C02029: Doctor of Philosophy
CRICOS Code: 00099F
May 2019

Video-based similar gesture action recognition using deep learning and GAN-based approaches

DI WU

School of Computer Science
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

Video-based similar gesture action recognition using deep learning and GAN-based approaches

A thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

by

DI WU

to

School of Computer Science
Centre for Artificial Intelligence
Faculty of Engineering and Information Technology
University of Technology Sydney
NSW - 2007, Australia

May 2019

Abstract

Human action is not merely a matter of presenting patterns of motion of different parts of the body, in addition, it is also a description of intention, emotion and thoughts of the person. Hence, it has become a crucial component in human behavior analysis and understanding. Human action recognition has a wide variety of applications such as surveillance, robotics, health care, video searching and human-computer interaction. Analysing human actions manually is tedious and easily prone to errors. Therefore, computer scientists have been trying to bring the abilities of cognitive video understanding to human action recognition systems by using computer vision techniques. However, human action recognition is a complex task in computer vision because of the camera motion, occlusion, background cluttering, viewpoint variation, execution rate and similar gestures. These challenges significantly degrade the performance of the human action recognition system. The purpose of this research is to propose solutions based on traditional machine learning methods as well as the state-of-the-art deep learning methods to automatically process video-based human action recognition. This thesis investigates three research areas of video-based human action recognition: traditional human action recognition, similar gesture action recognition, and data augmentation for human action recognition.

To start with, the feature-based methods using classic machine learning algorithms have been studied. Recently, deep convolutional neural networks (CNN) have taken their place in the computer vision and human action recognition research areas and have achieved tremendous success in comparison to traditional machine learning techniques. Current state-of-the-art deep convolutional neural networks were used for the human action recognition task. Furthermore, recurrent neural networks (RNN) and its variation of long-short term memory (LSTM) are used to process the time series features which are handcrafted features or extracted from the CNN. However, these methods suffer from similar gestures, which appear in the human action videos. Thus, a hierarchical classification framework is proposed for similar gesture action recognition, and the performance is improved by the multi-stage classification approach. Additionally, the framework has been modified into an end-to-end system, therefore, the similar gestures can be processed by the system automatically.

In this study, a novel data augmentation framework for action recognition has been proposed, the objective is to generate well learnt video frames from action videos which can enlarge the dataset size as well as the feature bias. It is very important for a human action recognition system to recognize the actions with similar gestures as accurately

as possible. For such a system, a generative adversarial net (GAN) is applied to learn the original video datasets and generate video frames by playing an adversarial game. Furthermore, a framework is developed for classifying the original dataset in the first place to obtain the confusion matrix using a CNN. The similar gesture actions will be paired based on the confusion matrix results. The final classification result will be applied on the fusion dataset which contains both original and generated video frames. This study will provide realtime and practical solutions for autonomous human action recognition system. The analysis of similar gesture actions will improve the performance of the existing CNN-based approaches.

In addition, the GAN-based approaches from computer vision have been applied to the graph embedding area, because graph embedding is similar to image embedding but used for different purposes. Unlike the purpose of the GAN in computer vision for generating the images, the GAN in graph embedding can be used to regularize the embedding. So the proposed methods are able to reconstruct both structural characteristics and node features, which naturally possess the interaction between these two sources of information while learning the embedding.

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, *Di Wu* declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the *School of Computer Science* and *Centre for Artificial Intelligence*, *Faculty of Engineering and Information Technology* at the University of Technology Sydney. This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. This document has not been submitted for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

SIGNATURE: Production Note: Signature removed prior to publication.

[Di Wu]

DATE: 06th May, 2019

PLACE: Sydney, Australia

ACKNOWLEDGMENTS

I would like to express my profound gratitude to my supervisors Prof. Michael Blumenstein (Principal supervisor) and Dr Nabin Sharma (Associate Supervisor) for their fruitful guidance and support throughout my PhD candidature. Their encouragement, interactive regular meetings have made my experience very productive. I am especially thankful to Prof. Michael for his support during the stressful time when I was under highly research pressure and also for my transfer of PhD studies from Griffith University to the University of Technology Sydney. I am also especially thankful Dr Nabin Sharma for taking his time help me to improve my research papers and take care of my research progress.

I am thankful to Griffith University for giving me the opportunity to pursue my PhD studies by granting me scholarships to cover my tuition fee and living expenses. I am thankful to the School of ICT, IIIS and GGRS of Griffith University for their support during the first year of my PhD.

I am thankful to UTS for providing me with the scholarship for the remaining period of my PhD. I cannot forget to mention the state-of-the-art facilities for doing my experiments. I express my gratitude to all staff at the School of Computer Science, CAI, GRS and FEIT staffs of UTS for their assistance. I am lucky to find great friends. I am grateful to Dr Ruiqi Hu, Junjun Chen, Jiale Zhang, Muhammad Saqib, Dr Ranju Mandal, Chandranath Adak and Fujin Zhu for their support and great company.

Finally, I would not have been able to focus on my PhD studies without the constant support, especially the support from my wife, grand father, grand mother and my parents.

List of Publications

Conference :

1. Wu, D., Sharma, N. and Blumenstein, M., 2017, May. Recent advances in video-based human action recognition using deep learning: a review. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2865-2872). IEEE.

2. Wu, D., Sharma, N. and Blumenstein, M., 2018, December. Similar Gesture Recognition using Hierarchical Classification Approach in RGB Videos. In 2018 Digital Image Computing: Techniques and Applications (DICTA) (pp. 1-7). IEEE. **Research Question 1**

3. Wu, D., Sharma, N. and Blumenstein, M., 2018, November. An End-to-End Hierarchical Classification Approach for Similar Gesture Recognition. In 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ) (pp. 1-6). IEEE. **Research Question 2**

4. Wu, D., Chen, J., Sharma, N., Pan. S., Long, G. and Blumenstein, M., 2019. Adversarial Action Data Augmentation for Similar Gesture Action Recognition. In 2019 International Joint Conference on Neural Networks (IJCNN). IEEE. (Accepted) **Research Question 3**

5. Wu, D., Hu, R., Zheng, Y., Jiang. J., Sharma, N. and Blumenstein, M., 2019. Feature-Dependent Graph Convolutional Autoencoders with Adversarial Training Methods. In 2019 International Joint Conference on Neural Networks (IJCNN). IEEE. (Accepted)

6. Zhang, J., Chen, J., Wu, D., Chen, B., and Yu, S., 2019. Achieving Poisoning Attack in Federated Learning using Generative Adversarial Nets. International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom). IEEE. (Accepted)

7. Zhao, Y., Chen, J., Zhang, J., Wu, D., Teng, J., and Yu, S., 2019. Using Generative Adversarial Network to Defend Poisoning Attacks in Federated Learning. In 2019 International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP). Springer. (Accepted)

JOURNAL :

1. Wu, D., Chen, J., Sharma, N., Zhang, J., Zhang, Q. and Blumenstein, M., 2019. An End-to-End Adversarial Video Data Augmentation Framework for Similar Gesture Action Recognition. Expert Systems with Applications (Under Review), **Research Question 3**

2. Chou, K.P., Prasad, M., Wu, D., Sharma, N., Li, D.L., Lin, Y.F., Blumenstein, M., Lin, W.C. and Lin, C.T., 2018. Robust Feature-Based Automated Multi-View Human Action Recognition System. IEEE Access, 6, pp.15283-15296.

3. Chen, J., Wu, D., Zhao, Y., Sharma, N., Blumenstein, M., and Yu, S., 2019. Fooling Intrusion Detection Systems by Using Adversarially Regularized Autoencoder. Digital Communications and Networks, (Under Review)

4. Zhao, Y., Chen, J., Wu, D., Sharma, N., Sajjanhar, A., and Blumenstein, M., 2019. Network Anomaly Detection by Using Time-decay Closed Frequent Pattern. Information, 10(8), p.262.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Human actions are something that people do or cause to happen. The human actions can be extended to events which is something that happens at the given place and time. Analysing a human action is not merely a matter of presenting patterns of motion of different parts of the body, rather, it is also a description of a person's intention, emotion and thoughts. Hence, it has become a crucial component in human behavior analysis and understanding, which are essential in various domains including surveillance, robotics, health care, video searching, human-computer interaction, etc. Take surveillance for example, human operators are required to continuously monitor the sheer number of CCTV cameras installed, to detect any unusual and abnormal actions by suspicious person in public places. It is very difficult for a human operator to continuously focus and watch multiple monitors at the same time and analyze a huge amount of video stream from different cameras. Therefore, a large staff is required to monitor a large crowd, which results in very high costs and the real risk may be ignored due to the limitation of the human eyes. In addition, works such as robotics, health care, video searching and human-computer interaction, automatic video processing is required, thus the human actions cannot be recognized manually. Many researchers are turning toward computer vision for human action recognition system, which will automate the whole process of action recognition, thus improving the efficiency and accuracy of the system very significantly. An important application of video-based human action recognition is monitoring human actions in public places. The multiple cameras will be used to capture the human actions from different view-points. And the feature-based action

recognition methods will be applied to detect and analyze the captured human actions to prevent suspicious damage. Over the past decades, deep learning methods take the place in computer vision and human action recognition research area, which have achieved tremendous success. Convolution neural networks (CNN) have been widely used to process the images and recognize the human action by extracting and learning the features from the video frames automatically. The CNN use local connectivity of a region in the input image to the output, unlike the traditional feed forward neural network that every input layer is fully connected with the output layer. However, CNN-based methods are not sensitive to the different actions with similar gestures, thus, the performance of the classifier will be decreased. In addition, the state-of-the-arts work requires large amount of training data, which are hardly to implement into real-world applications due to the lack of the training data for specific datasets.

The main contribution of the thesis will be to automate the process of human action recognition, improve the performance of similar gesture action recognition as well as the data augmentation for action recognition. The thesis will study the human action recognition methods using traditional machine learning and deep learning based approaches.

1.1 Background and motivation

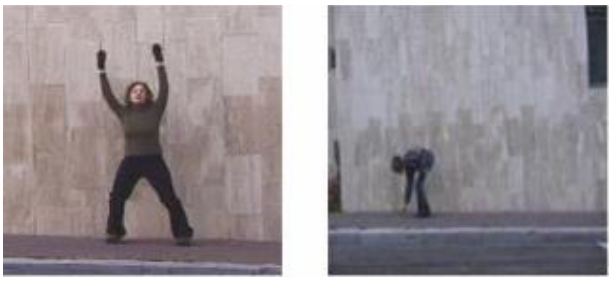
Human acts with purposes and does not matter how trivial it is. For example, in order to play golf, a player must bend and swing the pole which is a combination of gestures. However, some of the purpose are simple such as move from one place to another places, so people act "walking", "jogging", and "running" to achieve this target. An action can be observed by either human eyes or captured by the visual sensors. Furthermore, human brain can process the observed actions based on the basic understanding of the action purpose. People can understand the reason why a person is kicking a ball when they playing soccer, and people could recognize that if the action is follow the instruction or not with a certain confidence. However, it is results in very high costs to involve so many human labours to recognize human actions in different real-world scenarios. Therefore, a smart visual surveillance and automate process system is needed to recognize human actions.

Recognizing human actions from a video stream is a challenging task and has received significant attention from the computer vision research community recently and it has a wide range of applications such as smart video surveillance [1] [163], video indexing [122]

[160], human-machine interaction [92] [97] and identity recognition [93]. The ultimate goals of the artificial intelligence human action recognition research is to develop a system which can efficiently and accurately understand human actions as well as their intentions. Some of the actions are shown in Fig 1.1, where actions can be classified from simple gestures to complex activities such as "Waving" and "Bend" can be determined as simple stand gestures, "Running" and "Walking" are running actions, some of the actions are interacting with objects like "Pickup phone call" and "Hugging", and lastly human group activities also can be considered as human actions which contains complex sequences of gestures. Therefore, understanding the dynamics of different gestures is very important in human action recognition system.

Most of the human action recognition approaches are based on specific data, such as RGB data, depth data, or skeleton data. The focus of the thesis will be about the visual analysis of the human actions as well as the similar gesture recognition in RGB videos. The aim is to make a novel human action recognition system which can capture and analysis human actions efficiently and accurately, so that actions can be correctly identified. Therefore, human action recognition has been the area of interest of most of the computer scientist and researchers. Although, the main purpose of the human action recognition is to design an intelligent system which can provide a chance to help people fully leverage the ability of the computers. It provide the valuable information to identify abnormal actions to warn the authorities before bad things happen. It also provide valuable information about different action videos to help people obtain the useful videos quick. Lastly, it provide the chance that machine can understand what are the people doing using their "eyes" (cameras) instead of passing the signals with specific devices.

With the most intelligent visual device, human eye can capture the low-level information and pass it to the brain for further processing. Brain takes the signals and extracts high-level semantic and contextual information about the scene. Computer vision researchers are trying to bring the same intelligence and perceptual capabilities as the human vision system. Till now, researchers in the computer vision community have succeed in the human actions by hand-craft features in a variety of scenes. But the action is much more complex than the features what extracted. And therefore hand-crafted features are problem dependent which cannot apply to real-time and complex scenes. Because in the case of human action recognition, researchers face difficult challenges to deal with camera motion, occlusion, background cluttering, view point variation, execution rate and similar gestures.



(a) Gestures: Waving and Bend [9]

(b) Actions: Running and Walking [9]

(c) Interactions: Pickup phone call and Hugging [67]

(d) Group activities: Volleyball [79] and Basketball [125]

Figure 1.1: Different types of human actions reported in [146]

The general implementation pipeline for the human action recognition include following steps. The first important step in human action analysis is to extract the video into frames, where the features can be easily extracted from frames such as optical flow or interest points. Generally, the feature-based methods will be extracted from the frames, and followed by send the features to train a classifier and do the classification. Another mostly used methods are deep learning-based, unlike feature-based methods, the extracted frames will be the input of the neural network directly. The neural network will process the localized features by different neurons just like human brain. Deep learning-based methods seem to be more accurate and used widely as compared to feature-based methods. Moreover, the hand-craft features can be the input of the neural network as well. These two-stream methods provide both spatial and temporal information about the human actions. In the second step, to obtain the final result, the results from two streams will be assembled by one fully connected layer of the neural network.

Many human action recognition works have been done by researchers. However, the early works have the drawbacks such as they did not consider the temporal and motion information in the video frames. To adequately address this problem, new approaches have been proposed to process both spatial and temporal information. Indeed, the temporal information approaches improved the performance of the action recognition. However, human actions in videos are not as simple as static objects. With the different actions, the body parts will follow different sequence of gestures listed Fig 1.2. The gestures will be very similar in the most of videos frames when the people perform certain actions. For instance, playing golf is very similar as picking up something, because in the most frames people are supposed to bend their back which is very similar as in the Fig 1.2(a). Similar situations will happen in the case of "Swing and Throw"(Fig 1.2(b)), "Chew and Laugh" (Fig 1.2(c)) and "Turn and Walk" (Fig 1.2(d)). Hence, the drawback of CNN in videos are obvious, as CNN will generate almost the similar features on some of the actions with the similar gestures.

1.2 Significance of video-based human action recognition

Human action recognition has many real-world critical applications. The applications are given below.

(a) Rope Climbing and Rock Climbing Indoor

(b) Blow Dry Hair and Hair Cut

(c) Mopping Floor and Baby Crawling

(d) Running and Jogging

(e) Golf and Pick

(f) Turn and Walk

Figure 1.2: Similar gesture human actions

1. In public place surveillance, this study of human action recognition help the security sectors to evaluate the abnormal actions and reduce reaction time when bad thing happens.
2. Baby room monitoring provides a save place while monitoring the baby actions, which send real-time warning to parents by monitoring actions of the baby.
3. Human machine interaction gives the robots the ability to recognize human actions by cameras rather than other sensors.
4. Video indexing can help people to find different videos based on different topics which can be categorized by different actions.

1.3 Challenges

In early stages, researchers made assumptions on certain scale or fixed viewpoint when the video was captured. However, those assumptions doesn't reflect the real-world environment. Besides, early research also followed the two-steps approach to design the system. First, the hand-craft features are extracted from the video frames, followed by the design of classifiers based on the extracted features. Thus, most of the early research works calculate the motion and texture descriptors using spatio-temporal interest points which are built manually. In the real-world scenario, the performance of these hand-crafted features is low as they are highly problem-depended and lacks generalization. Especially, for human action recognition, different actions may correspond to totally different patterns due to the environment changes and motion patterns. However, they overlook one truth, unlike still objects, human actions in videos are the combination of the sequence of gestures, for some different actions contain the same gestures in most of the video frames. Hence the problem is apparent, feature-based methods tend to learn the features which extracted from the video frames; however, the neural network will generate almost the same features on the similar gesture actions. Some important challenges need to be addressed for the understanding of actions. I have discussed some of the challenges given below.

1. Camera motion will create different temporal features for human actions compare to the invariant camera.
2. Background cluttering is another challenge of human action recognition which will hard to detect human gestures from the complex background.

3. Actions are hard to detect during the occlusion happens.
4. Observing actions from different view-points can represent totally different gestures and features, which makes recognition process much harder than the single view-point.
5. Similar gestures makes the classifier confuse due to the captured features are very similar.
6. Deep learning approaches requires large amount of training data, however under certain circumstances the training data is not enough.

These similar gesture action challenges can be seen in the Fig 1.2.

1.4 Aims and objectives

The objectives of this research are to develop robust algorithms for human action recognition. In this study, Core technologies such as machine learning and computer vision will be applied to make intelligent solutions for video-based human action recognition. Substantial efforts have been made in human action recognition for different types of datasets. Furthermore, the similar gesture features and dataset shortage problem will be discussed. The aims of the stakeholder for video-based human action recognition are listed below.

1. The system should be able to classify similar gesture actions accurately.
2. The system should be an end-to-end system which requires no data pre-processing.
3. The system should be able to generate more data for augmentation to increase the bias and differences of the similar gesture actions.

To achieve the aims of the stakeholder, the following objectives have been proposed.

1. Developing a human action recognition framework for human action recognition using computer vision and deep learning approaches
2. To evaluate the proposed methods for similar gesture action recognition.
3. Design an end-to-end framework which can provide a one-stop procedure.
4. To study the data augmentation using generative adversarial nets.

1.5 Research questions

The above motivation and objectives lead to the following research questions.

1. How to identify the similar gestures and classify the actions from the videos?
2. Is it possible to process the similar gesture actions automatically in an end-to-end fashion?
3. How to increase the size of the datasets and bias between similar gesture classes?

