**97.194 %**	86.224 %	91.400 %	92.620 %	92.710 %	96.290 %	97.360 %
	Precision	95.000 %	89.000 %	93.000 %	85.000 %	88.800 %	95.300 %	91.100 %	94.600 %	96.200 %
Our Dataset One	Total Accuracy	**99.846 %**	98.697 %	**98.934 %**	98.677 %	-	-	-	96.370 %	97.360 %
	Precision	100%	95.000 %	100%	99.000 %	-	-	-	95.280 %	97.50%

Moreover, we took use of LSTM network to extract the temporal information (Lu, Yan, & Nguyen, 2020), YOLO methods were employed to extract the spatial information, finally we combine these two networks together by using score fusion and our own datasets. Table 5.2.4 illustrates the comparison of different deep learning methods on our dataset I.

Table 5.2.4 The comparison of different deep learning methods on our dataset I

Our Dataset One	Walk	Skip	Run	Jack	Jump	Accuracy
YOLOv3+LSTM	97.28%	96.41%	98.46%	100%	95.76%	**97.58%**
YOLOv3	96.55%	92.15%	97.82%	100%	95.33%	96.37%
YOLOv4+LSTM	98.13%	97.04%	98.12%	100%	96.06%	97.87%
YOLOv4	97.53%	95.69%	97.79%	100%	95.79%	97.36%
CapsNet	97.66%	97.84%	97.36%	100%	94.74%	97.52%
CapsNet + LSTM	98.44	97.95%	97.86%	100%	96.80%	98.21%

Throughout our experiments, we took use of various deep learning methods to compare our experimental results. The deep learning models with attention mechanism are much stable and robust by adopting our own datasets. In this research project, by combining the YOLOv3 and LSTM together to extract both spatiotemporal information, we are able to achieve the accuracy 97.58%, which has 1.21% growth compared with only extracting spatial information by using YOLOv3 on our own dataset I. Moreover, YOLOv4 gets the accuracy 97.36%. By combining YOLOv4 with LSTM, the total accuracy is up to 97.87%. The CapsNet individually achieved 97.52% accuracy; by combining CapsNet and LSTM with the class score fusion, it achieved 0.69% growth on our dataset I. With regard to this network structure on our dataset II, the result reached up to 96.42% accuracy.

Table 5.2.5 The results of different deep learning methods on our dataset II

Our Dataset Two	**Hello**	**Nice**	**Meet**	**You**	**Accuracy**
DenseNet	95.23%	93.82%	95.28%	94.11%	94.61%
ResNet	94.77%	93.27%	94.89%	91.83%	93.69 %
YOLOv3	94.35%	95.26%	96.71%	97.12%	95.86 %
YOLOv4	96.37%	97.49%	97.55%	98.35%	97.44%
ResNeXt	96.87%	98.64%	98.25%	97.52%	97.82%
ResNext + Attention	97.63%	98.66%	98.36%	98.11%	98.19%
SKNet	97.89%	97.02%	97.93%	98.96%	97.95%
SKNet + Attention	98.91%	98.64%	98.93%	99.04%	98.88%
YOLOv3 + LSTM	97.28%	97.79%	95.33%	95.28%	96.42%
CapsNet	98.76%	98.96%	98.94%	98.22%	98.72%
CapsNet + LSTM	99.24%	98.53%	99.45%	98.62%	98.96%

Table 5.2.5 shows the results of different deep learning methods on our dataset II. SKNet net with attention mechanism shows positive results on our own dataset II. The network of SKNet with attention mechanism is able to get 98.88% accuracy which has 5.19% growth of the total accuracy compared with the traditional deep learning ResNet model. CapsNet reaches 98.72% accuracy by adopting our own dataset II. The combination of CapsNet with LSTM is able to achieve 98.96% total accuracy of recognition rate, which has 2.54% increasing compared with YOLOv3 + LSTM and 0.24% growth compared with only extracting spatial information by adopting CapsNet.

Chapter 6
Conclusion and Future Work

In this thesis, in-depth articulation of the proposed deep learning methods was discussed which is utilized to recognize human behavior in real time. The corresponding state-of-the-art methods in deep learning have been implemented as the results of this thesis. In this chapter, we will present this thesis at a scholarly level, also highly organize and integrate the conclusion into the context, meanwhile the future work will be listed by the end of this thesis.

6.1 Conclusion

The objective of this thesis is to develop the algorithms for human behaviour recognition from surveillance videos and utilize the state-of-the-art methods in deep learning by attaining this goal. In this thesis, we demonstrated these deep learning models are employed for human behaviour recognition which are faster than the traditional machine learning approaches, they also are able to achieve real-time recognition. The main contributions are summarized below.

In this thesis, we have presented multiple deep learning models to fulfil human behavior recognition. Throughout our experiments, we see deep learning methods are well implemented in this research project. Based on our experiments, the results were already up to 90% for the well-selected datasets. Throughout the experiments, the overall results were positive, but the proposed models may have a bit of misclassifications in particular behaviours such as “skip/run” in Weizmann dataset and “jog/run” in KTH dataset. The combination of ResNeXt with attention mechanism and SKNet with attention mechanism also shows the positive outcomes in human behaviour recognition. By adding the attention mechanism, the results are also promising. Moreover, by adopting the YOLO methods, LSTM network with class score fusion to acquire both spatiotemporal information also indicates positive results, which has 1.21% growth of accuracy by only using YOLOv3 network.

From the experimental outcomes, we see that most of deep learning models could be extended through either the depth or width of the network layers so as to improve the accuracy of our methods. However, convolution-based deep learning uplifts the efficiency without beefing up the complexity. It trims off the number of the hyperparameters and enhances the representation of the network. Meanwhile, YOLOv4 + LSTM network is also outperformed in this thesis.

In this research project, we proposed a spatial attention-based SKNet model. According to the research outcomes, by utilizing the attention mechanism, it shows the

positive result in human behavior recognition. YOLO + LSTM network is also proposed in this research work. After the experiments, the result shows that by coping with the sequence data, the temporal information should be further investigated.

Moreover, to the best of our knowledge, it is the first time we proposed a method for human behavior recognition based on ensemble learning by using WEKA (ver3). We took use of two different ensemble learners to evaluate the results based on public datasets. By combining bagging learners with random forest together, both public datasets show that the results are stable and robust. Finally, we collected our own datasets and compared them with public dataset to test the stability and robustness of the proposed models.