1.6 Evaluation of action recognition methodology

There are several publicly available benchmark datasets for the evaluation of video-based human action recognition algorithms. The brief description of these datasets is given below.

1. KTH Dataset [107]
   KTH is one of the old dataset but still very challenging. It includes six actions. Each actions were performed by 25 different actors with four different backgrounds.
2. UCF101 Dataset [115]
   UCF101 contains 101 different actions which collected from Youtube which has 13320 realistic action videos in total with a large diversity regarding to different actions and their presence of variations in pose, object scale, object appearance, object scale, illumination conditions, cluttered background and camera motion etc.
3. HMDB51 Dataset [61]
   HMDB51 is another popular dataset which generated by Serre Lab from Brown University. HMDB51 is one of the large and generic available public dataset for real-world actions. The total 7000 video clips are collected from some of the commercial movies and Youtube.

1.7 Contributions

The proposed work has led to the following contributions in this thesis:

- A study of various human action recognition methods and their performance have been evaluated for different types of datasets.

- A new approach for similar gesture action recognition presented. The proposed generic hierarchical classification model can be applied to any datasets/real-world application involving gesture recognition.
- An end-to-end system has been proposed which can process the similar gestures automatically.
- An action data augmentation framework using GAN has been proposed, which can generate more training data and enlarge the differences between similar class.
- An graph embedding framework using the idea of GAN has been proposed, which apply the computer vision approaches to graph research area and achieved high performance in graph embedding.

1.8 Stakeholders

The thesis will assist the different applications which require to apply the human action recognition efficiently and accurately.

1.9 Outline of the thesis

Based on the proposed research questions, the proposed methodologies are divided across several chapters. The thesis is divided into seven chapters and is briefly described as follow:

- Chapter 1 provides an introduction to the research area, aims, and motivation of the research, its significance, major challenges, brief description of the datasets and evaluation criteria, and outline the contributions to the research.
- Chapter 2 discuss the background and related works in the human action recognition using computer vision and machine learning approaches. The limitations and open research questions is discussed as well.
- Chapter 3 presents the hierarchical classification framework designed for similar gesture action recognition as well as extend the framework into the end-to-end system which can process the similar gesture classification automatically.

- Chapter 4 presents the developed action data augmentation framework with a GAN features generator and evaluate the performance for classification on different datasets.
- Chapter 5 presents the feature-dependent graph convolutional autoencoders with adversarial training methods, which can embed and reconstruct the both node features and structural characteristics.
- Chapter 6 Summarizes the thesis and outline the future research work in the human action recognition.

CHAPTER 2

BACKGROUND AND RELATED WORKS

This chapter provides a brief review of current state-of-the-art approaches and their limitations. The discussion about limitation and shortcomings will indicates research direction and will lay the foundation for the work discussed in subsequent chapters.

2.1 Introduction

Video-based human action recognition has become one of the most popular research areas in the field of computer vision and pattern recognition in recent years. There are many challenges involved in human action recognition in videos, such as cluttered backgrounds, occlusions, viewpoint variation, execution rate, and camera motion. A large number of techniques have been proposed to address the challenges over the decades. From a computer vision perspective, the aim of analysis is to find the solutions based on different types of datasets and methods.

The problems presented by human action recognition are more complicated than a typical computer vision system problem. Here we will discuss some of the common circumstances faced by typical computer systems as well as by the system used for video-based human action recognition.

1. **Single-view points:** The single viewpoint datasets normally use a single camera recording human actions from a certain invariant angle without camera movement as illustrated in Fig 2.1.

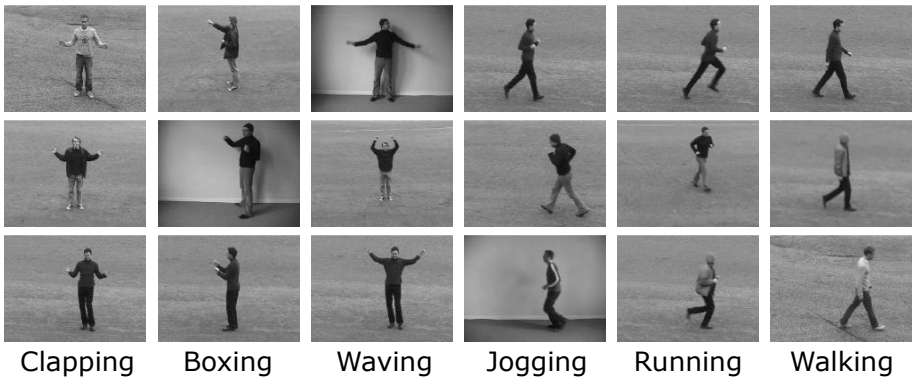


Figure 2.1: Samples of single-view point dataset[146]

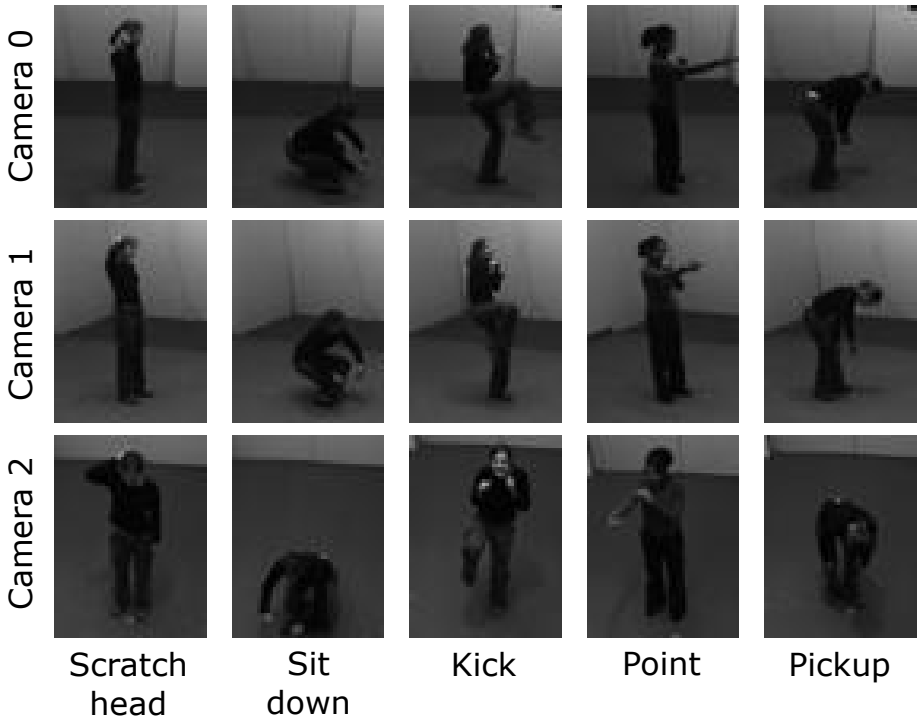


Figure 2.2: Samples of multi-view point dataset[146]

2. **Multi-view points:** In a real-world scenario, multiple cameras are used for monitoring large public spaces, such as shopping malls, airports, trains and bus stations. Some multi-view datasets have been created specifically for studying the problem of processing multiple views of the same human as illustrated in Fig 2.2.

This literature review aims to do a critical analysis of current state-of-the-art techniques. In particular, our survey will be closely related to gesture estimation, feature

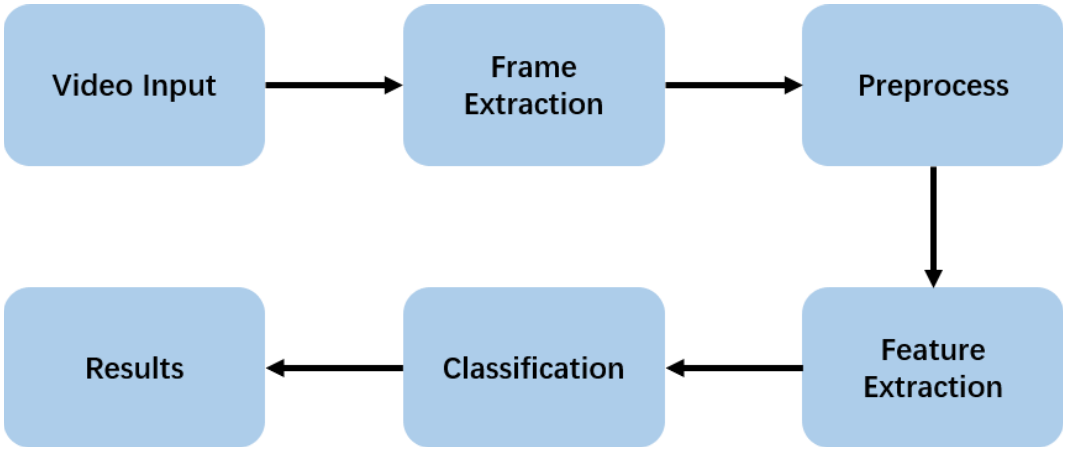


Figure 2.3: A typical video-based human action recognition system.

extraction, gesture recognition, and action recognition. And we broadly classify the methods into two categories: feature-based and deep learning-based.

In recent years, much work has been done in different areas in the computer vision research area, such as video classification [77], resolution [52] and segmentation [60] etc. However, the research on video-based human activity recognition has not been explored much, due to the challenges in processing temporal information from the video stream. Action recognition from a video stream can be defined as recognizing human actions automatically using a pattern recognition system with minimal human-computer interaction. Typically, Fig 2.3 shows an action recognition system which analyzes certain video sequences or frames to learn the patterns of a particular human action in the training process and use the learnt knowledge to classify similar actions during the testing phase [10, 18, 26, 35, 37, 43, 49, 51, 54, 65, 71, 80, 86, 100, 151, 154]. Among the early approaches [17, 28, 36, 67, 70] for human action recognition, all of these investigations use motion and texture descriptors calculated based on the spatio-temporal interest points, which are built manually. Subsequently, they compute features from raw video frames and classifiers are trained based on the features obtained. Thus, even the features can be fully extracted automatically, and these hand-crafted features are used for specific problems. Therefore, the main drawback of these approaches is that they are problem-dependent, which is challenging to apply in the real-world, even though they may achieve high performance in action recognition.

Over the past decades, deep learning methods take the place in computer vision and human action recognition research area, which have achieved tremendous success. Convolution neural networks (CNN) and Recurrent neural networks (RNN) have been

widely used to process the images and recognize the human action by extracting and learning the features from the video frames automatically. A two-step framework learn the features from the video frames by using CNN and RNN are proposed by [6], however, the proposed method ignore the feature correlation between frames when the frame sequential information has been learnt by the RNN. [53] introduced 3DCNN to address this problem, which can preserve the features on the same pixel spots between adjacency frames, by convolving the features on sequential frames. However, the 3D convolution only can preserve partial temporal information by convolving the changing features on the same pixels to the next level. Recurrent neural networks (RNN) and long short term memory (LSTM) models have shown great achievements in many time series methods such as natural language processing (NLP) tasks, these time series models perform the same task on each of the sequential elements. LRCNs and HM-AN are the recent RNN works which process the temporal features inside the videos are proposed by [29] and [152]. In these methods, the new temporal features which extracted from the frames will be used to update the hidden states and the previous hidden states will be forgotten. However, the temporal features are handcraft features or extracted from the CNN. Hence, the similarity of the extracted features would reduce the performance of the RNN. To recognize human actions accurately, many blend methods have been proposed. [112] proposed a two-stream convolutional networks for video-based human action recognition which blend the results from spatial and temporal channels in the last layer of the network. The two channels simultaneously process the information, and many other works followed this idea by modifying the networks [74] or choose the different features [33]. Nonetheless, two-stream methods still cannot solve the similar feature problem, because they are feature dependent methods.

2.2 Datasets

With the development of human action recognition technology, many different types of datasets have been prepared and released recently. These datasets are widely used for experimental purposes to evaluate the performance and accuracy of existing/new approaches and to ensure appropriate comparison with other approaches. Generally, deep learning can be applied to different types of datasets with raw input data. In addition, the complexity of the networks may be determined by the different types of the datasets. For example, single viewpoint data may require less steps than multiple viewpoint data, which needs to generate multiple networks to obtain the final output. Therefore, we

classify the datasets as single viewpoint, multiple viewpoints videos. These datasets offer dedicated features for different research purposes, such as gestures, 3D body modeling and joints etc. In this section, we review the popular public datasets on which deep learning techniques have been successfully applied. Table 2.1 lists the various datasets which are popularly used for research.

Table 2.1: Comparison of the human action recognition datasets

Datasets	Type	Views	Subjects	Year
KTH [107]	single-view	1	6	2004
Weizmann [9]	single-view	1	10	2005
UCF sports [114]	single-view	1	150	2008
Hollywood [67]	single-view	1	8	2008
Hollywood2 [76]	single-view	1	12	2009
Olympic Sports [82]	single-view	1	16	2010
HMDB51 [61]	single-view	1	51	2011
UCF50 [103]	single-view	1	50	2012
UCF101 [115]	single-view	1	101	2012
Kinetics [56]	single-view	1	400/600	2017
IXMAS [141]	multi-view	5	14	2006
CASIA Action [138]	multi-view	3	8	2007
i3DPost [39]	multi-view	8	12	2009
MuHAVi [113]	multi-view	8	17	2010
Videoweb [25]	multi-view	4-8	51	2011

2.2.1 Single-view point datasets

The single viewpoint datasets normally use a single camera recording human actions from a certain invariant angle without camera movement. These datasets were used for the analysis of human actions in the early stage of research, as shown in Fig 2.1. The earliest single viewpoint dataset was released in 2001 by Weizmann Institute [9]. This dataset recorded ten actions and each action was performed by ten persons. The foreground silhouettes are included in the dataset and the backgrounds are static as the viewpoints are static. In 2004, another dataset named KTH [107] was published. The KTH dataset contains six actions with four different scenarios, performed by twenty five actors. Similar to the Weizmann dataset, the backgrounds are static as well, except in the zooming scenarios. These early datasets have some drawbacks, such as videos are recorded in constrained environments and the actors perform simple identical actions in the video clips which are not the representative of human actions in the real world. To

consider real scenarios, several other datasets were introduced, including UCF sports [114] and Hollywood datasets [67] which are extracted from YouTube or from movies. The UCF sports dataset contains 150 sports motions considering human appearance, camera movement, viewpoint change, illumination and background. The Hollywood dataset proposes eight actions to address the challenges of occlusions, camera movements and dynamic backgrounds. These datasets have a fixed viewpoint to monitor the actions in the video stream. UCF101 is one of the state-of-the-art dataset which proposed by the by Center for Research in Computer Vision in the University of Central Florida. UCF101 contains 101 different actions which collected from Youtube which has 13320 realistic action videos in total with a large diversity regarding to different actions and their presence of variations in pose, object scale, object appearance, object scale, illumination conditions, cluttered background and camera motion etc. All the 101 action classes are categorized by 25 groups, each group has four to seven videos for one of the action. The videos from same group share the common features, for example, similar viewpoint or similar background. Thus, classifying the video contains similar gesture actions could be a challenge. HMDB51 [61] is another popular dataset which generated by Serre Lab from Brown University. HMDB51 is one of the large and generic available public dataset for real-world actions which firstly published on ICCV 2011. The total 7000 video clips are collected from some of the commercial movies and Youtube, and the video clips contain 51 human actions such as some facial actions, body movements and body and objects interactions and each class contain around 100 videos. It is a very challenging dataset because the many actions have similar gestures performed by different person. Each action would be captured through different viewpoints and recorded in four to six video clips. Kinetics [56] is a state-of-the-art dataset which published in 2017. Kinetics consists of approximately 500,000 video clips, and covers 600 human action classes with at least 600 video clips for each action class. Each clip lasts around 10 seconds and is labeled with a single class.

2.2.2 Multi-view point datasets

In a real-world scenario, multiple cameras are used for monitoring large public spaces, such as shopping malls, airports, trains and bus stations. Some multi-view datasets have been created specifically for studying the problem of processing multiple views of the same human. The advantages of these datasets is that they model a 3D human body shape from different angles and occlusion problems are avoided in contrast with single viewpoint streams.

Weinland et al. [141] released the IXMAS dataset which contains 14 actions performed by 11 persons. For each action, there are five cameras capturing the action from five angles with a static background and illumination settings. Sample images taken from the IXMAS dataset are shown in Fig 2.2, where multiple views of the same human actions are captured by different cameras placed at different viewpoints. Another indoor dataset, the i3DPost Multi-view dataset [39] was published in 2009. Eight high definition cameras were used to capture twelve actions performed by eight persons. Kingston University released their dataset in 2010 which was called MuHAVi [113]. They used eight non-synchronized cameras to capture 17 actions performed by 14 actors and it was designed to test different action recognition algorithms. Unlike the indoor datasets with static backgrounds, several datasets captured actions under real conditions, such as Videoweb [25] and the CASIA Action datasets [138]. In the Videoweb dataset, four groups of actors perform actions, which were captured by four to eight cameras tailored for group activity recognition. The CASIA Action dataset mainly focuses on interactions between persons and it contains eight types of single person actions performed by 24 people and seven types of interactions captured by three static cameras from different angles. These multi-view datasets can provide multiple streams as inputs for researchers.

2.3 Human action recognition approaches

In order to recognize high-level activities hierarchically, the multi-layered Hidden Markov Model (HMM) was introduced in the early stages of human action recognition research. Most HMM-based work has been performed on single viewpoint datasets. A fundamental form of the multi-layer approach was presented by Oliver et al. [85]. At the lower level, HMMs were used to recognize various sub-events, such as stretching and withdrawing. The upper level treats the result from the lower level as input and recognizes the punching activity when stretching and withdrawing occurred in a certain sequence. However, by nature, HMMs require strict sequences in each layer. Therefore, HMM approaches may not be able to meet the expectations of processing speed and system performance. This section focuses on the use of deep learning techniques with raw input data used by researchers for human action recognition on three types of video datasets. Since the approaches proposed were on different datasets and testing strategies, it is difficult to make a quantitative performance comparison. Even deep learning is still relatively new in this research area, however, it is crucial that these approaches have the ability to undertake high-level action recognition with high performance.

2.3.1 Hand-crafted feature methods

In the early stages of action recognition research, the techniques were based on optical flow [32, 113], tracking [2, 101, 102, 109] and a spatio-temporal shape template [9, 57, 155]. The computation of optical flow helps to construct action templates for flow and tracking-based approaches. However, at the boundary of the segmented human body, the features are more sensitive to noise, which are extracted from the flow templates. The action recognition problem is treated as 3D object recognition by spatio-temporal shape template approaches. These approaches require the extraction of highly detailed silhouettes, which may not be possible when there is real-world noisy video input. Further, a recognition rate with 100% accuracy has been demonstrated on the WEIZMAN dataset, however, these approaches do not work properly on a dataset which contains noise such as the KTH dataset. The KTH dataset contains noises such as low resolution, zooming, and camera movement, which makes it impossible to extract a clean silhouette. The spatio-temporal interest point-based approaches have become increasingly popular to address this problem. Further, the 2D SIFT descriptors [72] are extended to 3D with the addition of dimension to the histogram orientation by Scovanner et al. [108]. Due to the encoded temporal information, the extended 3D descriptors perform better than the 2D descriptors in action recognition. Furthermore, Willems et al. [142] proposed the spatio-temporal domain which is an extension of the SURF descriptor. Schuldt et al. [66] and Dollar et al. [28] described sparse spatio-temporal features to deal with the complexity of human action recognition [31, 113]. Schuldt et al.[66] proposed the representation of action using 3D spatio-temporal interest points captured from video frames. Schuldt also produced a histogram of informative words for each action adopting the codebook and bag-of-words (BOW) approach.

A dictionary of prototypes or video-words can be formed based on the clustering of the detected points of interest. Similarly, Dollar et al. [28] introduced a multi-dimensional linear filter detector which is able to detect denser points of interest. The BOW approach was applied but it took sparser sampling of the points of interest. Niebles and Fei-Fei [83] introduced a hierarchical model which can be characterized as a constellation of bags-of-features to improve the performance. The approaches [28, 83] represent BOW features, which are adopted successfully for 2D object categorization and recognition. The BOW features are robust against noises, camera movements and low resolution datasets compared with object tracking and shape-based approaches. Moreover, these approaches mainly focus on individual local space time descriptors rather than global space time descriptors.

Trajectory features are widely used at the beginning of the action recognition research. The trajectory features such as dense trajectory which represents the spatio-temporal can be through detectors or descriptors [131] [128] [130]. Besides, the fisher vector (FV) coding model is another way to extract spatio-temporal local features. Approaches such as discriminative dictionary learning (MIDDL) [69] and Gaussian Mixture Model (GMM) [50] have been proposed to process the FV. Furthermore, some methods learn the human actions by generating histogram features such as the bag of visual words model (BoVW) [95], dictionary learning [150] and universal movement model (UMM) [104]. However, those hand-crafted feature methods are designed for certain problems specifically, which are not universally applicable.

However, the early work did not consider noise. In recent years, researchers have applied different new methods to tackle the challenges from noise in the human action recognition area, such as camera in-variation, camera motion and occlusion. Most of the early work assumes that the action is captured from a static viewpoint without any camera movement. However, the patterns of human actions appear to be different from different angles. A person‚Äôs gestures and their location vary according to each camera angle. Some of the approaches train a single classifier for all viewpoints or a set of classifiers where each classifier deals with one viewpoint [35, 51]. However, these approaches only extend the system from a single viewpoint to a multi-view dataset. Therefore, the performance only depends on the extracted features and the trained classifiers. Lu et al. [73] introduced motion history and motion energy images to observe the additional action features in the images. This approach may disrupt the background of the image especially if there is more than one person in the image. In order to obtain accurate multi-view action representations, researchers proposed some models to generate 3D or 2D body gestures through the multi-view datasets. The human body can be distinguished into several parts, and action recognition depends on the features extracted from the different body parts. Kumar and Madhavi [63] used an envelope shape to represent the human body and model the action recognition classifier.

2.3.2 CNN and RNN methods

Due to the recent success of the deep learning on human action recognition, CNN and RNN methods draw the attention of the researchers, many methods based on CNN and RNN have been designed. However, CNN was specifically designed for still images, researchers firstly learn the temporal information by integrating hand-craft features with CNN. For instance, Zhang et al. [159] leverage the CNN work with the linear dynamical

Table 2.2: Performance comparison of the human action recognition approaches

Methods	Datasets	Performance (%)	Year
[6]	KTH	94.39	2011
[53]	KTH	90.02	2013
[42]	KTH	90.7	2013
[112]	UCF101	88.0	2014
	HMDB51	59.4	
[111]	Weizmann	98.63	2014
	KTH	92.3	
[3]	KTH	94.3	2014
[117]	UCF101	88.1	2015
	HMDB51	59.1	
[124]	KTH	93.96	2015
[110]	KTH	95.60	2015
	UCF50	95.24	
[132]	UCF101	95.1	2015
	HMDB51	65.9	
[7]	UCF101	80.7	2015
[133]	UCF101	88.3	2016
	HMDB51	61.7	
[33]	UCF101	92.5	2016
	HMDB51	65.4	
[94]	UCF101	78.86	2016
	Olympic Sports	94.8	
[129]	UCF101	86.0	2016
	HMDB51	60.1	
	UCF50	91.7	
	Olympic Sports	90.4	
	Hollywood2	66.8	
[14]	UCF101	97.9	2017
	HMDB51	80.2	
[161]	UCF101	95.8	2017
	HMDB51	74.8	
[64]	UCF101	95.3	2017
	HMDB51	75.0	
[27]	UCF101	93.2	2017
	HMDB51	63.5	
[81]	Olympic Sports	94.0	2018
	Hollywood2	68.1	
[44]	Kinetics	78.4	2018
	UCF101	94.5	
	HMDB51	70.2	
[46]	Kinetics	78.99	2018
	UCF101	95.7	

system (LDS) to learn spatio-temporal features in videos. In addition, Banerjee and Murino [8] use an efficient pooling strategy to incorporate image based deep features. Factorized spatio-temporal CNNs [117] were designed to handle the spatial and temporal kernels in different layers which could reduce the number of learning parameters of the network. With the transformation and permutation operator, a training and inference strategy along with a sparsity concentration index scheme produced the final result, which outperformed existing CNN-based methods. Another work [6] shared a similar idea, the only difference being that they extracted the spatial and temporal information as a single frame and a multi-frame optical flow. This spatial and temporal information was fed into a spatial and temporal stream CNN, respectively. Ballas et al. [7] used a convolutional GRU-RNN (GRU-RCN) to process the visualization of convolutional maps on successive frames in a video. The results show that the Bi-Directional GRU-RCN Encoder outperforms the VGG-16 Encoder by 3.4% and 10% for action recognition compared to both RGB and Flow inputs, respectively.

Long Short Term Memory (LSTM), a variation of RNN, also received increasing attention in sequence processing. LSTMs use memory blocks to replace the regular network units. The gate neurons of the LSTM determine when it should remember, forget or output the value. It was previously used to recognize speech and handwriting. A robust LSTM [42] with recurrent cell connections was tested for action recognition to show that classification accuracy may be affected by training set size, length of the video sequence and quality of the video. Veeriah et al. [124] delivered a different gating scheme to address the problem of conventional LSTMs which emphasizes the change in information gain caused by the salient motions between successive frames. Then, the LSTM model was termed as differential RNN.

To involve the temporal features in the CNN, Baccouche et al. [6] introduced a two-step neural network-based deep learning model. The first step uses CNNs to learn the spatio-temporal features automatically and the following step uses a Recurrent Neural Network (RNN) to classify the sequence. In addition, Ji et al. [53] proposed a 3DCNN architecture, they preprocess the video data into multi-channels such as grey, gradient and optical flow by the hardwired layer as the input, and the features will be extracted between adjacent frames. Tran et al. [121] took the 3DCNN one step further by replacing all the kernels with 3DCNN kernels and built a new VGG-style 3DCNN network, namely C3D. Varol et al. [123] proposed a long-term temporal convolution (LTC) networks. It requires 60 to 100 frames, which can generate high-quality optical flow as the input. To learn the temporal features between frames, Donahue et al. [29] fuse the CNN and

long short term memory (LSTM) which is a variation of RNN. Yan et al. [152] proposed hierarchical multi-scale attention network (HM-AN) by using attention mechanism with RNN. Besides, the hierarchical recurrent neural encoder (HRNE) [88] can exploit video temporal structure and model the temporal transitions between frames as well as the transitions between segments. However, the approaches are not sensitive to similar features.

Unlike other neural networks, Spiking Neural Networks (SNNs) work similarly to their biological counterparts. A special model based on SNNs was designed by Shu et al. [111], which is a hierarchical architecture of the feed-forward spiking neural networks modeling two visual cortical areas: primary visual cortex (VI) and middle temporal area (MT), neurobiologically dedicated to motion processing. It simulates the working mechanism from the VI and MT. After detecting the motion energy, the information is processed by the VI layer and MT layer. The motion energy is first transformed by the spiking neuron model in the VI layer, then the MT cell pools the information received from the VI cell according to the mapping connection between the two layers. Features are extracted from the spike trains which are generated by MT spiking neurons. The final output is recognized by an SVM classifier. Ali and Wang [3] built a Deep Brief Network (DBN) which is another variant of deep neural networks. It is composed of multiple hidden unit layers with connections between the layers to the learning feature for action recognition.

Some of the methods prefer to extract different descriptors as input before using deep learning techniques. In [110], researchers firstly extract dense trajectories from raw data with multiple consecutive frames and then project the trajectories onto a canvas. In this way, they can transfer the raw 3D space into a 2D space and import them, hence, the complexity of the data is reduced. Subsequently, they input the data into a Deep Neural Network (DNN) which is utilized to learn a more macroscopical representation of dense trajectories. Some additional features are extracted and used as inputs to the classifier. Wang et al. [132] claimed that their trajectory-pooled deep-convolutional descriptor (TDD) outperformed the hand- crafted features with higher discriminative capacity. A posed-based CNN [19] descriptor was used for action recognition which was generated based on human poses. The input data was divided into five part patches. For each patch, two kinds of frames were extracted from the video, namely RGB and flow frames. The P-CNN features are generated by both frames and processed in the CNN, respectively after aggregation and normalization stages. Table 2.2 presents a comparative study of different single/multiple view approaches.

2.3.3 Two-stream convolutional methods

Simonyan and Zisserman [112] proposed the first two-stream convolutional architecture. The advantage of the two-stream methods is the architecture can process the spatial and temporal information simultaneously, which can achieve comparable performance despite the limited training data. Hence two-stream convolutional became the most popular and effective approach for action recognition. Ng et al. [156] fed both spatial and temporal features extracted from optical flow into the LSTM. The two-stream convolutional architecture also can feed hand-craft features, Wang et al. [135] combine the appearance and motion information via a feature concatenation layer. Feichtenhofer et al. [33] fuse spatio-temporal features via 3DCNN kernels and 3D pooling. Other works are replacing the stream from classic CNN to other networks such as temporal segment network (TSN) [134], two-stream semantic region based CNNs (SR-CNNs) [139], key volume mining deep framework [162] and two-stream Siamese network [137]. However, the performance of the two-stream convolutional architectures still depends on the input features, which are not specifically designed for similar gesture actions.

2.3.4 Deep neural networks

By the development of the human action recognition approaches, the layer of the neural networks has became deeper and deeper compare with early shallow networks. One of the success deep neural network is residual neural network (ResNet) which proposed by [47]. Unlike standard CNN, ResNet is utilizing skip connections, or short-cuts to jump over some layers. Typical ResNet models are implemented with single-layer skips. He et al. [46] leverage the both local and global spatial-temporal modeling in videos and a novel temporal Xception block has been added into the ResNet. In addition, Hara et al. [44] apply the spatio-temporal three-dimensional kernels on ResNet with different number of layers which from ResNet-18 to ResNet-101. The results shows that the deeper of the neural network will achieve higher performance. Furthermore, the Resnet approaches can transfer the classification from large dataset like Kinetics to small datasets such as UCF101 and HMDB51.

2.4 Limitations and open research problems

Based on the review of related work available in the literature, the following limitations and open research problems are listed below.

- The camera motion is challenging because the temporal features will be different compare to the invariant camera. Not much work has been reported in the literature. Considering surveillance scenario, robust techniques which work irrespective of variant camera are desired.
- Occlusion is another challenging problem, however, existing works assume that there is no occlusion in the existing public datasets. Where robust action prediction techniques is required when occlusion happens.
- Deep learning approaches requires large amount of training data, however under certain circumstances the training data is not enough. However, many works only focus on the classification performance rather than considering the cost.
- The reported methods achieved high performance on the global classification, however the gap still exists. Not much work discover the reason why the mis-classification happens.

The purpose of this thesis is to discover the reason of mis-classification and how to further improve the performance by solving the problems.

2.5 Preliminary study

An issue was discovered after analyzing the results of the literature, which is even methods such as [21] can achieve high global accuracy, mis-classification happens between some specific classes, which could be the reason of the limited global accuracy.

Chou et al. [21] captures sequence motions and occlusions at a low computational cost due the detection of the points of interest and apply the nearest neighbor classifier (NNC), Gaussian mixture model classifier (GMMC) and the nearest mean classifier (NMC) to do the classification. The results achieve 89.31%, 90.22% and 90.58% on KTH dataset using NNC, GMMC and NMC respectively. Confusion matrix reported in their paper are shown in Fig 2.4(a), Fig 2.4(b) and Fig 2.4(c).

According to the confusion matrix, some of the classes achieved high accuracy like "Boxing", "Hand clapping" and "Hand waving". However, the confusion rates are high between the class "Jogging", "Running" and "Walking". In NNC method, 12% "Jogging" are mis-classified as "Running" and 7% "Jogging" are mis-classified as "Walking". Similar as NNC, in GMMC 15% "Running" are mis-classified as "Jogging" and in NMC 10% are "Jogging" mis-classified as "Jogging".

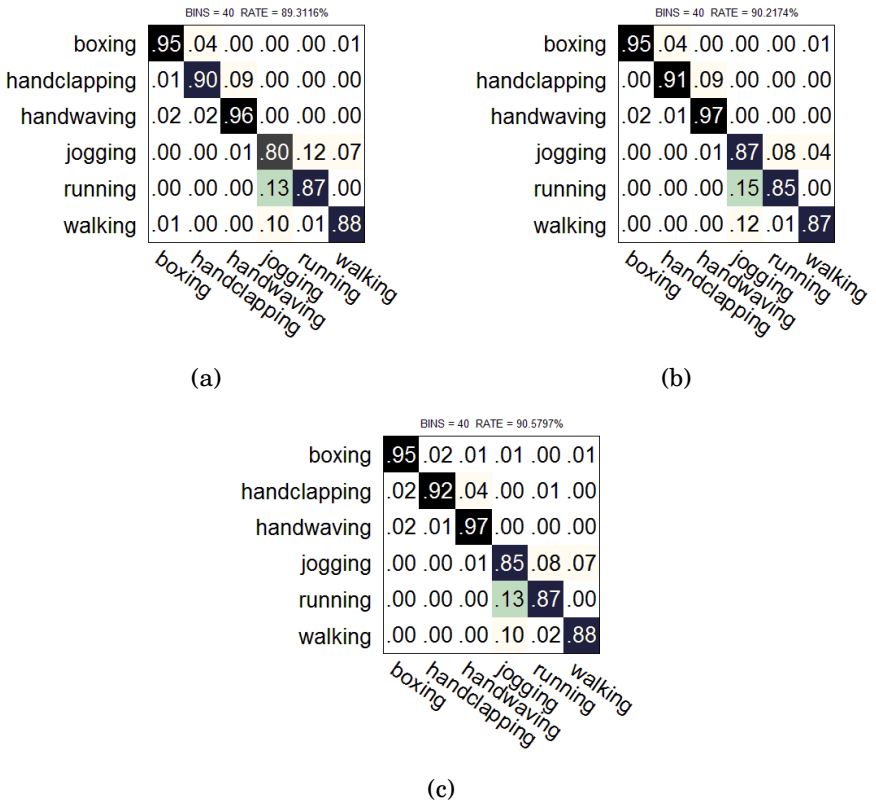


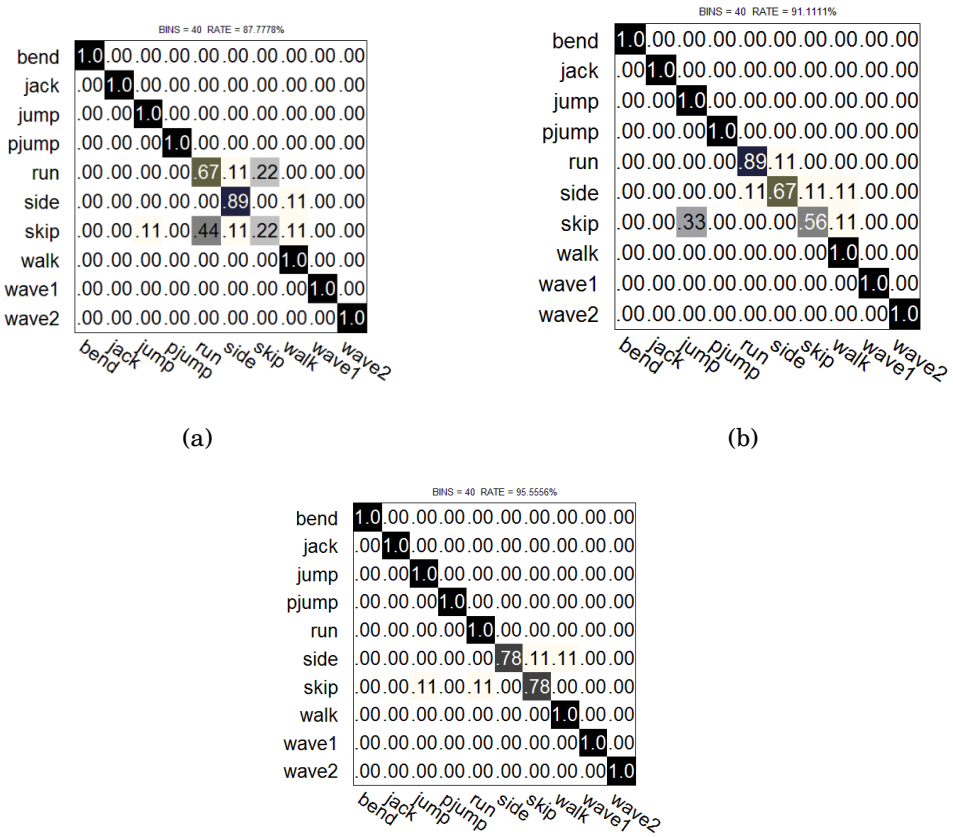
Figure 2.4: Confusion matrix proposed by Chou et al. [21] on KTH for (a) NNC, (b) GMMC and (c) NMC

Same issue happens in Weizmann [9] dataset which reported in Chou et al. [21]. The confusion matrix reported in their paper are shown in Fig 2.5.