6.2 Future Work

Our future work includes,

(1) How to reduce the misclassification will be investigated in future. The attention mechanism should be considered in LSTM network in the near future to make our model more robust and stable. Adding the attention model into the LSTM to the specified parts of the inputs that is thought as important, which improves the performance of the neural network model. This idea is based on the attention scores that can be used in image captioning and machine translation, so there has a way to solve the misclassification problem.

(2) From the research outcomes, we see that the attention mechanism is able to integrate any CNN architectures. Thus, YOLOv3 with attention mechanism will be also probed in the future. The attention mechanism in this case means by adopting the YOLOv3 model and the Convolutional Block Attention Module together. The original paper of CBAM mentioned that the module is able to be integrated into any CNN architectures seamlessly with the end-to-end structure. The next stage is to combine the attention mechanism into the YOLOv3 network. For the YOLOv3 network structure, it contains 5 groups of residual units, we hope to insert the attention module into each unit.

(3) We will mingle channel attention module and spectrum into our model in order to achieve better accuracy in human behavior recognition. In this research work, we only chose the spatial attention which is used to emphasise on those important features and suppress unnecessary ones to improve the representation of networks. Thus, we hope that in the future, the channel attention will be implemented in our research. The channel attention focuses on what is meaningful given an input image, so when we are dealing with the multiple objects, it can be more useful to extract feature maps. Moreover, more different types of data can be studied in future to make our research more suitable in different scenarios, such as depth maps, IR data and thermal spectrum data.

(4) Our deep learning methods for human behavior recognition such as CapsNet, ShuffleNet (the computation-efficient CNN for mobile devices), and NasNet will be explored and exploited in the future. The simulated neurons of spiking neural networks (SNNs) are more realistic, it is also good at the sequence data; in addition, it should be taken into considerations of the temporal information. Moreover, there is not much relevant research right now.

(5) During the research time and the limitation of computation, we will work for multiperson behaver recognition in the future. In addition, more complex human behaviors with multi-person interactions such as talking, fighting, robbery etc. will be explored in the future.

References

Achard, C., Qu, X., Mokhber, A., & Milgram, M. (2008). A novel approach for recognition of human actions with semi-global features. *Machine Vision and Applications*, *19*(1), 27-34

Aggarwal, J. K., & Cai, Q. (1997). Human motion analysis: A review. *Nonrigid and Articulated Motion Workshop*, pp. 99-102

Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Computing Surveys*, pp. 16:1–16:43

Aggarwal, J. K., Cai, Q., Liao, W., & Sabata, B. (1997). Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, 142-156

Aggarwal, J. K., Cai, Q., Liao, W., & Sabata, B. (1997). Nonrigid motion analysis: Articulated and elastic motion. *Computer Vision and Image Understanding*, pp. 142-156

Ahmed, S. M., Lejbolle, A. R., Panda, R., & Roy-Chowdhury, A. K. (2020). Camera on-boarding for person re-identification using hypothesis transfer learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12144-12153)

Al-Sarayreh, M., Reis, M. M., Yan, W. Q., & Klette, R. (2018). Deep spectral-spatial features of snapshot hyperspectral images for red-meat classification. In *International Conference on Image and Vision Computing New Zealand* (pp. 1-6)

Artacho, B., & Savakis, A. (2020). UniPose: Unified human pose estimation in single images and videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7035-7044)

Asadi-Aghbolaghi, M., Clapes, A., Bellantonio, M., Escalante, H. J., Ponce-López, V., Baró, X., ... & Escalera, S. (2017). A survey on deep learning based approaches for action and gesture recognition in image sequences. In *IEEE International Conference on Automatic Face & Gesture Recognition* (pp. 476-483)

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* (pp. 29-39)

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations* (pp. 1-15)

Bai, S., He, Z., Qiao, Y., Hu, H., Wu, W., & Yan, J. (2020). Adaptive dilated network with self-correction supervision for counting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4594-4603)

Bajpai, S., Jain, K., & Jain, N. (2011). Artificial neural networks. *International Journal of Soft Computing and Engineering*, 27-31

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). SURF: Speeded up robust features. In *European Conference on Computer Vision* (pp. 404-417)

Benzine, A., Chabot, F., Luvison, B., Pham, Q. C., & Achard, C. (2020). PandaNet: Anchor-based single-shot multi-person 3D pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6856-6865)

Blank, M., Gorelick, L., Shechtman, E., Irani, M., & Basri, R. (2005). Actions as space-time shapes. In *IEEE International Conference on Computer Vision* (pp. 1395-1402)

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123-140

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5-32

Breuel, T. M. (2017). High performance text recognition using a hybrid convolutional-LSTM implementation. In *International Conference on Document Analysis and Recognition (ICDAR)* (Vol. 1, pp. 11-16)

Bobick, A. F., & Davis, J. W. (2001). The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 257-267

Bobick, A., & Davis, J. (1996). An appearance-based representation of action. In *International Conference on Pattern Recognition* (pp. 307-312)

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*

Bregonzio, M., Gong, S., & Xiang, T. (2009). Recognising action as clouds of space-time interest points. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1948-1955)

Brendel, W., & Todorovic, S. (2009). Video object segmentation by tracking regions. In *International Conference on Computer Vision* (pp. 833-840)

Carreira, J., & Zisserman, A. (2017). Quo Vadis, action recognition? A new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6299-6308)

Chen, L., Ai, H., Chen, R., Zhuang, Z., & Liu, S. (2020). Cross-view tracking for multi-human 3D pose estimation at over 100 FPS. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3279-3288)

Chen, L., & Nugent, C. D. (2019). *Human Activity Recognition and Behavior Analysis*. Springer International Publishing.