In the results of Weizmann dataset, the results achieve 87.78%, 91.11% and 95.56% using NNC, GMMC and NMC respectively. However, in all the three algorithms, classes such as "Run", "Side" and "Skip" have the high confusion rates. For example, in NNC, the accuracy of "Skip" is only 22% because 44% and 11% of "Skip" are mis-classified into "Run" and "Side".

For further investigate this issue, the mis-classification happens between specific classes has been analyzed. Fig 2.6 shows the three confused classes "Jogging", "Running" and "Walking" in KTH dataset.

From the extracted frames of KTH dataset, "Jogging", "Running" and "Walking" actions have the similar gestures which will supply very similar features for the classifier even human eye cannot distinguish the difference if we do not label the classes. These similar features will confuse the classifier during the training stage to reduce the performance on single classes as well as the global classification performance.



(a) (b)

(c)

Figure 2.5: Confusion matrix proposed by Chou et al. [21] on Weizmann for (a) NNC, (b) GMMC and (c) NMC

Table 2.3: Confusion rates of classes from multiple actions

Classes	Confusion Rates
Jogging	0.56
Running	0.31
Drink	0.19
Eat	0.16
Sit	0.25
Stand	0.26
FloorGymnastics	0.17
Rafting	0.08
Sword	0.09
Swordexercise	0.07
Turn	0.15
Walk	0.12

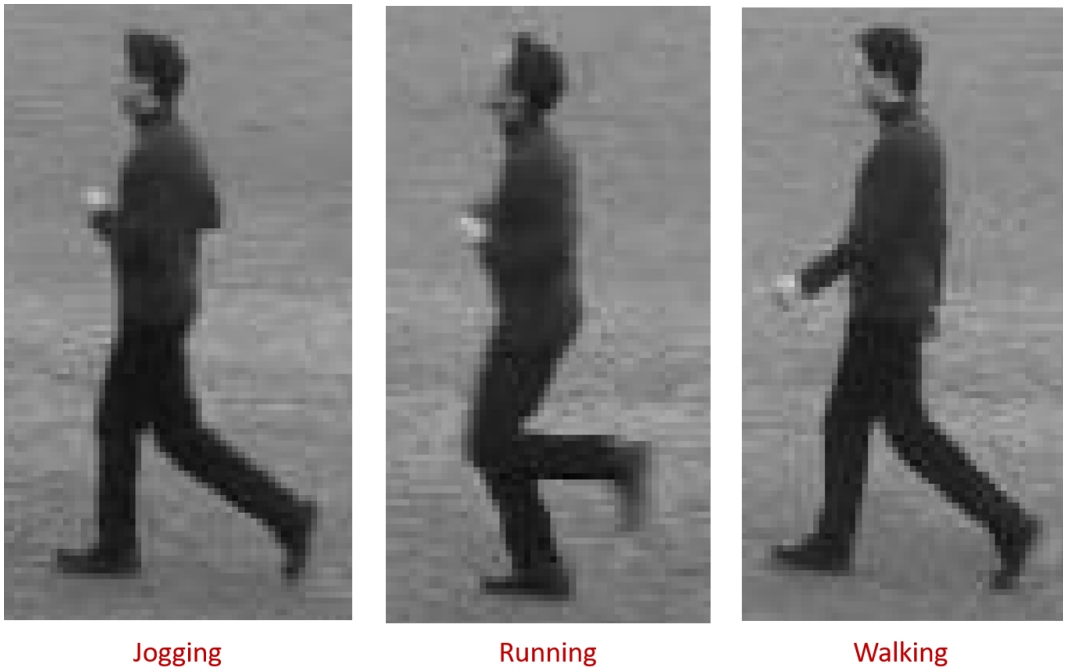


Figure 2.6: "Jogging", "Running" and "Walking" in KTH dataset which have similar gestures

Some work argue that deep learning methods can improve the classification performance. Because comparing to 2DCNN, 3DCNN not only gathering the pixel features but also preserving the correlations between frames. To resolve this question, the experiments have been conducted on KTH and HMDB51 datasets for 3DCNN methods, the results shows that the 3DCNN is not sensitive to the similar gestures as well as shown in Table 2.3

This thesis will tackle the similar gesture action recognition problems and propose the solutions for similar gesture action recognition to improve the performance of the global classification.

2.6 Summary

In this chapter, a comprehensive literature review of the different aspects of human action recognition was presented. This literature review shows that most of the work has been done in the area of human action recognition using feature-based and deep learning-based methods. This chapter also presented techniques mainly focusing on developments in deep learning over the past ten years. Many investigations have been conducted to deal

with different types of datasets. For single/multiple viewpoint approaches, the inputs are normally frames, so researchers have performed 3D convolution operations to add the temporal information in order to recognize videos. Additionally some of the approaches could also be used to generate features for different classifiers. Table 2.2 shows some of the performance comparison between different approaches. As it is evident from literature, most early traditional machine learning works are problem dependent, which apply the texture descriptors on the extracted handcraft motion features. In addition, all the reviewed methods overlook one factor, unlike still objects, human actions in videos are the combination of the sequence of gestures. Because of some different actions contain the same gestures in most of the video frames, the mis-classification happens. This chapter tackled the similar gesture action problem and then proposed hierarchical classification approach and data augmentation framework for similar gesture action recognition in this thesis.

CHAPTER 3

SIMILAR GESTURE ACTION RECOGNITION

This chapter is divided into two parts. The focus of part 1 is to explore similar gesture problems in existing human action recognition and propose a hierarchical classification approach to improve the classification performance for both similar gesture class pairs and global classification results.

Part 2 of this chapter proposes an end-to-end system which can process similar gesture videos automatically. In addition, experiments were conducted on multiple datasets to evaluate the performance.

The major parts of this chapter have been published in the paper titled "Similar Gesture Recognition using Hierarchical Classification Approach in RGB Videos" by Wu et al. [148] and in the paper titled "An End-to-End Hierarchical Classification Approach for Similar Gesture Recognition" by Wu et al. [147].

3.1 Introduction

Human action recognition is one of the most popular research areas in computer vision. Diverse applications have been designed based on human action recognition technology such as, surveillance, video indexing, human-computer interaction, customer behaviour monitoring and analysis, etc. across multiple domains. However, recognizing human actions accurately from a video stream is a challenging task due to occlusion, low resolution, cluttered backgrounds and viewpoint variations, etc. [77] [55] [60]. Unlike action recognition from still images, videos include temporal information and genetic data

augmentation which is essential to classify actions/gestures more accurately. In the early stages, researchers made assumptions on certain scales or fixed viewpoints when a video was captured. However, these assumptions do not reflect the real-world environment. Furthermore, early research also followed the two-steps approach to design the system. First, the hand-craft features are extracted from the video frames, followed by the design of the classifiers based on the extracted features. Thus, most of the early research works calculate the motion and texture descriptors using spatio-temporal interest points which are built manually. In a real-world scenario, the performance of these hand-crafted features is low as they are highly problem-dependent and lack generalization. In particular, for human action recognition, different actions may correspond to totally different patterns due to environmental changes and motion patterns.

Deep learning models [48] [143] [68] have become a priority choice to deal with computer vision problems due their impressive performance in various computer vision-related tasks. These models have the advantage of learning features from hierarchical neural network layers and automatically build the high-level representation from the raw video inputs. Hence, unlike traditional hand-crafted feature extraction methods, the CNN-based feature extraction and classification process is embedded in an end-to-end pipeline. In short, a deep learning model applies multiple techniques such as local perception, weight sharing, a multi-convolution kernel, down-pooling, etc. to study the features from the image or frames. The classifiers can be trained using either supervised or unsupervised methods, and the final result can be generated by ensembling the results of multiple network layers. Deep learning techniques are widely used in visual object detection and tracking [106], handwriting and signature recognition [84], natural language processing [22], human action recognition [33], and image segmentation [16], etc. Convolutional neural networks (CNNs) are one of the popular deep learning models in the computer vision research area. Convolutional neural networks are a type of deep model which include an input layer and an output layer. Between the two layers, there are multiple convolutional layers, pooling or sub-sampling layers, fully connected layers and normalization layers, which can be termed as hidden layers. Many research works have shown that, with a well-trained CNN model [78], the classifier can achieve high performance on object detection and recognition.

CNNs have been widely used for processing still images because of their ability in relation to feature construction through different deep layer models. In this chapter, the use of CNN models have been investigated on video-based human action recognition. A simple way to apply CNNs on videos is as follows. First, extract the frames from a

video. Then, treat each frame as an individual image and apply CNN models to recognize human actions at the image level. This strategy was used in early research to analyze human actions in videos [107]. However, the early work has serveral drawbacks, such as not taking into consideration the temporal and motion information in the video frames. To adequately address this problem, a 3DCNN architecture [53] was proposed by Ji et al. In the proposed method, the video is analyzed by multiple convolutional layers with 3D convolution and both the spatial and the temporal features are captured from three adjacent frames. Therefore, motion and temporal information can be analyzed simultaneously.

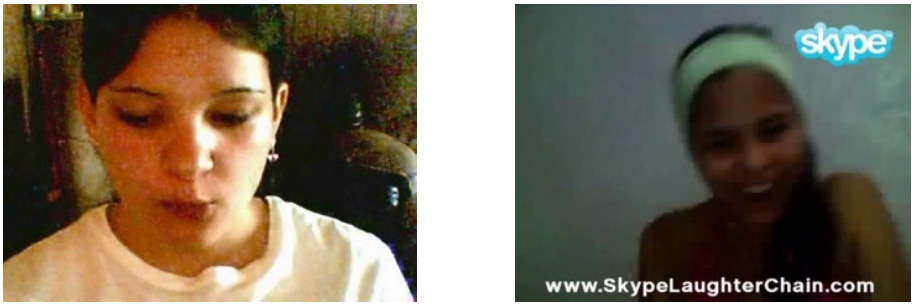
The 3DCNN approaches improved the performance of action recognition tasks. However, identifying human actions in videos is not as simple as static objects. With different actions, the body parts follow a different sequence of gestures listed in Figure 1. The gestures will be very similar in most of the video frames when people perform certain actions. For instance, playing golf is similar to picking up something, because in most frames, the subjects are bending over. This similarity is shown in Fig 3.1(a). Similar situations are "Swing and Throw"(Fig 3.1(b)), "Chew and Laugh" (Fig 3.1(c)) and "Turn and Walk" (Fig 3.1(d)). Hence, the drawback of CNNs in relation to videos is obvious, as CNNs will generate almost similar features for some of the actions with similar gestures.

Thus, the performance of the classifier is decreased by the misclassified classes. To analyze similar actions effectively and accurately, a hierarchical classification model has been proposed in which the first stage classifies multiple classes, whereas the second stage focuses on classifying similar gestures. Specifically, in the first stage, two of the confusing/similar gesture classes with highest confusion rates are merged to form a single class. Hence, the problem space for the first stage of classification is reduced to a lower number of classes and higher accuracy can be achieved. In the second stage of the classification, the merged pair of classes are handled explicitly and the problem space is reduced to binary classification. A binary classifier is applied to the respective merged pair of classes in order to resolve the similar gesture problem. The overall performance is measured by combining the first and second stages results.

Experiments using the proposed method have been conducted on the state-of-the-art human action datasets. The actions containing similar gestures (i.e., turn and walk, etc.) have been combined into single actions/classes as the input. The proposed system achieves high performance compared with the baseline CNN models. The experiment results also show that the developed hierarchical model outperforms other baseline models on similar actions.

(a) Similar gesture: Golf and Pick

(b) Similar gesture: Swing and Throw



(c) Similar gesture: Chew and Laugh

(d) Similar gesture: Turn and Walk

Figure 3.1: Human actions with similar gestures [148]

The major contributions of this work can be summarised as follows:

- The proposed method concentrates on the misclassification of similar gestures, instead of focusing on the whole dataset to solve the similar gesture problems in human action recognition.
- The results from the hierarchical 3DCNN architecture (H3DCNN) are combined to boost the performance of the final classification results by combining the global classification results and binary classification results.
- The evaluation of the proposed hierarchical model are conducted on the on the state-of-the-art datasets in comparison with the baseline CNN methods. The experiment results show that the proposed method outperforms other baseline methods on similar gesture actions, and also in relation to overall accuracy.

3.2 Methodology Part 1

3.2.1 Data preparation

Experiments were conducted on the HMDB51 dataset, which is a state-of-the-art dataset to evaluate the proposed architecture as shown in Fig 3.2. HMDB51 is a large and generic publicly available dataset for real-world actions collected by SERRE LAB from Brown University and was firstly released on ICCV 2011 [62]. The videos in this dataset were collected from YouTube and several movies which include a variety of actions with different human gestures including human body movements, body and object interactions and some facial actions. It contains 7000 video clips distributed across 51 action classes, in which each class has around 100 video clips. It is a challenging dataset because in the video clips of each class, a different person is performing the same gesture. Each subject performing the same action using different gestures and viewpoints has been recorded into 4 to 6 video clips. The proposed architecture is capable of handling the misclassified actions which involve similar gestures.

The most important process is how the similar gesture classes are merged to form a single class. To determine which classes to merge, two rules have been defined:

- Rule 1: Choose the classes with the highest misclassification rate, and
- Rule 2: Choose two classes which have similar gestures with highest confusion rate.

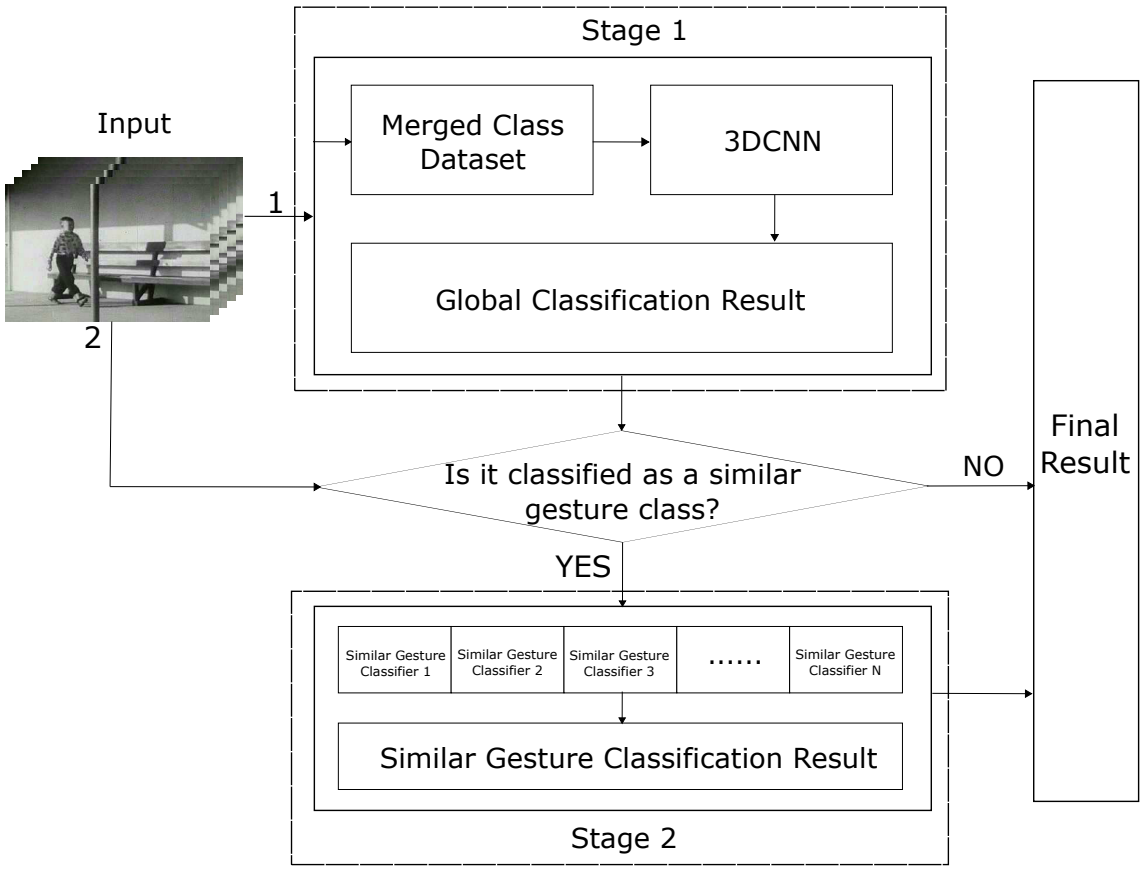


Figure 3.2: The proposed hierarchical 3DCNN architecture [148]

In order to identify the similar and confusing gesture classes, the overall performance of the state-of-the-art method [105] reported recently was considered. Table 3.1 provides details about the performance of the similar and most confusing gestures, and also provides information about the gesture pairs merged together to form a single class. Similar gesture actions such as, "Jump & Catch", "Pick Up Object & Golf", "Laugh & Chew" and "Sit & Stand" etc. are chosen and merged into one class as shown in Fig 3.3. After merging the classes, the number of classes in the complete dataset (HMDB51) will reduce from 51 classes to 42 classes. Moreover, the size/number of samples in the complete dataset remains the same. This process decreases the misclassification rate and improves the overall accuracy of the dataset, as the dataset now has unique gestures.

3.2.2 Architecture Description

Fig 3.2 presents the proposed hierarchical action recognition architecture which has two stages. The proposed architecture can be applied to different real-world scenarios for

Table 3.1: Merging the similar gesture classes

Classes	Accuracy reported in[105]	**Merged Classes**
Jump	**0.38**(low)	New Class 1
Catch	1.00	
Kick Ball	**0.31**(low)	New Class 2
Punch	0.51	
Laugh	0.41	New Class 3
Chew	0.47	
Pick	**0.27**(low)	New Class 4
Golf	1.00	
Sit	**0.39**(low)	New Class 5
Stand	**0.27**(low)	
Throw	**0.16**(low)	New Class 6
Swing Baseball	**0.16**(low)	
Turn	**0.222**(low)	New Class 7
Walk	**0.38**(low)	
Wave	**0.14**(low)	New Class 8
Shake Hands	0.82	
Sword	**0.13**(low)	New Class 9
Sword Exercise	0.42	

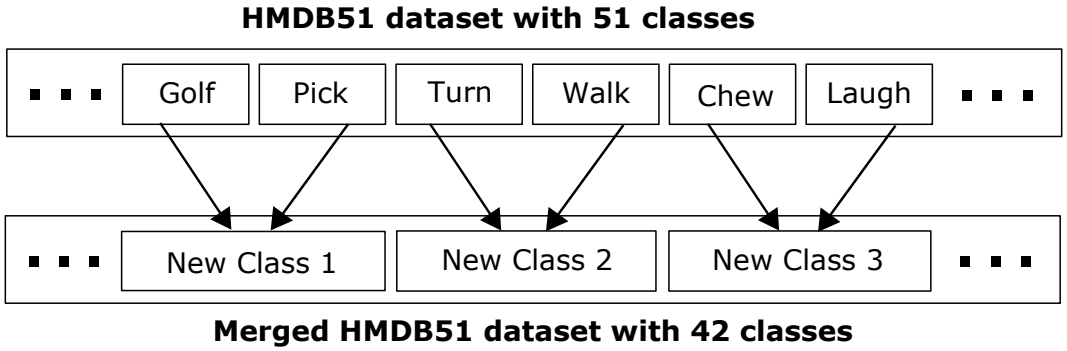


Figure 3.3: Merging class process [148]

gesture recognition. In this work, the HMDB51 dataset is used for the experiments and validating the proposed hierarchical architecture. The input data from the HMDB51 dataset has 51 classes initially. After merging the similar gesture class pairs based on the rules defined in the previous section, 42 classes are formed.

The first stage of the proposed hierarchical classification model focuses on classifying the generic classes (complete dataset), whereas the second stage resolves similar/confusing gesture problems. Once an input video is classified as one of the similar/confusing gesture classes in the first stage, the sample will be passed to the second stage for further classification. The second stage comprises different binary classifiers, each of the target binary classifier is responsible for one of the confusing gesture pairs, which is selected automatically based on the first stage classification results. Additionally, in the first stage, if a sample video is not classified as one of the similar gesture classes, the sample video will not be passed to the second stage and the predicted result from the first stage will be considered as the final result.

The final results are calculated by combining the global classification results from Stage 1 and similar gesture classification results from Stage 2. To obtain the final result of the dataset, the results of the similar gesture classes from the first stage will be replaced by the results from the binary classification in the second stage.

3.2.3 Experimental setup

Tensorflow [38] and Keras [20] frameworks were used to construct and train the neural networks. These frameworks are used to help researchers to design the neural network architectures and algorithms with GPUs. In the experimental setup, NVIDIA P6000 and CUDA 8 platform were used to complete the experiments. A 3DCNN was designed in Tensorflow and Keras for both global classification and binary similar gesture classification as shown in Figure 3.4, the kernel size of each 3DCNN layer is $3 \times 3 \times 3$ and the pooling size is 2×2. During the experiment, 60% and 30% of the whole dataset videos will be used as the training and testing sets, respectively, and the remaining 10% videos will be used as the validation set. The original RGB video frames were used as the input and the original classification results from the 3DCNN will be used as the benchmark to compare the performance between the original 3DCNN and the proposed hierarchical architecture.

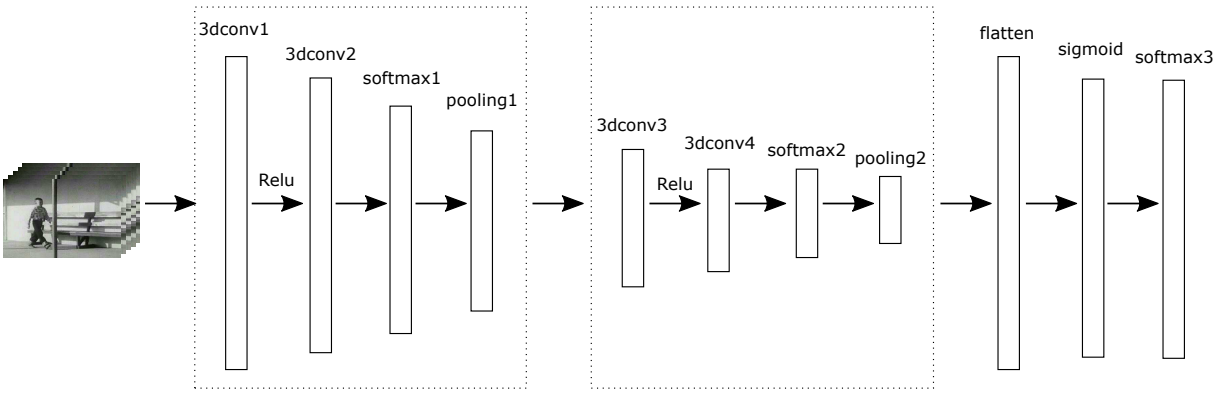


Figure 3.4: The architecture of the proposed 3DCNN [148]

Table 3.2: Global accuracy on the HMDB51 dataset [148]

Method	Accuracy
3DCNN on original dataset	0.46
H3DCNN on merged dataset	0.52
H3DCNN with binary classification	**0.632**

3.3 Evaluation Part 1

In this section, the proposed methodology is evaluated on the HMDB51 dataset. Accuracy (ACC) is used as an evaluation metric. The original 3DCNN architecture achieves accuracy of 0.46 as reported in Table 3.2. The low accuracy similar gesture classes are grouped into the pair of new classes based on the proposed rules. The total number of classes after grouping is reduced from 51 to 42, and the accuracy is increased to 0.52 globally after merging process. However, the increased classification result does not represent the performance of the original dataset. Therefore, to assess the performance orginal dataset, the binary classification is applied to the paired classes which can be extended to the classification result for all the classes. The binary classification in the hierarchical architecture further boosts the accuracy to 0.632.

The accuracy for the newly paired classes is reported in Table 3.3. The results show that the average accuracy of each pair of the proposed method is greatly improved compared to the result reported [105]. There is a s significant increase in accuracy from 0.69 to 0.82 in the classes Jump & Catch. There is also a huge improvement of 0.16 to 0.82 for the classes Throw Baseball & Swing Baseball.

The binary classification result for a new pair of classes is reported in Table 3.4. In comparison with [105], accuracy for the sit action improved from 0.39 to 0.49, and the

Table 3.3: Comparison of global accuracy on paired classes [148]

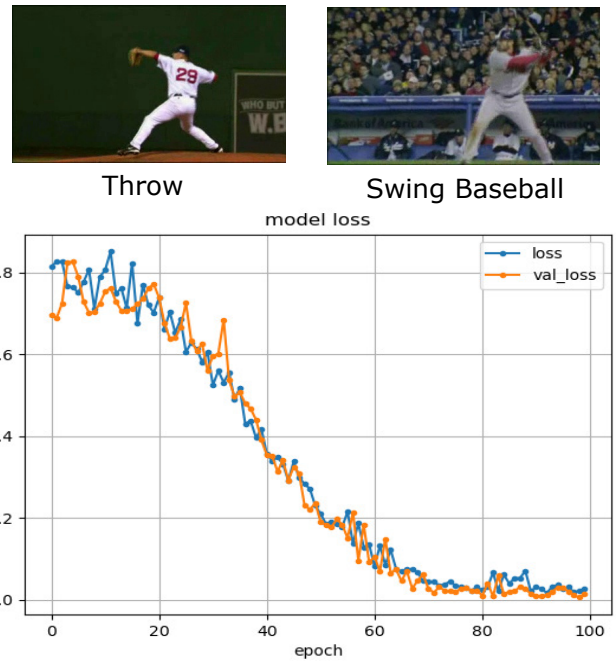
Similar Gesture Pair	Accuracy reported in[105]	**Proposed Method**
Jump & Catch	0.69	**0.82**
Kick Ball & Punch	0.41	**0.95**
Laugh & Chew	0.44	**0.86**
Pick & Golf	0.64	**0.95**
Sit & Stand	0.33	**0.76**
Throw & Swing Baseball	0.16	**0.82**
Turn & Walk	0.3	**0.72**
Wave & Shake Hands	0.48	**0.8**
Sword & Sword Exercise	0.28	**0.84**

Table 3.4: Comparison of accuracy for each class in pairs after binary classification [148]

Classes	Accuracy reported in[105]	**Proposed Method**
Jump	0.38	**0.95**
Catch	1.00	0.77
Kick Ball	0.31	**0.83**
Punch	0.51	**1.00**
Laugh	0.41	**0.93**
Chew	0.47	0.44
Pick	0.27	**0.94**
Golf	1.00	0.94
Sit	0.39	**0.49**
Stand	0.27	**0.66**
Throw	0.16	**0.7**
Swing Baseball	0.16	**0.93**
Turn	0.222	**0.57**
Walk	0.38	**0.81**
Wave	0.14	**0.55**
Shake Hands	0.82	**0.88**
Sword	0.13	**0.83**
Sword Exercise	0.42	**0.83**

pick up object action improved from 0.27 to 0.94. A similar improvement in accuracy is noted for the classes Wave, Throw, and Jump. The losses in Fig 3.5 show that for the most confusing pairs (Throw Baseball & Swing Baseball) and (Turn & Walk), the loss dramatically declines after 100 epochs. Although there is a slight decrease in accuracy for some of the actions, with our proposed hierarchical approach, global performance increases.

Table 3.5 compares the proposed method and some of the state-of-the-art methods. In



(a) Similar gesture: Throw and Swing Baseball

(b) Similar gesture: Turn and Walk

Figure 3.5: The training and validation loss of the binary classification with the similar gestures [148]

Table 3.5: Comparison of recognition accuracy on the HMDB51 dataset with state-of-the-art methods

Method	Accuracy
LSTM mode[116]	0.44
Two-stream CNN[112]	0.594
Learning to rank[34]	0.618
Coherence learning to rank with MKL [105]	0.62
Proposed Method	**0.632**

the HMDB51 dataset, the accuracy achieved 0.632. Thus, the proposed hierarchical architecture can effectively classify datasets with similar gesture actions, and it outperforms the other state-of-the-art methods [105].

3.4 Methodology Part 2

This section describes the proposed end-to-end architecture which used to perform action recognition with similar gesture classes automatically. This section details the dataset preparation method and presents the proposed hierarchical classification architecture.

3.4.1 Dataset for the proposed framework

Experiments are conducted on the KTH and UCF101 datasets, which are the most common datasets in the computer vision area to evaluate the proposed architecture as shown in Fig 3.6. KTH is an old dataset collected by KTH Royal Institute of Technology. It contains six types of actions performed by 25 actors in four different environments, namely outdoors, outdoors with scale variation, outdoors with different clothes and indoors. The dataset contains 2391 video clips. All the video clips were captured against homogeneous backgrounds using a static camera with 25 fps rate. The resolution of the KTH dataset is 160×120 pixels and each clip is around four seconds in length. The UCF101 dataset is a state-of-art dataset built by the Center for Research in Computer Vision in the University of Central Florida. This dataset contains realistic action videos collected from YouTube. It includes 13320 videos from a total of 101 action classes and it also has the largest diversity in terms of actions and with large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. The videos in 101 action classes are placed into 25 groups, each group containing four to seven videos of an action. The videos from the same group may share some common features, such as similar background and similar viewpoint,

which is why this dataset has been challenging till now. Two datasets have been used to evaluate whether the proposed architecture is capable of handling misclassified actions which have similar gestures with both low resolution videos and high-quality videos.

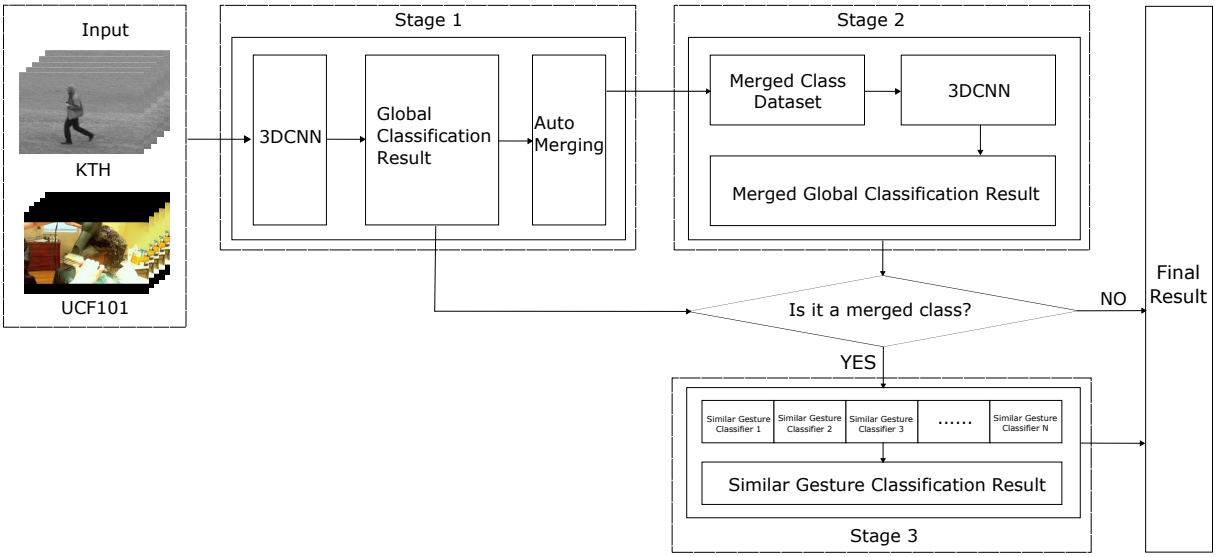


Figure 3.6: The proposed End-to-End 3DCNN architecture [147]

3.4.2 Architecture Description

Fig 3.6 presents the proposed action recognition architecture. The hierarchical structure includes three stages which is not data dependable and it can be applied to differen real-world scenarios. In this work, the KTH and UCF101 datasets were used for the experiments and validating the proposed hierarchical architecture. The input data from the KTH dataset has 6 classes and the UCF101 dataset has 101 classes initially.

Specifically, for the UCF101 dataset, the threshold follows the rules: if the accuracy of one of the classes is lower than 75% and has the highest confusion rate with another class, the system will merge all the samples from these two classes and form one new class. For the KTH dataset, because it only contain six classes, the most similar classes jogging and running will be formed automatically.

The first stage of the proposed hierarchical method focuses on classifying the generic classes (complete dataset) and identifying the high confusion classes, whereas the second stage perform the classification on the merged datasets. The last stage classifies the classes with similar/confusing gestures. The raw RGB video is taken in stage 1 to get the global classification result for whole dataset and each class. After the results have

Table 3.6: Global accuracy on the KTH dataset [147]

Method	Accuracy
3DCNN on original dataset	0.52
H3DCNN on merged dataset	0.64
H3DCNN with binary classification	**0.65**

Table 3.7: Global accuracy on the UCF101 dataset [147]

Method	Accuracy
3DCNN on original dataset	0.82
H3DCNN on merged dataset	0.83
H3DCNN with binary classification	**0.88**

been obtained, the system merges the samples from classes based on the rules and sends the samples as input to the second stage. The second stage classified the auto-processed dataset and check the result is from the merged class or not. If the result is from the merged class and the merged accuracy are higher than 75%, the sample will be passed to the third stage for further classification, otherwise, the merging process of this pair will be terminated. The third stage comprises different binary classifiers, which are used to classify the merged classes from the second stage. If the sample video is not classified as one of the merged classes in stage two, the sample video will not be passed to the third stage and the result from the second stage is obtained as the final classification result.

The final results are calculated by combining the global classification result from the second stage and the similar gesture binary classification result from the third stage. To obtain the final result for the original dataset, the results have been replaced to the merged similar gesture classes in the second stage by the results from the binary classification in the last stage.

3.5 Evaluation Part 2

In this part, the proposed methodology is evaluated on the KTH and UCF101 datasets. Accuracy (ACC) is used as an evaluation metric. The proposed 3DCNN architecture achieves global accuracy of 0.52 and 0.82 as reported for the KTH dataset and UCF101 dataset in Table 3.6 and Table 3.7 respectively. The low accuracy classes are merged into new classes based on confusion rates automatically. The total number of classes of UCF101 reduces from 101 to 79 after merging process, where the global classification accuracy increases to 0.64 for KTH and 0.83 for UCF101 respectively. However, the

Table 3.8: Accuracy of the similar classes after stage 1 (Table 1) [147]

Classes	Accuracy in stage 1	**Merged Classes**
Jogging	0.26	KTH New Class
Running	0.61	
BabyCrawling	0.76	UCF101 New Class 1
MoppingFloor	0.69	
BalanceBeam	0.6	UCF101 New Class 2
ParallelBars	0.48	
BandMarching	0.7	UCF101 New Class 3
MilitaryParade	0.73	
BlowingCandles	0.71	UCF101 New Class 4
Mixing	0.88	
CliffDiving	0.63	UCF101 New Class 5
Kayaking	0.63	
FloorGymnastics	0.67	UCF101 New Class 6
Rafting	0.58	
Haircut	0.75	UCF101 New Class 7
BlowDryHair	0.77	
Hammering	0.73	UCF101 New Class 8
BodyWeightSquats	0.94	
HandstandWalking	0.35	UCF101 New Class 9
Lunges	0.74	
HeadMassage	0.68	UCF101 New Class 10
TrampolineJumping	0.74	

reported improvements does not represent the performance of the original dataset. Therefore, to assess the performance on the original dataset, a futher binary classification is applied to the merged classes and the results are used to replace the results of the merged classes in the second stage. The inclusion of binary classifiers in the hierarchical architecture further boosts the accuracy to 0.65 for KTH and 0.88 for UCF101 datasets.

Table 3.8 and Table 3.9 provides details the accuracy of the similar gesture classes in the first stage, and also provides information about the merged classes. The first line in the pairs indicates the low performance classes and the second line in the pairs indicates the similar gesture classes. Similar gesture actions are identified by the system such as, "Jogging & Running", "Baby Crawling & Mopping Floor", "Hair Cut & Blow Dry Hair" and "Indoor Rock Climbing & Rope Climbing" etc. which are merged into one class automatically as shown in Fig 3.6. After merging process, the number of classes in the complete dataset (UCF101) is reduced from 101 classes to 79 classes and the number of classes in the KTH dataset is reduced from 6 classes to 5 classes. Moreover, the size/number of samples in the complete dataset remains the same. This process

Table 3.9: Accuracy of the similar classes after stage 1 (Table 2) [147]

Classes	Accuracy in stage 1	**Merged Classes**
HorseRiding	0.67	UCF101 New Class 12
HorseRace	0.87	
HulaHoop	0.68	UCF101 New Class 13
JumpRope	0.95	
PizzaTossing	0.66	UCF101 New Class 14
TableTennisShot	0.9	
PullUps	0.72	UCF101 New Class 15
BrushingTeeth	0.78	
RopeClimbing	0.38	UCF101 New Class 16
RockClimbingIndoor	0.83	
Rowing	0.7	UCF101 New Class 17
Skijet	0.81	
Skiing	0.72	UCF101 New Class 18
SkyDiving	0.85	
StillRings	0.65	UCF101 New Class 19
UnevenBars	0.78	
ThrowDiscus	0.61	UCF101 New Class 20
HammerThrow	0.85	
WalkingWithDog	0.56	UCF101 New Class 21
SkateBoarding	0.87	
YoYo	0.72	UCF101 New Class 22
JugglingBalls	0.91	

decreases the misclassification rate and improves the global accuracy of the dataset, as the dataset now has no similar gesture actions.