Chen, X., Fu, C., Zhao, Y., Zheng, F., Song, J., Ji, R., & Yang, Y. (2020). Salience-guided cascaded suppression network for person re-identification. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (pp. 3300-3310)

Chen, X., & Zhang, C. (2006). An interactive semantic video mining and retrieval platform - Application in transportation surveillance video for incident detection. In *International Conference on Data Mining* (pp. 129-138)

Chen, Y., Zhao, L., Peng, X., Yuan, J., & Metaxas, D. N. (2019). Construct dynamic graphs for hand gesture recognition via spatial-temporal attention. *In British Machine Vision Conference* (pp. 1-13)

Cheng, B., Xiao, B., Wang, J., Shi, H., Huang, T. S., & Zhang, L. (2020). HigherHRNet: Scale-aware representation learning for bottom-up human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5386-5395)

Cheng, Z., Dong, Q., Gong, S., & Zhu, X. (2020). Inter-task association critic for cross-resolution person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2605-2615)

Chéron, G., Laptev, I., & Schmid, C. (2015). P-CNN: Pose-based CNN features for action recognition. In *IEEE International Conference on Computer Vision* (pp. 3218-3226)

Cho, K., van Merrienboer, B., Gulcehre, C., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing* (pp. 1724-1734)

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1251-1258)

Chu, X., Zheng, A., Zhang, X., & Sun, J. (2020). Detection in crowded scenes: One proposal, multiple predictions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 12214-12223)

Ciresan, D., Giusti, A., Gambardella, L. M., & Schmidhuber, J. (2012). Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in Neural Information Processing Systems* (pp. 2843-2851)

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 273–297

Cui, W., Yan, W., (2016) A scheme for face recognition in complex environments. *International Journal of Digital Crime and Forensics*, 8 (1), 26-36

Cutler, R., & Davis, L. S. (2000). Robust real-time periodic motion detection, analysis, and applications. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 781-796

Dadashzadeh, A., Targhi, A. T., Tahmasbi, M., & Mirmehdi, M. (2019). HGR-Net: A fusion network for hand gesture segmentation and recognition. *IET Computer Vision*, 13(8), 700-707

Dai, P., Di, H., Dong, L., Tao, L., & Xu, G. (2008). Group interaction analysis in dynamic context. In *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 275-282

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp.886–893)

Damen, D., & Hogg, D. (2009). Recognizing linked events: Searching the space of feasible explanations. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 927-934)

Diba, A., Fayyaz, M., Sharma, V., Karami, A. H., Arzani, M. M., Yousefzadeh, R., & Van Gool, L. (2017). Temporal 3D ConvNets: New architecture and transfer learning for video classification. *arXiv:1711.08200*.

Diba, A., Sharma, V., & Van Gool, L. (2017). Deep temporal linear encoding networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2329-2338)

Dobhal, T., Shitole, V., Thomas, G., & Navada, G. (2015). Human activity recognition using binary motion image and deep learning. *Procedia Computer Science, 58*, 178-185

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2625-2634)

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., & Darrell, T. (2014). Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning* (pp. 647-655)

Dos Santos, C. C., Samatelo, J. L. A., & Vassallo, R. F. (2020). Dynamic gesture recognition by using CNNs and star RGB: A temporal information condensation. *Neurocomputing*, 238-254

Du, W., Wang, Y., & Qiao, Y. (2017). RPAN: An end-to-end recurrent pose-attention network for action recognition in videos. In *IEEE International Conference on Computer Vision* (pp. 3725-3734)

Duan, J., Zhou, S., Wan, J., Guo, X., & Li, S. Z. (2016). Multi-modality fusion based on consensus-voting and 3D convolution for isolated gesture recognition. *arXiv:1611.06689*

Duta, I. C., Ionescu, B., Aizawa, K., & Sebe, N. (2017). Spatio-temporal vector of locally max pooled features for action recognition in videos. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3205-3214)

Efros, A. A., Berg, A. C., Mori, G., & Malik, J. (2003). Recognizing action at a distance. In *International Conference on Computer Vision* (pp. 726-733)

Elboushaki, A., Hannane, R., Afdel, K., & Koutti, L. (2020). MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for

gesture recognition in RGB-D image sequences. *Expert Systems with Applications*, 139

Elhabian, S. Y., El-Sayed, M. K., & Ahmed, S. H. (2008). Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent Patents on Computer Science*, pp. 32-54

Erhan, D., Szegedy, C., Toshev, A. & Anguelov, D. (2014). Scalable object detection using deep neuron networks. *In IEEE Conference on Computer Vsion and Pattern Recognition* (pp. 2147-2154)

Ertel, W. (2018). *Introduction to Artificial Intelligence*. Springer International Publishing

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, *88*(2), 303-338

Fabbri, M., Lanzi, F., Calderara, S., Alletto, S., & Cucchiara, R. (2020). Compressed volumetric heatmaps for multi-person 3D pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7204-7213)

Fan, C., Peng, Y., Cao, C., Liu, X., Hou, S., Chi, J., ... & He, Z. (2020). GaitPart: Temporal part-based model for gait recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14225-14233)

Fang, H. S., Xie, S., Tai, Y. W., & Lu, C. (2017). RMPE: Regional multi-person pose estimation. In *IEEE International Conference on Computer Vision* (pp. 2334-2343)

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). Learning hierarchical features for scene labeling. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1915 - 1929

Feichtenhofer, C. , Pinz, A. , & Wildes, R. P. . (2016). Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing*

Systems, 3468-3476

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1933-1941)

Freund, Y., & Schapire, R. E. (1995). A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory* (pp. 23-37)

Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD: Deconvolutional single shot detector. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp.1-3)

Fujiyoshi, H., Lipton, A. J., & Kanade, T. (2004). Real-time human motion analysis by image skeletonization. *IEICE Transactions on Information and Systems*, 113-120

Gao, S., Wang, J., Lu, H., & Liu, Z. (2020). Pose-guided visible part matching for occluded person ReID. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11744-11752)

Garcia-Hernando, G., Yuan, S., Baek, S., & Kim, T. K. (2018). First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 409-419)

Gers, F. A., & Schmidhuber, J. (2000). Recurrent nets that time and count. In *International Joint Conference on Neural Networks. Neural Computing: New Challenges and Perspectives for the New Millennium,* (Vol. 3, pp. 189-194)

Gers, F. A., & Schmidhuber, E. (2001). LSTM recurrent networks learn simple context-free and context-sensitive languages. In *IEEE Transactions on Neural Networks*, *12*(6), 1333-1340

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual

prediction with LSTM. *Neural Computation*, *12*(10), 2451-2471

Gerónimo, D., López, A., Ponsa, D., & Sappa, A. D. (2007). Haar wavelets and edge orientation histograms for on-board pedestrian detection. In *Iberian Conference on Pattern Recognition and Image Analysis* (pp. 418-425)

Girdhar, R., & Ramanan, D. (2017). Attentional pooling for action recognition. In *Advances in Neural Information Processing Systems* (pp. 34-45)

Girdhar, R., Ramanan, D., Gupta, A., Sivic, J., & Russell, B. (2017). ActionVLAD: Learning spatio-temporal aggregation for action classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 971-980)

Girshick, R. (2015). Fast R-CNN. In *IEEE International Coferernce on Computer Vision* (pp. 1440-1448)

Girshick, R., Donahue, J., Darrell, T., & Malik., J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580–587)