The accuracy for the newly paired classes between stage 1 and stage 2 is reported in Table 3.10. The 44 low accuracy classes in stage 1 are merged into 22 classes in stage 2. The results show that the average accuracy of each pair in stage 2 is overwhelming the result in stage 1. There is a significant increase in accuracy from 0.545 to 0.86 for the classes (HandstandWalking & Lunges) and there is also a huge improvement from 0.745 to 0.97 for the classes HighJump & JavelinThrow.

The binary classification results of the merged classes is reported in Table 3.11 and Table 3.12. In comparison with stage 1, the accuracy of the action "PizzaTossing" is improved from 0.66 to 1.00, and the accuracy of the action "HandstandWalking" is improved from 0.35 to 0.85. A similar improvement in accuracy is noted for the classes, Jogging, Rafting, and HeadMassage. Although the performance of some of the actions may have a slight decrease in accuracy. After all, the global classification performance is increased with the proposed hierarchical approach.

Table 3.10: Comparison of accuracy between stage 1 and stage 2 on paired classes [147]

Similar Gesture Pair	Accuracy in stage 1	Accuracy in stage 2
Jogging & *Running*	0.565	**0.66**
BabyCrawling & *MoppingFloor*	0.73	**0.85**
BalanceBeam & *ParallelBars*	0.54	**0.76**
BandMarching & *MilitaryParade*	0.715	**0.9**
BlowingCandles & *Mixing*	0.795	**0.92**
CliffDiving & *Kayaking*	0.63	**0.88**
FloorGymnastics & *Rafting*	0.625	**0.9**
Haircut & *BlowDryHair*	0.76	**0.91**
Hammering & *BodyWeightSquats*	0.835	**0.95**
HandstandWalking & *Lunges*	0.545	**0.86**
HeadMassage & *TrampolineJumping*	0.71	**0.95**
HighJump & *JavelinThrow*	0.745	**0.97**
HorseRiding & *HorseRace*	0.77	**0.92**
HulaHoop & *JumpRope*	0.815	**0.98**
PizzaTossing & *TableTennisShot*	0.78	**0.97**
PullUps & *BrushingTeeth*	0.75	0.74
RopeClimbing & *RockClimbingIndoor*	0.605	**1.00**
Rowing & *Skijet*	0.755	0.74
Skiing & *SkyDiving*	0.785	**0.81**
StillRings & *UnevenBars*	0.715	**0.78**
ThrowDiscus & *HammerThrow*	0.73	**0.87**
WalkingWithDog & *SkateBoarding*	0.715	**0.79**
YoYo & *JugglingBalls*	0.815	**0.93**

3.6 Discussion

The results show that the H3DCNN architecture improves the performance of the classifiers. Although some of the unconfusing classes can achieve a high classification accuracy of around 90%, on average, both globe accuracy and accuracy on similar gesture actions have been improved by combining the results from stage 2 and stage 3. This combination obtains better global results and boosts the results on similar classes compared with state-of-art works. Using 3DCNN combined with other methods such as LSTM does not achieve as the same performance as our architecture, which can be explained with the advantage of binary classification.

Dynamic analysis and evaluation are also critical. In this work, the proposed 3DCNN were used as both the global classifier and binary classifier. There could be other classifiers which can achieve a better result, which would be explored in the future work. By joining other classifiers or methods, different parameters can be tested which may

Table 3.11: Comparison of accuracy for each class in pairs after binary classification (Table 1)[147]

Classes	Accuracy in stage 1	Accuracy in stage 3
Jogging	0.26	**0.52**
Running	0.61	**0.68**
BabyCrawling	0.76	**0.89**
MoppingFloor	0.69	**0.81**
BalanceBeam	0.6	**0.85**
ParallelBars	0.48	**0.68**
BandMarching	0.7	**0.91**
MilitaryParade	0.73	**0.9**
BlowingCandles	0.71	**0.97**
Mixing	0.88	0.88
CliffDiving	0.63	**0.92**
Kayaking	0.63	**0.85**
FloorGymnastics	0.67	**0.89**
Rafting	0.58	**0.91**
Haircut	0.75	**0.95**
BlowDryHair	0.77	**0.88**
Hammering	0.73	**0.97**
BodyWeightSquats	0.94	0.92
HandstandWalking	0.35	**0.85**
Lunges	0.74	**0.87**
HeadMassage	0.68	**0.98**
TrampolineJumping	0.74	**0.92**
HighJump	0.71	**0.95**
JavelinThrow	0.78	**1.00**
HorseRiding	0.67	**1.00**
HorseRace	0.87	0.87
HulaHoop	0.68	**0.97**
JumpRope	0.95	**0.98**
PizzaTossing	0.66	**1.00**
TableTennisShot	0.9	**0.95**
PullUps	0.72	**1.00**
BrushingTeeth	0.78	**1.00**
RopeClimbing	0.38	**0.66**
RockClimbingIndoor	0.83	0.76
Rowing	0.7	0.53
Skijet	0.81	**1.00**
Skiing	0.72	**0.95**
SkyDiving	0.85	0.67
StillRings	0.65	**0.83**
UnevenBars	0.78	0.72

Table 3.12: Comparison of accuracy for each class in pairs after binary classification (Table 2)[147]

Classes	Accuracy in stage 1	Accuracy in stage 3
ThrowDiscus	0.61	**0.97**
HammerThrow	0.85	0.79
WalkingWithDog	0.56	**0.89**
SkateBoarding	0.87	0.71
YoYo	0.72	**0.92**
JugglingBalls	0.91	**0.95**

improve the results as well. Also, only three datasets were used to obtain the results, which achieves high performance. However, there are still many datasets containing actions with similar gestures. The future work will test different methods or algorithms on multiple datasets which involves considerable work to select and build a dataset with similar gestures.

3.7 Summary

In the first half of the chapter, a new approach to handle the actions with similar gestures to improve the overall accuracy of a gesture recognition system is proposed. The analysis shows that a major reason for low performance is due to the confusion arising from similar gestures. Hence, this work focuses on resolving the confusion among the classes with similar gestures in the current work. A generic hierarchical classification model is proposed in this work which can be applied to any dataset/real-world application involving gesture recognition. The first stage classifies the individual class as well as the new class formed by merging similar gestures. In the second stage, binary classification is used to resolve the confusion among the similar gesture classes. The experiment results indicate that the proposed approach outperforms not only the proposed 3DCNN but also the other state-of-the-art works. Overall, the proposed method achieves better performance on the HMDB51 dataset compared to the state-of-the-art action recognition approaches.

The second half of the chapter describes the framework being assembled into a smart end-to-end system. Using a smart processing procedure, the system can automatically merge similar gesture actions in Stage 1 and send the result from Stage 1 to Stage 3 for binary classification. The system has been evaluated on the two popular datasets, KTH and UCF101, where the results indicate that our framework improves the both global

and binary classification performance.

CHAPTER 4

DATA AUGMENTATION FOR SIMILAR GESTURE ACTION RECOGNITION

In this chapter, an end-to-end adversarial video data augmentation framework (ADAF) is proposed to enlarge the bias between the actions which contain similar gestures frames. By using the GAN generated frames, the proposed framework can boost the classification performance on similar gesture action pairs as well as the complete datasets. Experiments were conducted on three typical human action recognition datasets: KTH, UCF101 and HMDB51 show that the data augmentation can boost the classification performance on either similar gesture pairs or complete datasets.

Some of parts of this chapter have been accepted in the paper titled "Adversarial Action Data Augmentation for Similar Gesture Action Recognition", Wu et al. [144], and the major parts of this chapter is under review by journal of Expert Systems with Application as an extended version of the conference version.

4.1 Introduction

Video-based human action recognition is more challenging than image-based human action recognition, because a sequence of video frames includes both spatial and temporal information. Unlike action recognition in still images, temporal information plays an important role in recognizing human actions in continuous frames which include additional features such as time series information and gesture sequence information.

Video-based human action recognition therefore seeks to analyze the spatial features using the additional information in each frame. Human action recognition methods have achieved a huge success by using deep neural networks, however, actions with similar gestures could reduce the performance of the classification.

Human actions are ubiquitous in our lives and can consist of simple gestures or combinations of multiple simple gestures according to the action. For example, a person swinging their arm might be simply recognized as the simple action "waving", or action "walking" can be recognized from the raising and lowering of the feet, but the same action conducted at speed would be "running". Some actions are combinations of different gestures; for instance, the action "pick up" might include the simple gestures of bending at the waist and grabbing, but this could be easily be confused with the action "bend over". Distinguishing between similar gesture actions is one of the problems affecting classification results. To address this problem, many human action recognition approaches have proposed algorithms such as recurrent neural networks (RNN) to process the time series information, in an effort to improve the performance on similar gestures. However, many similar gestures also include similar time series information, such as "running" and "jogging". The similar gesture problem thus persists.

Human actions in video streams are considered to be sequential gestures, so a video-based human action recognition method learns the features on each video frame and the time series information between frames. Most early traditional machine learning (ML) works [67] [28] [70] are problem-dependent and apply texture descriptors on the extracted hand-crafted motion features.

Over the past 10 years, deep learning (DL) methods have achieved tremendous success in the computer vision and human action recognition research area. Convolutional neural networks (CNN) have been widely used to process images and recognize human actions by extracting and learning the features in the video frames automatically. A two-step framework that uses CNN and RNN to learn the features from video frames was proposed by [6] however; the proposed method ignores the feature correlation between frames when the sequential information in the frame has been learnt by the RNN. Ji et al. [53] introduced 3DCNN to address this problem; the method preserves the features of the same pixel spots between adjacent frames, by convolving the features on sequential frames. However, 3D convolution can only partially preserve the temporal information by convolving the changing features of the same pixels.

Recurrent neural network (RNN) and long short term memory (LSTM) models have achieved good performance in many time series methods such as natural language

processing (NLP). These time series models perform the same task on all sequential elements. LRCNs [29] and HM-AN [152] are recent RNN works which process temporal features. In these methods, the new temporal features extracted from the frames are used to update the hidden states and the previous hidden states are forgotten. However, the temporal features are hand-crafted features or extracted from the CNN, hence the similarity of the extracted features will reduce the performance of RNN.

To recognize human actions accurately, many combined methods have been proposed. Simonyan et al. [112] proposed a two-stream convolutional network for video-based human action recognition which blends the results from the spatial and temporal channels in the last layer of the network. The two channels simultaneously process the information, and many other works have followed this idea by modifying the networks [74] or choosing the different features [33]. Nonetheless, two-stream methods still cannot solve the similar feature problem because they are feature dependent methods.

Deep neural networks have demonstrated promising performance in the action recognition area. Hara et al. [44] and He et al. [46] achieved high performance when deep ResNets were conducted on Kinetics400 and Kinetics600 datasets with 34 layers and 101 layers, respectively. Deep layer methods require huge datasets to supply more features and the cost is extremely high. However, deep neural networks will not able to achieve the same performance on small datasets such as UCF101 and HMDB51.

All the human action recognition approaches discussed above that rely on ML and DL methods are feature-dependent. They overlook the fact that, unlike still image action recognition, video-based human actions consist of sequential gestures, and most valid frames may have similar gestures. Figure. 1.2 shows some of the similar gesture pairs from UCF101, KTH and HMDB51 datasets. Figure. 1.2(d) illustrates the action pair "running" and "jogging" which share similar gestures. Even a human cannot recognize the differences between these two actions in a still image. In addition, the time series information of "running" and "jogging" are similar, which will confuse the classifiers. The same thing happens in Figure. 1.2(a), Figure. 1.2(b) and Figure. 1.2(c) which show similar gesture pairs such as "Rope Climbing and Rock Climbing Indoor", "Blow Dry Hair and Hair Cut" and "Mopping Floor and Baby Crawling". The problem is obvious: feature-based methods lack the ability to process the extracted similar features, and the deep learning methods will also obtain similar features from similar gestures.

One solution to address the problems of similar features and data shortage is video augmentation. Traditional video augmentation methods modify the forms of frames based on the original video frames such as "Super pixel", "Gaussian Blur", "Invert Color",

"Random Rotate" and "Vertical Flip" etc. These augmentation methods can only change some of the value on certain pixels, and the duplicate features will not make significant changes because the augmented frames are almost as same as the original frames with added noise. Generative adversarial nets (GAN) [40] have achieved high performance in generating still images. Bowles et al. [11] introduced GAN to generate still Fluid-Attenuated Inversion Recovery (FLAIR) and Magnetic Resonance (MR) images for the segmentation task and compared the performance with the other augmentation methods.

Inspired by GAN, an end-to-end adversarial video data augmentation framework (ADAF) is proposed to tackle the similar gesture recognition problem. The experiments of the proposed framework have been conducted on three typical human action recognition datasets, KTH [107], UCF101 [115] and HMDB51 [61], which were proposed by the KTH Royal Institute of Technology, the Center for Research in Computer Vision at the University of Central Florida, and the Serre Lab at the Brown University, respectively. For data augmentation purposes, GAN is used to generate frames which can create more features and increase the bias of the original data features. The experiments show that the proposed ADAF boosts the overall performance of the dataset as well as improving the accuracy of similar pairs recognition compared to the baseline CNN model.The framework could also be adapted for use in other CNN-based methods. The major contributions in this chapter are as follows:

- The GAN-based data augmentation method is proposed to address the problem of similar gesture recognition.

- The proposed end-to-end action data augmentation framework (ADAF) not only improves classification performance on similar gesture action pairs but also the performance on the complete dataset.

- To combine similar gestures, the framework automatically identifies similar gestures based on the highest confusion rate from the baseline CNN classifier.

- The proposed ADAF has been evaluated on the KTH, UCF101 and HMDB51 datasets and compare the results with those of the baseline CNN model. Experimental results show that the proposed framework outperforms the baseline methods in both global accuracy and the identification of similar gesture pairs.

4.2 Problem definition

For a certain CNN input frame **f** from a video, the output of the CNN layer $h(i,j)$ is shown in Eq. 4.1:

$$h(i,j) = \sum_{k,l} f(i,j) * g(i-k, j-l) \tag{4.1}$$

where f denotes the frame, and the pixel position is represented as i,j; thus, $f(i,j)$ is the feature value of the pixel position i,j of the frame f. In addition, g indicates the kernel and k represents the row and l represents the column.

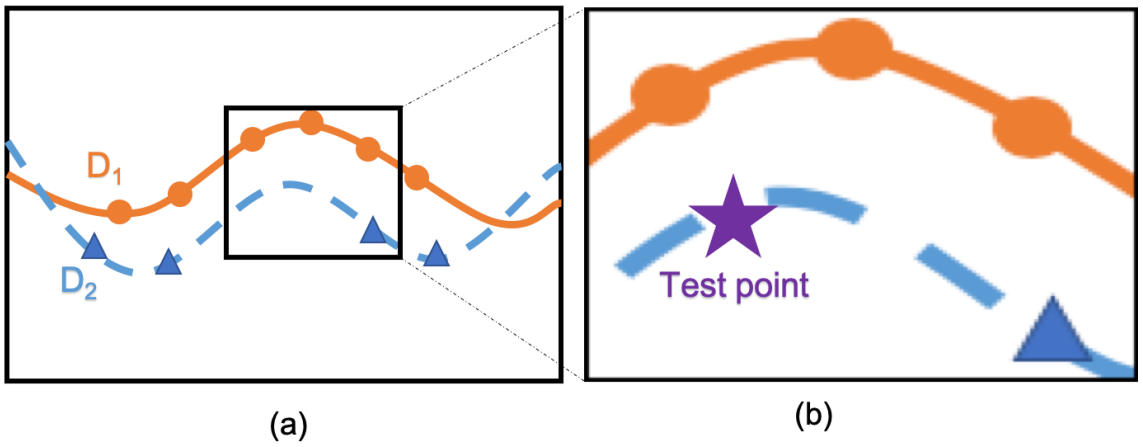


Figure 4.1: Learning distribution with original data [144]

Because CNNs are pixel-based approaches, frames that include similar actions gestures with similar backgrounds will generate a similar data distribution D. The circles(D_1) and triangles(D_2) represent the learnt distribution of two similar gestures from the video stream in Figure. 4.1(a). Figure. 4.1(b) shows that the test point (star) belonging to class D_1 is misclassified into D_2 due to the similarity of distribution.

4.3 Methodology

In this section, the problems encountered in solving the task of similar gesture recognition will be discussed and the structure of ADAF will be described.

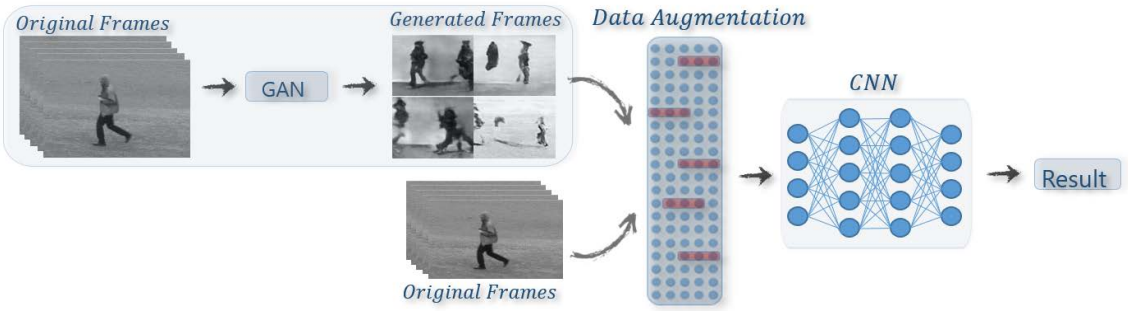


Figure 4.2: The framework of the proposed ADAF [144]

4.3.1 Datasets for evaluation

The experiments were conducted on three popular human action recognition datasets, KTH [107], UCF101 [115] and HMDB51 [61]. KTH is an old dataset but is still very challenging. It consists of six actions. Each action is performed by 25 different actors with four different backgrounds. The whole dataset has 2391 video streams in total, and each video was captured by a static camera at a rate of 25 fps.