Gorelick, L., Blank, M., Shechtman, E., Irani, M., & Basri, R. (2007). Actions as space-time shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(12), 2247-2253

Gould, S., Gao, T., & Koller, D. (2009). Region-based segmentation and object detection. In *Advances in Neural Information Processing Systems* (pp. 655-663)

Graves, A., Fernández, S., & Schmidhuber, J. (2007). Multi-dimensional recurrent neural networks. In *International Conference on Artificial Neural Networks* (pp. 549-558)

Graves, A., & Jaitly, N. (2014). Towards end-to-end speech recognition with recurrent neural networks. In *International Conference on Machine Learning* (pp. 1764-1772)

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5-6), 602-610

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. In *IEEE Transactions on Neural Networks and Learning Systems*, *28*(10), 2222-2232

Gu, D., Nguyen, M., Yan, W. (2016) Cross models for twin recognition. *International Journal of Digital Crime and Forensics,* 8 (4), 26-36

Gu, Q., Yang, J., Yan, W. Q., Li, Y., & Klette, R. (2017). Local Fast R-CNN flow for object-centric event recognition in complex traffic scenes. In *Pacific-Rim Symposium on Image and Video Technology* (pp. 439-452)

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

Guo, Y., Xu, G., & Tsuji, S. (1994). Tracking human body motion based on a stick figure model. *Journal of Visual Communication and Image Representation*, 1-9

Habermann, M., Xu, W., Zollhofer, M., Pons-Moll, G., & Theobalt, C. (2020). DeepCap: Monocular human performance capture using weak supervision. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5052-5063)

Hahnloser, R., & Seung, H. S. (2006). Permitted and forbidden sets in symmetric threshold-linear networks. *Neural Computation, 15*(3), 621-638

Hahnloser, R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J., & Seung, H. S. (2000). Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 947-951

Hahnloser., R., & Seung, H. S. (2002). Selectively grouping neurons in recurrent networks of lateral inhibition. *Neural Computation, 14*(11), 2627-2646

Han, M., Chen, J., Li, L., & Chang, Y. (2016). Visual hand gesture recognition with convolution neural network. *In IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 287-291)

Haritaoglu, I., Harwood, D., & Davis, L. S. (1998). W4S: A real-time system for detecting and tracking people in 2 1/2D. In *European Conference on Computer Vision* (pp. 877-892)

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017). Single shot text detector with regional attention. In *IEEE International Conference on Computer Vision* (pp. 3047-3055)

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In *IEEE International Conference on Computer Vision* (pp. 2961-2969)

He, K., & Sun, J. (2015). Convolutional neural networks at constrained time cost. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5353-5360)

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1904-1916

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778)

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *European Conference on Computer Vision* (pp. 630-645)

He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017). Single shot text detector with regional attention. In *IEEE International Conference on Computer Vision* (pp.1-12)

Herath, S., Harandi, M., & Porikli, F. (2017). Going deeper into action recognition: A

survey. *Image and Vision Computing*, *60*, 4-21

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 504-507

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*

Hinton, G. E., Osindero, S., & Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 1527-1554

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554-2558

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4700-4708)

Huang, J., Zhou, W., Li, H., & Li, W. (2015). Sign language recognition using 3D convolutional neural networks. *In IEEE International Conference on Multimedia and Expo* (pp.1-6)

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., ... & Murphy, K. (2017). Speed/accuracy trade-offs for modern convolutional object detectors. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7310-7311)

Huang, J., Zhu, Z., Guo, F., & Huang, G. (2020). The Devil is in the details: Delving into unbiased data processing for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5700-5709)

Huang, X., Ge, Z., Jie, Z., & Yoshie, O. (2020). NMS by representative region: Towards crowded pedestrian detection by proposal pairing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10750-10759)

Huang, Y., Zha, Z. J., Fu, X., Hong, R., & Li, L. (2020). Real-world person re-identification via degradation invariance learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14084-14094)

Ibrahim, M. S., Muralidharan, S., Deng, Z., Vahdat, A., & Mori, G. (2016). A hierarchical deep temporal model for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1971-1980)

Ioffe, S., & Szegedy, C., (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (pp. 448-456)

Isogawa, M., Yuan, Y., O'Toole, M., & Kitani, K. M. (2020). Optical non-line-of-sight physics-based 3D human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7013-7022)

Ivanov, Y. A., & Bobick, A. F. (2000). Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 852-872

Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. In *Advances in Neural Information Processing Systems* (pp. 2017-2025)

Jagadeesh, B., & Patil, C. M. (2019). Video based human activity detection, recognition and classification of actions using SVM. *Transactions on Machine Learning and Artificial Intelligence*, 22-34

Ji, H., Liu, Z., Yan, W. Q., & Klette, R. (2019). Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. In *ACPR* 2(1), 503-515

Ji, H., Liu, Z., Yan, W. Q., & Klette, R. (2019). Early diagnosis of Alzheimer's disease using deep learning. In *International Conference on Control and Computer Vision* (pp. 87-91)

Ji, S., Xu, W., Yang, M., & Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 221 - 231

Jiang, X., Zhang, L., Xu, M., Zhang, T., Lv, P., Zhou, B., ... & Pang, Y. (2020). Attention scaling for crowd counting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4706-4715)

Jin, X., Lan, C., Zeng, W., Chen, Z., & Zhang, L. (2020). Style normalization and restitution for generalizable person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3143-3152)

Joo, S.-W., & Chellappa, R. (2006). Attribute grammar-based event recognition and anomaly detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop* (pp. 107-107)

Joshi, K. A., & Thakore, D. G. (2012). A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 44-48

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., & Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1725-1732)

Ke, Y., Sukthankar, R., & Hebert, M. (2007). Spatio-temporal shape and flow correlation for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8)

Kiciroglu, S., Rhodin, H., Sinha, S. N., Salzmann, M., & Fua, P. (2020). ActiveMoCap: Optimized viewpoint selection for active human motion capture. In *IEEE/CVF*

Conference on Computer Vision and Pattern Recognition (pp. 103-112)

Klaser, A., Marszałek, M., & Schmid, C. (2008). A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference* (pp. 275-284)

Kocabas, M., Athanasiou, N., & Black, M. J. (2020). VIBE: Video inference for human body pose and shape estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5253-5263)

Koller, O., Ney, H., & Bowden, R. (2016). Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. *In IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3793-3802)

Kontschieder, P., Fiterau, M., Criminisi, A., & Bulo, S. R. (2015). Deep neural decision forests. In *IEEE International Conference on Computer Vision* (pp. 1467-1475)

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097-1105)

Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M., & Shafer, S. (2000). Multi-camera multi-person tracking for EasyLiving. In *IEEE International Workshop on Visual Surveillance* (pp. 3-10)

Kulchandani, J. S., & Dangarwala, K. J. (2015). Moving object detection: Review of recent research trends. In *International Conference on Pervasive Computing* (pp. 1-5)

Kundu, J. N., Seth, S., Jampani, V., Rakesh, M., Babu, R. V., & Chakraborty, A. (2020). Self-supervised 3D human pose estimation via part guided novel image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6152-6162)

Lan, Z., Zhu, Y., Hauptmann, A. G., & Newsam, S. (2017). Deep local video feature for

action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 1-7)

Laptev, I. (2005). On space-time interest points. *International Journal of Computer Vision*, *64*(2-3), 107-123

Laptev, I., & Lindeberg, T. (2004). Local descriptors for spatio-temporal recognition. In *International Workshop on Spatial Coherence for Visual Motion Analysis* (pp. 91-103)

Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008). Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8)

Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3361-3368).

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The Handbook of Brain Theory and Neural Networks*

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278-2324

LeCun, Y., Huang, F. J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Conference on Computer Vision and Pattern Recognition*

Lee, C. Y., Gallagher, P. W., & Tu, Z. (2016). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics* (pp. 464-472)

Lee, C. Y., Xie, S., Gallagher, P., Zhang, Z., & Tu, Z. (2015). Deeply-supervised nets.

Artificial Intelligence and Statistics, 562-570

Lev, G., Sadeh, G., Klein, B., & Wolf, L. (2016). RNN fisher vectors for action recognition and image annotation. In *European Conference on Computer Vision* (pp. 833-850)

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, R., Nguyen, M., Yan, W. (2017) Morse codes enter using finger gesture recognition. In *International Conference on Digital Image Computing: Techniques and Applications*

Li, S., Ke, L., Pratama, K., Tai, Y. W., Tang, C. K., & Cheng, K. T. (2020). Cascaded deep monocular 3D human pose estimation with evolutionary training data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6173-6183)

Li, X., Makihara, Y., Xu, C., Yagi, Y., & Ren, M. (2020). Gait recognition via semi-supervised disentangled representation learning to identity and covariate features. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13309-13319)

Li, X., Wang, W., Hu, X., & Yang, J. (2019). Selective kernel networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 510-519)

Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., & Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. In *European Conference on Computer Vision* (pp. 203-220)

Lin, M., Chen, Q., & Yan, S. (2013). Network in network. In *International Conference on Learning Representations*

Lin, T, Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature

pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2117-2125)

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *IEEE International Conference on Computer Vision* (pp. 2980-2988)

Lin, T, Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European Conference on Computer Vision* (pp. 740-755)

Lin, T., Zhao, X., & Shou, Z. (2017). Single shot temporal action detection. In *ACM International Conference on Multimedia* (pp. 988-996)

Lin, Y., Xie, L., Wu, Y., Yan, C., & Tian, Q. (2020). Unsupervised person re-identification via softened similarity learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3390-3399)

Liu, C., Chang, X., & Shen, Y. D. (2020). Unity style transfer for person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6887-6896)

Liu, C., Yan, W. (2020) Gait recognition using deep learning. Handbook of Research on Multimedia Cyber Security, 3, 214-226

Liu, J., Liu, Y., Wang, Y., Prinet, V., Xiang, S., & Pan, C. (2020). Decoupled representation learning for skeleton-based gesture recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5751-5760)

Liu, J., Shahroudy, A., Wang, G., Duan, L. Y., & Kot, A. C. (2018). SSNet: Scale selection network for online 3D action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8349-8358)

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 (pp. 816-833)

Liu, J., Wang, G., Hu, P., Duan, L. Y., & Kot, A. C. (2017). Global context-aware attention LSTM networks for 3D action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1647-1656)

Liu, Q., Zhou, F., Hang, R., & Yuan, X. (2017). Bidirectional-convolutional LSTM based spectral-spatial feature learning for hyperspectral image classification. *Remote Sensing*, *9*(12), 1330

Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8759-8768)

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*, (pp. 21-37)

Liu, X., Nguyen, M., & Yan, W. Q. (2019). Vehicle-related scene understanding using deep learning. In *Asian Conference on Pattern Recognition* (pp. 61-73)

Liu, Z., Zhang, C., & Tian, Y. (2016). 3D-based deep convolutional neural network for action recognition with depth sequences. *Image and Vision Computing*, 55, 93-100

Lu, J., Shen, J., Yan, W., & Bacic, B. (2017). An empirical study for human behavior anaalysis. *International Journal of Digital Crime and Forensics*, 11-27

Lu, J., Yan, W., & Nguyen, M. (2018). Human behaviour recognition using deep learning. In *IEEE International Conference on Advanced Video and Signal Based Surveillance* (pp. 1-6)

Lu, J., Yan, W., & Nguyen, M. (2020). Deep learning methods for Human behaviour recognition. In *IEEE International Conference on Image and Vision Computing*

New Zealand (pp. 1-6)

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

Lu, J., Yan, W., & Nguyen, M. (2021). Sign language recognition from digital videos using deep learning methods. In *International Symposium on Geometry and Vision* (pp. 1-11)

Luvizon, D. C., Picard, D., & Tabia, H. (2018). 2D/3D pose estimation and action recognition using multitask deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5137-5146)

Luo, W., Liu, W., & Gao, S. (2017). Remembering history with Convolutional LSTM for anomaly detection. In *IEEE International Conference on Multimedia and Expo* (pp. 439-444)

Luo, Y., Zhang, C., Zhao, M., Zhou, H., & Sun, J. (2020). Where, what, whether: Multi-modal learning meets pedestrian detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 14065-14073)

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing* (pp. 1412-1421)

Ma, S., Sigal, L., & Sclaroff, S. (2016). Learning activity progression in LSTMs for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1942-1950)

Mahasseni, B., & Todorovic, S. (2016). Regularizing long short-term memory with 3D human-skeleton sequences for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3054-3062)

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 115–133

Meng, Y., Jin, Y., & Yin, J. (2011). Modeling activity-dependent plasticity in BCM spiking neural networks with application to human behavior recognition. *IEEE Transactions on Neural Networks*, 1952-1966

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to WordNet: An on-line lexical database. *International Journal of Lexicography*, *3*(4), 235-244

Miller, J. F., & Khan, G. M. (2011). Where is the brain inside the brain? On why artificial neural networks should be developmental. *Memetic Computing*, 217-228