UCF101 contains 101 different actions collected from YouTube which has 13320 realistic action videos. The 101 action classes are categorized into 25 groups, and each group has four to seven videos for a single action. Videos from the same group share common features, for example, similar viewpoint or similar background, thus classifying videos containing similar gestures could present a challenge.

HMDB51 comprises a total of 7000 video clips collected from commercial movies and YouTube. The video clips contain 51 human actions such as facial gestures, body movements and interactions between bodies and objects, and each class contains approximately 100 videos. It is a very challenging dataset because the multiple actions consist of similar gestures performed by different persons. Each action has been captured from several viewpoints and recorded in four to six video clips.

All datasets present the problem of gesture recognition because they contain multiple actions made up of similar gestures. In addition, the volume of each dataset is smaller than that of the new proposed dataset. This will result in low performance on the new proposed methods, because these datasets cannot supply enough training data for deep neural networks.

4.3.2 Framework

In this subsection, an end-to-end framework that uses GAN-based data augmentation for similar gesture recognition is proposed. The objective of the proposed framework is to train a robust system which can accurately classify similar gestures. There are three stages in the proposed framework: data preparation, data augmentation and classification. First, the original videos are set as the input of the GAN, then new frames for each action video were generated. Based on the data augmentation progress, the original frames and the generated frames will be combined. In the final stage, the augmented data will be used as the CNN input and the final result will be obtained.

4.3.2.1 Data preparation

The original videos are fed into the CNN to obtain the confusion matrix. The system automatically pairs the classes based on two rules:

- Rule 1: List all the classes with accuracy lower than 0.75,
- Rule 2: Pair the classes which have the highest confusion matrix and train the pairs as the binary classification.

The framework then combines the original video frames with the GAN-generated frames for the purpose of training on the same CNN network. Lastly, the trained CNN classifies the original frames. The classification results for the original images following the application of Rule 1 and Rule 2 for KTH, UCF101 and HMDB51 are shown in Table 4.1, Table 4.2 and Table 4.3, respectively.

Table 4.1: Similar gesture action recognition result on original KTH dataset [144]

Classes	Rule 1 Result	Rule 2 Result
Jogging	0.24	0.55
Running	0.63	

The results from the data preparation stage show that most of the similar gestures demonstrate low performance on both global classification and binary classification tasks such as “jogging” and “running” from the KTH dataset, “baby crawling” and “mopping floor” from the UCF101 dataset and “sit” and “stand” from the HMBD51 dataset. Some classes in other pairs achieve high classification performance, such as “jump rope” in “hula hoop” and “jump rope” from UCF101, which achieves 0.95 accuracy, “golf” in “pick” and “golf” from HMBD51, which has 1.00 accuracy. However, the class “hula rope”

Table 4.2: Similar gesture action recognition result on original UCF101 dataset [144]

Classes	Rule 1 Result	Rule 2 Result
BabyCrawling	0.52	0.48
MoppingFloor	0.40	
BalanceBeam	0.62	0.55
ParallelBars	0.46	
BlowingCandles	0.43	0.42
Mixing	0.35	
CliffDiving	0.63	0.71
Kayaking	0.58	
Haircut	0.52	0.57
BlowDryHair	0.61	
Hammering	0.64	0.55
BodyWeightSquats	0.58	
HeadMassage	0.42	0.47
TrampolineJumping	0.53	
HighJump	0.52	0.54
JavelinThrow	0.55	
HorseRiding	0.22	0.49
HorseRace	0.87	
HulaHoop	0.68	0.82
JumpRope	0.95	
PizzaTossing	0.56	0.61
TableTennisShot	0.71	
PullUps	0.72	0.71
BrushingTeeth	0.78	
RopeClimbing	0.38	0.73
RockClimbingIndoor	0.83	
Rowing	0.61	0.66
Skijet	0.74	
Skiing	0.72	0.59
SkyDiving	0.52	
WalkingWithDog	0.56	0.63
SkateBoarding	0.87	
YoYo	0.42	0.43
JugglingBalls	0.51	

Table 4.3: Similar gesture action recognition result on original HMDB51 dataset [144]

Classes	Rule 1 Result	Rule 2 Result
$Jump$	0.38	0.65
$Catch$	1.00	
$KickBall$	0.31	0.44
$Punch$	0.51	
$Laugh$	0.41	0.48
$Chew$	0.47	
$Pick$	0.27	0.52
$Golf$	1.00	
Sit	0.39	0.49
$Stand$	0.27	
$Throw$	0.16	0.41
$SwingBaseball$	0.16	
$Turn$	0.22	0.75
$Walk$	0.38	
$Wave$	0.14	0.59
$ShakeHands$	0.82	
$Sword$	0.13	0.55
$SwordExercise$	0.42	

and “pick” demonstrates extremely low performance with 0.68 and 0.27 respectively, as well as a high confusion rate with the high performance classes, thus the pair will be established based on Rule 2. The binary classification result also proves that even when the performance of the pairs is imbalanced, a high confusion rate for one of the classes will reduce the performance of the binary classification. The precision, recall and F1-score in Table 4.4 and Table 4.5 show how the similar gestures affect the results. According to the result, the classifier classify all the frames into one class.

These results demonstrate that it is crucial to find a way to reduce the high confusion rate in order to improve classification performance. Data augmentation is one possible solution.

4.3.2.2 Data augmentation

Many data augmentation methods have been proposed, such as Rotation, Super Pixel, and Gaussian Blur. These methods only change the features on certain pixels following certain rules and do not enlarge the differences between classes. We randomly pick 2000 frames from original videos and combine them with 2000 rotated frames to test whether traditional data augmentation works on similar gestures. The reason for choosing rota-

Table 4.4: Precision, recall and F1-score on the original UCF101 pairs [144]

Classes	Precision	Recall	F1-score
BabyCrawling *MoppingFloor*	0.5	0.24	0.32
BalanceBeam *ParallelBars*	0.5	0.22	0.31
BlowingCandles *Mixing*	0.5	0.21	0.29
CliffDiving *Kayaking*	0.5	0.34	0.41
Haircut *BlowDryHair*	0.5	0.28	0.37
HeadMassage *TrampolineJumping*	0.5	0.24	0.32
Hammering *BodyWeightSquats*	0.5	0.28	0.36
HighJump *JavelinThrow*	0.5	0.27	0.35
HorseRiding *HorseRace*	0.5	0.25	0.33
HulaHoop *JumpRope*	0.5	0.41	0.45
PizzaTossing *TableTennisShot*	0.5	0.32	0.39
PullUps *BrushingTeeth*	0.5	0.35	0.41
RopeClimbing *RockClimbingIndoor*	0.5	0.37	0.43
Rowing *Skijet*	0.5	0.33	0.39
Skiing *SkyDiving*	0.5	0.21	0.29
WalkingWithDog *SkateBoarding*	0.5	0.31	0.38
YoYo *JugglingBalls*	0.5	0.21	0.31

Table 4.5: Precision, recall and F1-score on the original HMDB51 pairs [144]

Classes	Precision	Recall	F1-score
Jump *Catch*	0.5	0.32	0.39
KickBall *Punch*	0.5	0.22	0.31
Pick *Golf*	0.5	0.24	0.32
Sit *Stand*	0.5	0.24	0.33
Throw *SwingBaseball*	0.5	0.79	0.37
Turn *Walk*	0.5	0.38	0.43
Wave *ShakeHands*	0.5	0.29	0.37
Sword *SwordExercise*	0.5	0.28	0.36

Table 4.6: Accuracy, precision, recall and F1-score after rotation on typical similar gesture actions [144]

Classes	Accuracy	Precision	Recall	F1-score
BabyCrawling *MoppingFloor*	0.51	0.5	0.25	0.34
BalanceBeam *ParallelBars*	0.49	0.5	0.25	0.33
Turn *Walk*	0.49	0.5	0.24	0.33
Wave *ShakeHands*	0.49	0.5	0.25	0.33

tions is that other pixel-based augmentation methods only simultaneously change the value on certain pixels, whichdoes not have sufficiently high impact on the classifier, and the rotation will change the location of the original pixels, thus the augmented frames can be treated as different frames. Table 4.6 shows the results after applying one traditional data augmentation method, namely rotation, on typical similar gesture pairs.

The results in Table 4.6 show that traditional augmentation methods cannot tackle the similar gesture problem because the correlation between pixels remains the same after the change. Thus, to enlarge the feature differences between classes, a method is needed which can generate the frames to represent different features from the original

frames.

Generative adversarial networks, Ian Goodfellow proposed generative adversarial networks (GAN) [40] which created a new method of data augmentation in 2014. The GAN consists of two parts, a generator $\mathscr{G}$ and a discriminator $\mathscr{D}$; $\mathscr{G}$ generates image and $\mathscr{D}$ discriminates whether the generated image is from original dataset or from $\mathscr{G}$. By playing this adversarial game, the generator can be trained to generate high quality images. Considering the video frame sequences $\mathscr{F} = \{f_1, f_2, ..., f_n\}$, deep convolutional GAN (DCGAN) was used [5] to generate the fake frame $\hat{f}$ for data augmentation, which is shown in Eq. 4.2,

$$\min_{\mathscr{G}} \max_{\mathscr{D}} \underset{f \sim p_{(f)}}{E} [\log \mathscr{D}(f)] + \underset{\hat{f} \sim p(\hat{f})}{E} [\log(1 - \mathscr{D}(\mathscr{G}(\hat{f}))] \tag{4.2}$$

where, p_f represents the original frame distribution and $p_{\hat{f}}$ is the generated image distribution. Both $\mathscr{D}$ and $\mathscr{G}$ contain convolutional layers; $\mathscr{G}$ does not have any pooling or fully connected layers, whereas $\mathscr{D}$ uses a single dimension sigmoid function as the output layer. $\mathscr{D}$ determines whether the sample has been generated by the generator $\mathscr{G}(\hat{f})$ (fake) or whether it is from the original video frames p_f (real). The generator will generate the improved frame $\hat{\mathscr{F}}$ by minimizing the cost for both $\mathscr{D}$ and $\mathscr{G}$ during the training stage.

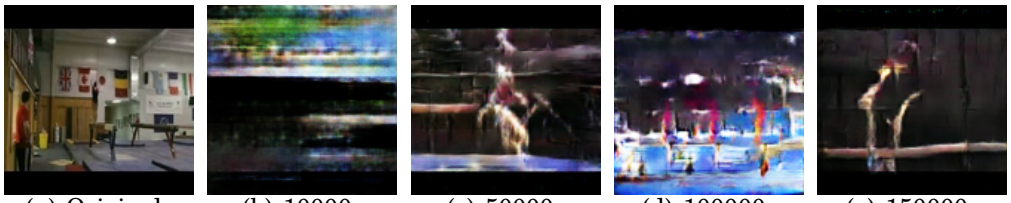
Unlike generating still images, human action in videos can be represented as sequences of gestures. The gestures on the adjacent frames f_{n-1}, f_n and f_{n+1} are different. However, $\mathscr{D}$ is only a binary classifier capable of discriminating whether an image is real or fake. It learns the features from the sequence frames, and the generated frame $\hat{f}$ from $\mathscr{G}$ will be updated by the significant features learnt by $\mathscr{D}$. Figure. 4.3 to 4.6 inclusive show the generated frames of some of the paired classes from 10000 iterations to 150000 iterations. For instance, the generated frames for “Baby crawling” and “Mopping floor” are very similar after training 10000 iterations, but after 50000 iterations, the actions become clear. The frames show multiple afterimages which follow the the original video gesture sequences. This is because the performance of discriminator $\mathscr{D}$ is not high enough in the middle of the training stage, so the generator $\mathscr{G}$ tries to generate a single image which can represent as many as features as possible to represent the sequences of the action to fool the discriminator $\mathscr{D}$. After 150000 iterations,the actions can be seen clearly, which means that both $\mathscr{G}$ and $\mathscr{D}$ achieved high performance. However, some classes such as “Turn” and “Walk” can only be determined by the frame sequences, as there is no difference between turning and walking on a single frame. The afterimages on the generated frames will therefore form the bias between these two classes, such as the

"Turn" and "Walk" in Figure. 4.6(c) and Figure. 4.6(g) generated after 100000 iterations. The afterimages in Figure. 4.6(g) for the action "Walk" are greater than for the action "Turn" in Figure. 4.6(c).

(a) Original (b) 10000 (c) 50000 (d) 100000 (e) 150000

(f) Original (g) 10000 (h) 50000 (i) 100000 (j) 150000

Figure 4.3: Generated frames for actions of Baby crawling (line 1) and Mopping floor (line 2) from 10000 iterations to 150000 iterations [144]



(a) Original (b) 10000 (c) 50000 (d) 100000 (e) 150000

(f) Original (g) 10000 (h) 50000 (i) 100000 (j) 150000

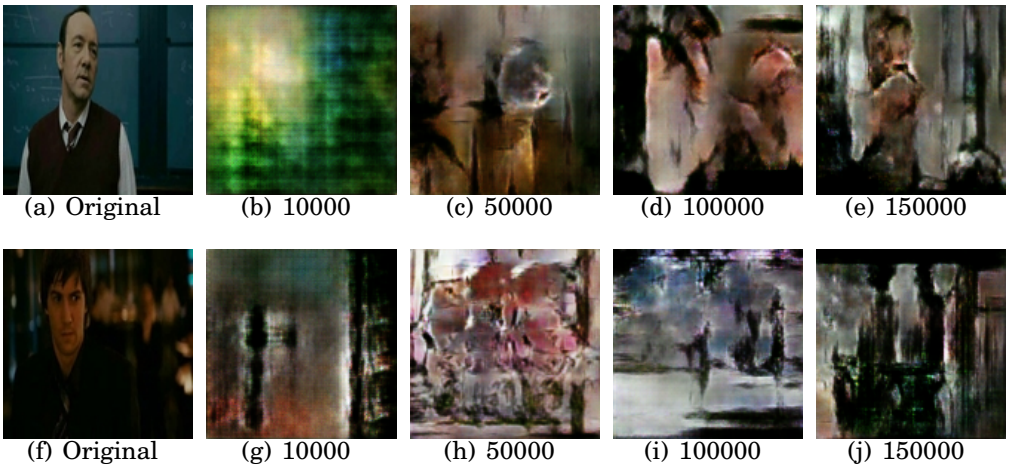
Figure 4.4: Generated frames for actions of Balance beam (line 1) and Parallel bars (line 2) from 10000 iterations to 150000 iterations [144]

Data augmentation, given the video frame set $\hat{\mathscr{F}}$ generated by the generator $\mathscr{G}$ based on the original video dataset $\mathscr{F}$, the augmented dataset S will be the combination of the original dataset $\mathscr{F}$ and the generated dataset $\hat{\mathscr{F}}$, where $S = \mathscr{F} + \hat{\mathscr{F}}$. According to [45] and [4], the augmented dataset will generate more samples in which the original data distribution will be changed, and the temporal information on the generated frames,

(a) Original (b) 10000 (c) 50000 (d) 100000 (e) 150000

(f) Original (g) 10000 (h) 50000 (i) 100000 (j) 150000

Figure 4.5: Generated frames for actions of Wave (line 1) and Shake hands (line 2) from 10000 iterations to 150000 iterations [144]



(a) Original (b) 10000 (c) 50000 (d) 100000 (e) 150000

(f) Original (g) 10000 (h) 50000 (i) 100000 (j) 150000

Figure 4.6: Generated frames for actions of Turn (line 1) and Walk (line 2) from 10000 iterations to 150000 iterations [144]

such as afterimages, will increase the bias between classes, as shown in Figure. 4.7. The additional samples move the decision boundary, where the solid line D_1 has been shifted to the dot line, thus the test point (star) will be located on the new decision boundary, and the D_2 moves down, because the added samples enlarge the data bias.

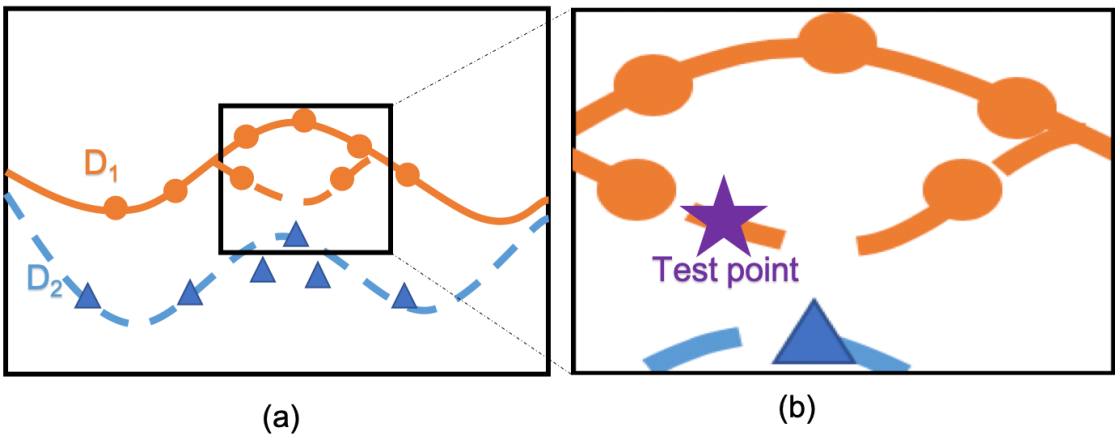


Figure 4.7: Learning distribution after data augmentation [144]

Convolutional neural networks, to evaluate the performance of the proposed ADAF, the spatial stream convolutional network reported in [112] is used as the CNN baseline. This convolutional network contains four convolutional layers and one fully connected layer with a softmax function as the output. The frames will be reshaped to 32×32 pixels and the kernel size of each 2DCNN layer is 3×3 with 2×2 pooling size after each convolutional layer. The structure of the convolutional network is shown in Figure. 4.8.