Minnen, D., Essa, I., & Starner, T. (2003). Expectation grammars: Leveraging high-level expectations for activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 626-632)

Misra, D. (2019). Mish: A self regularized non-monotonic neural activation function. *arXiv preprint arXiv:1908.08681*

Mitra, R., Gundavarapu, N. B., Sharma, A., & Jain, A. (2020). Multiview-consistent semi-supervised learning for 3D human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 6907-6916)

Mnih, V., Heess, N., & Graves, A. (2014). Recurrent models of visual attention. In *Neural Information Processing Systems* (pp. 2204-2212)

Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. *In IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4207-4215)

Moore, D., & Essa, I. (2002). Recognizing multitasked activities from video using

stochastic context-free grammar. *AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (pp. 770-776)

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning* (pp. 807-814)

Neverova, N., Wolf, C., Taylor, G. W., & Nebout, F. (2014). Hand segmentation with structured convolutional learning. *In Asian Conference on Computer Vision* (pp. 687-702)

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision* (pp. 483-499)

Nguyen, N. T., Phung, D. Q., Venkatesh, S., & Bui, H. (2005). Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 955-960)

Niebles, J. C., Wang, H., & Fei-Fei, L. (2008). Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 299-318.

Norvig, P., & Russell, S., (2016). Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River.

Osayamwen, F., & Tapamo, J.-R. (2019). Deep learning class discrimination based on prior probability for human activity recognition. *IEEE Access, 7*, 14747-14756

Pan, C., Yan, W. (2019) A learning-based positive feedback in salient object detection. IEEE IVCNZ, New Zealand.

Pan, C., Yan, W. (2020) Salient object detection based on perception saturation. Springer Multimedia Tools and Applications, 79 (27-28), 19925-19944

Papageorgiou, C., Oren, M., & Poggio, T. (1998). A general framework for object detection. In *International Conference on Computer Vision* (pp. 555-562)

Petrushin, V. A. (2005). Mining rare and frequent events in multi-camera surveillance video using self-organizing maps. *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 794-800)

Pezeshki, M., Fan, L., Brakel, P., Courville, A., & Bengio, Y. (2016). Deconstructing the ladder network architecture. In *International Conference on Machine Learning* (pp. 2368-2376)

Piccardi, M. (2004). Background subtraction techniques: a review. In *IEEE International Conference on Systems, Man and Cybernetics* (vol. 4, pp. 3099-3104)

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

Poppe, R. (2007). Vision-based human motion analysis: An overview. *Computer Vision and Image Understanding*, 4-18

Poppe R. (2010). A survey on vision-based human action recognition. *Image and vision computing*, *28*(6), 976-990

Qiu, Z., Yao, T., & Mei, T. (2017). Learning spatio-temporal representation with pseudo-3D residual networks. In *IEEE International Conference on Computer Vision* (pp. 5533-5541)

Rakibe, R. S., & Patil, B. D. (2013). Background subtraction algorithm based human motion detection. *International Journal of Scientific and Research Publications*, *3*(5), 2250-3153

Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems* (pp. 3546-3554)

Rastgoo, R., Kiani, K., & Escalera, S. (2018). Multi-modal deep hand sign language

recognition in still images using restricted Boltzmann machine. *Entropy*, 20(11), 809

Rastgoo, R., Kiani, K. & Escalera, S. (2020). Video-based isolated hand sign language recognition using a deep cascaded model. *Springer Multimedia Tools Application*, 79, 22965–22987

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

Redmon, J., Divvala, S., Girshick, R., & Farhadi., A. (2016). You only look once: Unified, real-time object detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 779-788)

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7263-7271)

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards real-time object detection with region proposal networks. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1137-1149

Rodriguez, M. D., Ahmed, J., & Shah, M. (2008). Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8)

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386-408

Rui, Y., & Anandan, P. (2000). Segmenting visual actions based on spatio-temporal motion patterns. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 111-118)

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., & Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of*

Computer Vision, 115(3), 211-252

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 3859-3869).

Saha, S., Singh, G., Sapienza, M., Torr, P. H., & Cuzzolin, F. (2016). Deep learning for detecting multiple space-time action tubes in videos. *arXiv:1608.01529*

Sande, K., Uijlings, J., Gevers, T., & Smeulders, A. (2011). Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision* (pp.1879–1886)

Sarikaya, R., Hinton, G. E., & Deoras, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *22*(4), 778-784

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. In *IEEE Transactions on Signal Processing*, *45*(11), 2673-2681

Scovanner, P., Ali, S., & Shah, M. (2007). A 3-dimensional sift descriptor and its application to action recognition. In *ACM International Conference on Multimedia* (pp. 357-360)

Sharma, S., Kiros, R., & Salakhutdinov, R. (2015). Action recognition using visual attention. *arXiv:1511.04119*

Shen, D., Xin, C. & Yan, W. (2018) Flame detection using deep learning. In *IEEE ICCAR*, (pp.53)

Shen, Y., & Yan, W. Q. (2018). Blind spot monitoring using deep learning. In *International Conference on Image and Vision Computing New Zealand* (pp. 1-5).

Shrivastava, A., Gupta, A., & Girshick, R. (2016). Training region-based object detectors

with online hard example mining. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 761-769)

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* (pp. 1227-1236)

Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In A*dvances in Neural Information Processing Systems* (pp. 568-576)

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*

Singh, B., Marks, T. K., Jones, M., Tuzel, O., & Shao, M. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1961-1970)

Singh, G., Saha, S., S. M., Torr, P. H., & Cuzzolin, F. (2017). Online real-time multiple spatiotemporal action localisation and prediction. In *IEEE International Conference on Computer Vision* (pp. 3637-3646)

Sondak, N. E., & Sondak, V. K. (1989). Neural networks and artificial intelligence. *ACM SIGCSE Bulletin*, 241-245

Song, C., He, L., Yan, W. Q., & Nand, P. (2019). An improved selective facial extraction model for age estimation. In *International Conference on Image and Vision Computing New Zealand* (pp. 1-6)

Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI Conference on Artificial Intelligence* (pp. 4263-4270)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014).

Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Training very deep networks. In *Advances in Neural Information Processing Systems* (pp. 2377-2385)

Stauffer, C., & Grimson, W. E. (1999). Adaptive background mixture models for real-time tracking. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2246)

Stringa, E. (2000). Morphological change detection algorithms for surveillance applications. In *British Machine Vision Conference* (pp. 1-10)

Sun, L., Jia, K., Yeung, D. Y., & Shi, B. E. (2015). Human action recognition using factorized spatio-temporal convolutional networks. In *IEEE International Conference on Computer Vision* (pp. 4597-4605).

Sutskever, I., Martens, J., & Hinton, G. E. (2011). Generating text with recurrent neural networks. In *International Conference on Machine Learning* (pp. 1017-1024)

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems* (pp. 3104-3112)

Szegedy, C., Toshev, A., & Erhan., D. (2013). Deep neural networks for object detection. *International Conference on Neural Information Processing Systems* (pp. 2553–2561)

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE Conference on Computer Vision and pattern recognition* (pp. 1-9)

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI Conference on Artificial Intelligence*

Tang, X., Du, D. K., He, Z., & Liu, J. (2018). Pyramidbox: A context-assisted single shot face detector. In *European Conference on Computer Vision* (pp. 797-813)

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatio-temporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision* (pp. 4489-4497)

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., & Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6450-6459)

Tu, Z., Li, H., Zhang, D., Dauwels, J., Li, B., & Yuan, J. (2019). Action-stage emphasized spatiotemporal VLAD for video action recognition. *IEEE Transactions on Image Processing*, *28*(6), 2799-2812

Uijlings, J., Sande, K., Gevers, T., & Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 154–171

Varamesh, A., & Tuytelaars, T. (2020). Mixture dense regression for object detection and human pose estimation. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition* (pp. 13086-13095)

Wang, D., & Zhang, S. (2020). Unsupervised person re-identification via multi-label classification. In *IEEE / CVF Conference on Computer Vision and Pattern Recognition* (pp. 10981-10990)

Wang, G., Lai, J. H., Liang, W., & Wang, G. (2020). Smoothing adversarial domain attack and *p*-memory reconsolidation for cross-domain person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10568-10577)

Wang, G. A., Yang, S., Liu, H., Wang, Z., Yang, Y., Wang, S., ... & Sun, J. (2020). High-order information matters: Learning relation and topology for occluded person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern*

Recognition (pp. 6449-6458)

Wang, H., Kläser, A., Schmid, C., & Liu, C. L. (2013). Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, *103*(1), 60-79

Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision* (pp. 3551-3558)

Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference* (pp. 124-1)

Wang, H., Wang, G., Li, Y., Zhang, D., & Lin, L. (2020). Transferable, controllable, and inconspicuous adversarial attacks on person re-identification with deep mis-ranking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 342-351)

Wang, J., Wei, Z., Zhang, T., & Zeng, W. (2016). Deeply-fused nets. *arXiv:1605.07716*

Wang, K., Wang, X., Lin, L., Wang, M., & Zuo, W. (2014). 3D human activity recognition with reconfigurable convolutional neural networks. *ACM International Conference on Multimedia* (pp. 97-106)

Wang, L., Qiao, Y., & Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4305-4314)

Wang, L., & Suter, D. (2006). Informative shape representations for human action recognition. In *International Conference on Pattern Recognition* (Vol. 2, pp. 1266-1269)

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition.

In *European Conference on Computer Vision* (pp. 20-36)

Wang, M., Tighe, J., & Modolo, D. (2020). Combining detection and tracking for human pose estimation in videos. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11088-11096)

Wang, P., Li, W., Liu, S., Gao, Z., Tang, C., & Ogunbona, P. (2016). Large-scale isolated gesture recognition using convolutional neural networks. In *International Conference on Pattern Recognition* (pp. 7-12)

Wang, T., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., & Shao, L. (2019). Learning rich features at high-speed for single-shot object detection. In *IEEE International Conference on Computer Vision* (pp. 1971-1980)

Wang, X., Yan, W. (2019) Multi-perspective gait recognition based on ensemble learning. *Springer Neural Computing and Applications*.

Wang, X., Yan, W. (2019) Gait recognition using multichannel convolution neural networks. *Neural Computing and Applications*

Wang, X., Feng, S., & Yan, W. Q. (2019). Human gait recognition based on self-adaptive hidden Markov model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1-10

Wang, X., & Yan, W. Q. (2020). Human gait recognition based on frame-by-frame gait energy images and convolutional long short-term memory. *International Journal of Neural Systems*, *30*(01), 1950027

Wang, X., Zhang, J., & Yan, W. (2020). Gait recognition using multichannel convolution neural networks. *Springer Neural Computing and Applications*, 1-11

Wang, X., Yan, W. (2020) Non-local gait feature extraction and human identification. *Multimedia Tools and Applications*, Springer.

Wang, Y., Long, M., Wang, J., & Yu, P. S. (2017). Spatiotemporal pyramid network for

video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1529-1538)

Wang, Y., Huang, K., & Tan, T. (2007). Human activity recognition based on R transform. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8)

Weinland, D., & Boyer, E. (2008). Action recognition using exemplar-based embedding. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-7)

Weinland, D., Boyer, E., & Ronfard, R. (2007). Action recognition from arbitrary views using 3D exemplars. In *International Conference on Computer Vision* (pp. 1-7)

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, *78*(10), 1550-1560

Willems, G., Tuytelaars, T., & Van Gool, L. (2008). An efficient dense and scale-invariant spatio-temporal interest point detector. In *European Conference on Computer Vision* (pp. 650-663)

Woo, S., Park, J., Lee, J. Y., & So Kweon, I. (2018). CBAM: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)* (pp. 3-19)

Wren, C. R., Azarbayejani, A., Darrell, T., & Pentland, A. P. (1997). Pfinder: Real-time tracking of the human body. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 780-785

Wu, B., & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors. In *IEEE International Conference on Computer Vision* (pp. 1:90–97)

Wu, C. Y., Zaheer, M., Hu, H., Manmatha, R., Smola, A. J., & Krähenbühl, P. (2018). Compressed video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6026-6035)

Wu, J., Ishwar, P., & Konrad, J. (2016) Two-stream CNNs for gesture-based verification

and identification: Learning user style. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 42-50)

Wu, J., Zhou, C., Yang, M., Zhang, Q., Li, Y., & Yuan, J. (2020). Temporal-context enhanced detection of heavily occluded pedestrians. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13430-13439)

Xiang, T., & Gong, S. (2005). Video behaviour profiling and abnormality detection without manual labelling. In *IEEE Conference on Computer Vision* (pp. 1238-1245)

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1492-1500)

Xingjian, S. H. I., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *IEEE Advances in Neural Information Processing Systems* (pp. 802-810)

Xu, J., Yu, Z., Ni, B., Yang, J., Yang, X., & Zhang, W. (2020). Deep kinematics analysis for monocular 3D human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 899-908)

Xu, H., Das, A., & Saenko, K. (2017). R-C3D: Region convolutional 3D network for temporal activity detection. In *IEEE International Conference on Computer Vision* (pp. 5783-5792)

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R. & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning* (pp. 2048-2057)

Xu, L., Xu, W., Golyanik, V., Habermann, M., Fang, L., & Theobalt, C. (2020). EventCap: Monocular 3D capture of high-speed human motions using an event camera.

In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4968-4978)

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI Conference on Artificial Intelligence*

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

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

Yan, Y., Qin, J., Chen, J., Liu, L., Zhu, F., Tai, Y., & Shao, L. (2020). Learning multi-granular hypergraphs for video-based person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2899-2908)

Yan, Z., Zhang, H., Wang, B., Paris, S., & Yu, Y. (2016). Automatic photo adjustment using deep neural networks. *ACM Transactions on Graphics*, *35*(2), 1-15

Yang, J., Zheng, W. S., Yang, Q., Chen, Y. C., & Tian, Q. (2020). Spatial-temporal graph convolutional network for video-based person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3289-3299)

Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., & Sebe, N. (2020). Reverse perspective network for perspective-aware object counting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 4374-4383)

Yao, Z., Cao, Y., Zheng, S., Huang, G., & Lin, S. (2020). Cross-iteration batch normalization. *arXiv:2002.05712*

Yeung, S., Russakovsky, O., Mori, G., & Fei-Fei, L. (2016). End-to-end learning of action detection from frame glimpses in videos. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2678-2687)

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2016). ABCNN: Attention-based

convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, *4*, 259-272

Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4694-4702)

Yu, Z., Yan, W. (2020) Human action recognition using deep learning methods. IEEE IVCNZ, New Zealand

Zagoruyko, S., & Komodakis, N. Wide residual networks (2016) In *BMVC* (pp. 35-67)

Zanfir, A., Marinoiu, E., & Sminchisescu, C. (2018). Monocular 3D pose and shape estimation of multiple people in natural scenes-the importance of multiple scene constraints. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2148-2157)

Zeng, K., Ning, M., Wang, Y., & Guo, Y. (2020). Hierarchical clustering with hard-batch triplet loss for person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13657-13665)

Zhai, Y., Lu, S., Ye, Q., Shan, X., Chen, J., Ji, R., & Tian, Y. (2020). Ad-cluster: Augmented discriminative clustering for domain adaptive person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 9021-9030)

Zhang, F., Zhu, X., Dai, H., Ye, M., & Zhu, C. (2020). Distribution-aware coordinate representation for human pose estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7093-7102)

Zhang, L., Zhu, G., Shen, P., Song, J., Afaq Shah, S., & Bennamoun, M. (2017). Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition. In *IEEE International Conference on Computer Vision*

Workshops (pp. 3120-3128)

Zhang, Q., & Yan, W. Q. (2018). Currency detection and recognition based on deep learning. In *IEEE International Conference on Advanced Video and Signal Based Surveillance,* (pp. 1-6)

Zhang, T., Huang, B., & Wang, Y. (2020). Object-occluded human shape and pose estimation from a single color image. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7376-7385)

Zhang, Y., An, L., Yu, T., Li, X., Li, K., & Liu, Y. (2020). 4D association graph for realtime multi-person motion capture using multiple video cameras. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 1324-1333)

Zhang, Y., Lee, K., & Lee, H. (2016). Augmenting supervised neural networks with unsupervised objectives for large-scale image classification. In *International Conference on Machine Learning* (pp. 612-621)

Zhang, Y., Yan, W., Narayanan, A. (2017) A virtual keyboard implementation using finger recognition, In *IEEE IVCNZ*

Zhang, Z., Gao, J., Mao, J., Liu, Y., Anguelov, D., & Li, C. (2020). STINet: Spatio-temporal-interactive network for pedestrian detection and trajectory prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11346-11355)

Zhang, Z., Lan, C., Zeng, W., Jin, X., & Chen, Z. (2020). Relation-aware global attention for person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3186-3195)

Zhang, Z., Lan, C., Zeng, W., & Chen, Z. (2020). Multi-granularity reference-aided attentive feature aggregation for video-based person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 10407-10416)

Zhao, X., Sang, L., Ding, G., Han, J., Di, N., & Yan, C. (2019). Recurrent attention model for pedestrian attribute recognition. In *AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 9275-9282)

Zheng, K., Yan, W., Nand, P. (2017) Video dynamics detection using deep neural networks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(3), 21

Zheng, Y., Shen, C., Hartley, R., & Huang, X. (2010). Pyramid center-symmetric local binary/trinary patterns for effective pedestrian detection. In *Asian Conference on Computer Vision* (pp. 281-292)

Zhou, B., Andonian, A., Oliva, A., & Torralba, A. (2018). Temporal relational reasoning in videos. In *European Conference on Computer Vision* (pp. 803-818)

Zhou, J., Su, B., & Wu, Y. (2020). Online joint multi-metric adaptation from frequent sharing-subset mining for person re-identification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 2909-2918)

Zhou, Y., Sun, X., Zha, Z. J., & Zeng, W. (2018). MiCT: Mixed 3D/2D convolutional tube for human action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 449-458)

Zhou, Y., Xu, W., Tao, H., & Gong, Y. (2005). Background segmentation using spatial-temporal multi-resolution MRF. In *IEEE Workshop on Applications of Computer Vision* (pp. 8-13)

Zhu, G., Zhang, L., Shen, P., & Song, J. (2017). Multimodal gesture recognition using 3D convolution and convolutional LSTM. *IEEE Access*, *5*, 4517-4524

Zhu, J., Zhu, Z., & Zou, W. (2018). End-to-End video-level representation learning for action recognition. In *International Conference on Pattern Recognition* (pp. 645-650)

Zhu, W., Hu, J., Sun, G., Cao, X., & Qiao, Y. (2016). A key volume mining deep framework for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1991-1999)

Zhu, Y., Lan, Z., Newsam, S., & Hauptmann, A. (2018). Hidden two-stream convolutional networks for action recognition. In *Asian Conference on Computer Vision* (pp. 363-378). Springer

Zitnick, C. L., & Dollár, P. (2014). Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision* (pp. 391-405)

Zong, M., Wang, R., Chen, Z., Wang, M., Wang, X., & Potgieter, J. (2020). Multi-cue based 3D residual network for action recognition. *Neural Computing and Applications*, 1-15