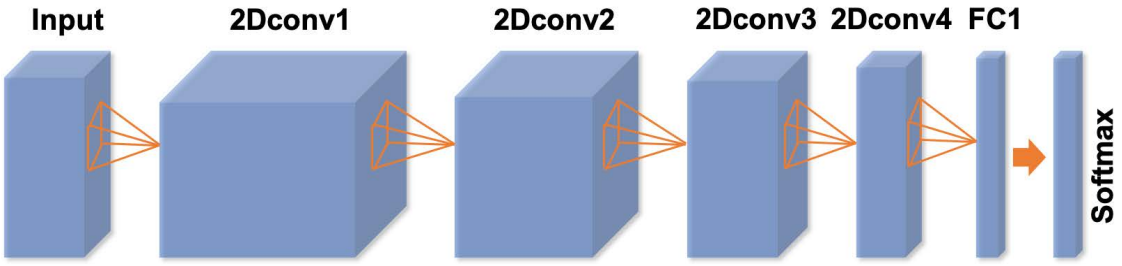


Figure 4.8: Structure of the convolutional network [144]

4.3.3 Algorithm

The algorithm of the proposed ADAF are shown in ALG. 1.

Algorithm 1 ADAF Algorithm

Require:

$\mathscr{F} = \{f^1, f^2, ..., f^n\}$: the original dat;
T: the number of iterations;
m: the batch size;
η: learning rate of the generator and discriminator;
θ_g: the parameters of generator $\mathscr{G}$;
θ_d: the parameters of discriminator $\mathscr{D}$;
z: noise samples $z^1, z^2, ..., z^i$ from the prior $P_{prior}(z)$; ACC: the accuracy from the convolutional network classifier;

1: initialize the generator $\mathscr{G}$ and discriminator $\mathscr{D}$;
2: **for** iterator = 1,2,3, $\cdots\cdots$, T **do**
3: **for** G-steps do **do**
4: $\mathscr{G}$ generates the augmentation frames based on each class;
5: Update the parameters of $\mathscr{G}$;

$$\overline{V} = -\frac{1}{m}\sum_{i=1}^{m}\mathscr{D}(\mathscr{G}(z^i))$$

$$\theta_g = \theta_g - \eta\sum\overline{V}(\theta_g)$$

6: **end for**
7: **for** D-steps do **do**
8: $\mathscr{D}$ classifies the original frames f and generated frames $\hat{f}$;
9: Update the parameters of $\mathscr{D}$;

$$\overline{V} = \frac{1}{m}\sum_{i=1}^{m}D(f^i) - \frac{1}{m}\mathscr{D}(\mathscr{G}(z^i))$$

$$\theta_d \leftarrow \theta_d - \eta\sum\overline{V}(\theta_d)$$

10: **end for**
11: **end for**
12: train the convolutional network with the original frames and augmented frames;
13: classifies the original frames with the convolutional network;
14: **return** ACC; =0

4.4 Experimental results

In this section, the results of the experiments on the KTH, UCF101 and HMDB51 datasets are reported. The evaluation metrics of the experiment will be introduced first, followed by the experiment settings, and then report the results of the classification

experiments conducted on the KTH, UCF101 and HMDB51 datasets with the baseline convolutional network. Lastly, the parameter evaluation is discussed.

4.4.1 Evaluation metrics

The metrics used in this chapter are accuracy (ACC), Precision, Recall and F1-score, which are common metrics in video-based human action recognition.

Positives and Negatives. Given a frame f belonging to the action class C and $\hat{f}$ belonging to another class, the output of the classifier is to determine whether f belongs to C class, the **True Positives**, **False Positives**, **True Negatives**, and **False Negatives** are common metrics for measuring the performance of the classifier and can be defined as follows:

- **True Positives (TP)**: f classified as belonging to C.
- **False Positives (FP)**: $\hat{f}$ classified as belonging to C.
- **True Negatives (TN)**: $\hat{f}$ classified as not belonging to C.
- **False Negatives (FN)**: f classified as not belonging to C.

according to the metrics, accuracy (ACC), precsion, recall and F1-score can be represented as Eq. 4.3, Eq. 4.4, Eq. 4.5 and Eq. 4.6, respectively.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.5}$$

$$F1 - score = \frac{2 * (Recall * Precision)}{Recall + Precision} \tag{4.6}$$

4.4.2 Experiment setup

Tensorflow and Keras were used to construct and train the convolutional networks. Tensorflow and Keras have complete libraries to support our experiment. Two NVIDIA P6000 GPUs with 24G graphics memory each, and CUDA 9 were used to conduct the experiments. The RGB frames extracted from the videos were set as the input of the DCGAN, and the GAN-generated frames fused with the original videos as the augmented dataset. Furthermore, the augmented dataset was used as the training set for the convolutional networks, and the original videos were used for testing. The purpose was to test whether our proposed ADAF was capable of boosting performance with both baseline models. Therefore, the original video was set as the input of the convolutional networks to obtain the results as the benchmark. For training purposes, we randomly spilled 80% videos from the dataset as the training set, and remaining videos was split into 15% as the testing set and 5% as the validation set. Only original videos were used for testing. For KTH and UCF101, 4000 frames generated from 80000 iterations combined with 4000 frames from original video of each class were used to train the convolutional neural networks. For the HMDB51 dataset, 2000 frames generated from 150000 iterations combined with 2000 frames from original video of each class were used to train the convolutional neural networks.

4.4.3 Results

In this section, the performance of the proposed ADAF on global classification and binary classification will be discussed. Both the global and binary classification results show that the proposed data augmentation framework boosts performance on the baseline CNN. Table. 4.7 compares the results of the original dataset and the augmented dataset for global classification on the three datasets. Before augmentation, the result on original KTH, UCF101 and HMDB51 was 0.52, 0.61 and 0.54, respectively. After data augmentation, the performance boost on KTH, UCF101 and HMDB51 was 67%, 52% and 68% respectively for the global classification.

The comparison of the binary classification results on the KTH dataset is shown in Table 4.8. The accuracy on the similar gesture pair “Jogging” and “Running” has been increased by about 13%.

After data augmentation, all the binary classification results in the UCF101 dataset were increased, as listed in Table 4.9. “Blowing Candles” and “Mixing” shows the highest improvement at about 129%. Others like “High Jump” and “Javelin Throw” only achieve

Table 4.7: Comparison global classification between original data and augmented data [144]

Dataset	Global classification result
Original KTH	0.52
Augmented KTH	**0.87**
Original UCF101	0.61
Augmented UCF101	**0.93**
Original HMDB51	0.54
Augmented HMDB51	**0.91**

Table 4.8: Comparison of binary classification accuracy on KTH dataset between original data and augmented data [144]

Class pairs	Original	Augmented
Jogging & Running	0.68	**0.77**

a slight improvement of about 2%. The reason for this is that some of the classes may be confused with multiple other classes, and we picked classes with a high misclassification rate for pairing, thus this kind of augmented pair can only improve the performance slightly. However, the performance of some pairs such as “Horse Riding” and “Horse Race"” is not improved. Even the human eye cannot distinguish the differences between these two classes, therefore, the generated images are not helpful.

Table 4.10 shows the accuracy, precision, recall and F1-score after augmentation on the UCF101 dataset. Compare to Table 4.4, most precision, recall and F1-scores have increased. In Table 4.4, we find that the precision of each pair is 0.5 but the accuracy is higher or lower than 0.5. This is because the original data size for different classes is imbalanced, which means that one class may have more frames than the other once all the frames belonging to one class have been classified. In Table 4.10, we apply the balanced training data of 8000 frames for each class, thus when misclassification occurs, the accuracy will be 0.5.

Similar results are shown in Table 4.11, in which the performance of all misclassified pairs has been improved. The highest improvement is 118% for “Kick ball” and “Punc”. The lowest improvement is about 30% for “Turn” and “Walk”.

Comparing precision, recall and F1-scores between Table 4.12 and Table 4.5, all the precision, recall and F1-scores have also increased after augmentation. The reason for the difference in accuracy from original dataset is because of the imbalanced data.

The experiment results show that the proposed ADAF boosts both global and binary classification performance on the baseline convolutional network. The results also

Table 4.9: Comparison of binary classification accuracy on UCF101 dataset between original data and augmented data [144]

Class pairs	Original	Augmented
BabyCrawling & *MoppingFloor*	0.48	**0.96**
BalanceBeam & *ParallelBars*	0.55	**0.95**
BlowingCandles & *Mixing*	0.42	**0.96**
CliffDiving & *Kayaking*	0.49	**0.74**
Haircut & *BlowDryHair*	0.57	**0.96**
Hammering & *BodyWeightSquats*	0.55	**0.95**
HeadMassage & *TrampolineJumping*	0.47	**0.94**
HighJump & *JavelinThrow*	0.54	0.51
HorseRiding & *HorseRace*	0.49	0.51
HulaHoop & *JumpRope*	0.82	**0.93**
PizzaTossing & *TableTennisShot*	0.61	**0.97**
PullUps & *BrushingTeeth*	0.70	**0.95**
RopeClimbing & *RockClimbingIndoor*	0.73	0.51
Rowing & *Skijet*	0.66	**0.92**
Skiing & *SkyDiving*	0.59	**0.95**
WalkingWithDog & *SkateBoarding*	0.63	**0.95**
YoYo & *JugglingBalls*	0.43	**0.94**

show that ADAF achieves a significant improvement by frame generation and data augmentation, which can improve the classifier's ability to learn similar gestures.

4.4.4 Parameter evaluation

To evaluate the parameters, the training loss of the CNN has been evaluated by the changes in epoch. The performance of the classifier has been evaluated by changing GAN epochs and different data fusion rates.

Training loss of the CNN, during the training stage, the changes in the loss from 0 to 200 epochs has been evaluated. Figure. 4.9 shows that the augmented data start with a lower loss rate, which is because the fused data increase the data bias. In addition, the loss of the augmented data to decrease more sharpely than the original data. The original loss represented by the dotted line remains stable and is much higher than the augmented data in relation to the low performance of the classifier. However, the augmented data loss line is rather high, as seen in Figure. 4.9(b) and Figure. 4.9(h). After 10 epochs, the lines show a sharp decrease before stabilizing and achieving high classification performance.

Performance on different GAN epochs, by increasing the number of GAN itera-

Table 4.10: Precision, recall and F1-score after augmentation on UCF101 dataset [144]

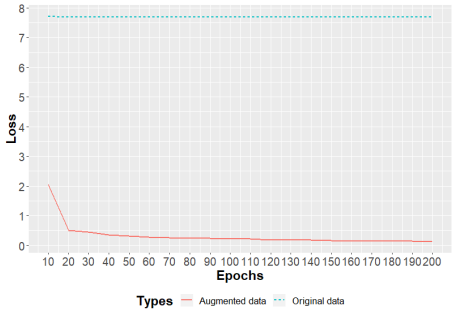
Classes	Precision	Recall	F1-score
BabyCrawling *MoppingFloor*	0.96	0.96	0.96
BalanceBeam *ParallelBars*	0.95	0.95	0.95
BlowingCandles *Mixing*	0.96	0.96	0.96
CliffDiving *Kayaking*	0.74	0.74	0.73
Haircut *BlowDryHair*	0.96	0.96	0.96
HeadMassage *TrampolineJumping*	0.94	0.94	0.94
Hammering *BodyWeightSquats*	0.95	0.95	0.95
HighJump *JavelinThrow*	0.5	0.27	0.35
HorseRiding *HorseRace*	0.5	0.25	0.33
HulaHoop *JumpRope*	0.93	0.93	0.93
PizzaTossing *TableTennisShot*	0.97	0.97	0.97
PullUps *BrushingTeeth*	0.95	0.95	0.95
RopeClimbing *RockClimbingIndoor*	0.5	0.25	0.33
Rowing *Skijet*	0.92	0.92	0.92
Skiing *SkyDiving*	0.95	0.95	0.95
WalkingWithDog *SkateBoarding*	0.95	0.95	0.95
YoYo *JugglingBalls*	0.94	0.94	0.94

Table 4.11: Comparison of binary classification accuracy on HMDB51 dataset between original data and augmented data [144]

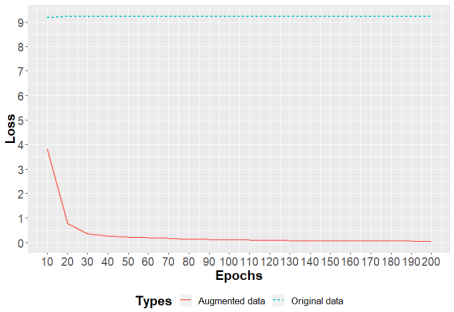
Class pairs	Original	Augmented
Jump & Catch	0.64	**0.95**
KickBall & Punch	0.44	**0.96**
Pick & Golf	0.47	**0.98**
Sit & Stand	0.48	**0.95**
Throw & SwingBaseball	0.59	**0.95**
Turn & Walk	0.75	**0.98**
Wave & ShakeHands	0.59	**0.97**
Sword & SwordExercise	0.55	**0.95**

Table 4.12: Precision, recall and F1-score after augmentation on HMDB51 dataset [144]

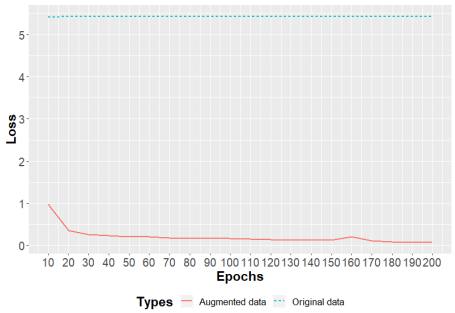
Classes	Precision	Recall	F1-score
Jump *Catch*	0.95	0.95	0.95
KickBall *Punch*	0.96	0.96	0.96
Pick *Golf*	0.98	0.98	0.98
Sit *Stand*	0.95	0.95	0.95
Throw *SwingBaseball*	0.95	0.95	0.95
Turn *Walk*	0.98	0.98	0.98
Wave *ShakeHands*	0.97	0.97	0.97
Sword *SwordExercise*	0.95	0.95	0.95



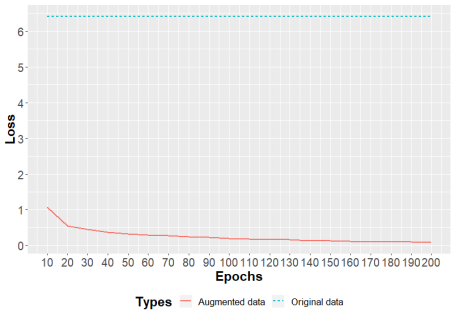
(a) Pick and Golf



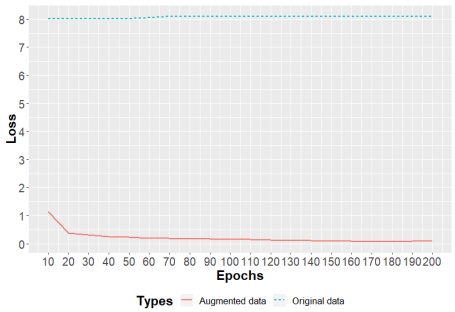
(b) Kick Ball and Punch



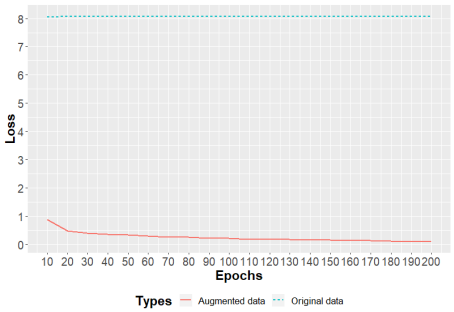
(c) Jump and Catch



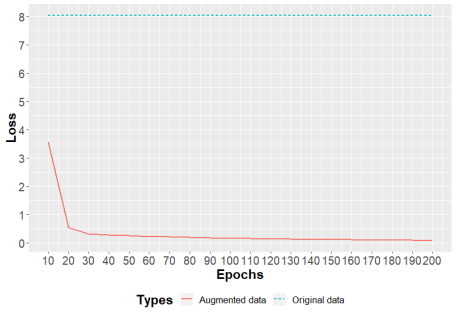
(d) Wave and Shake Hands



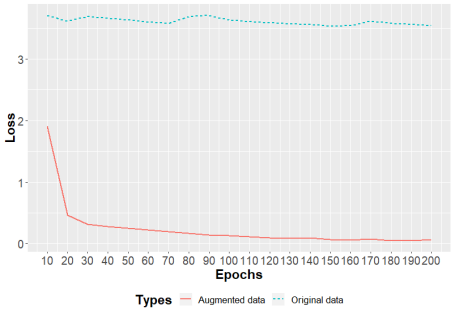
(e) Skiing and SkyDiving



(f) Rowing and Skyjet

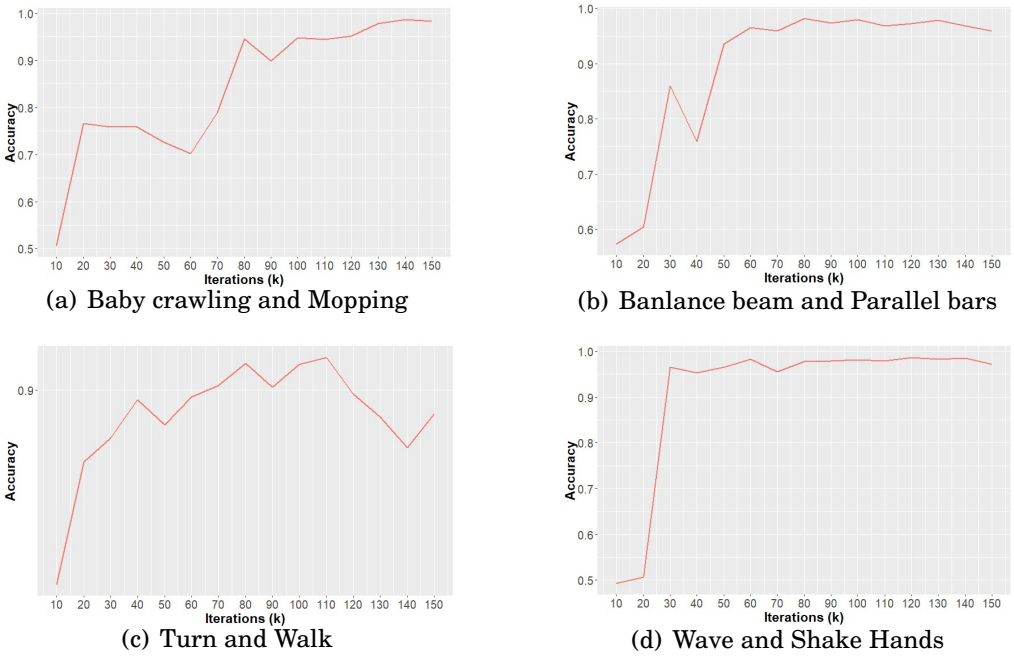


(g) Walking With Dog and Skate Boarding



(h) Yoyo and JugglingBalls

Figure 4.9: Training loss of the baseline CNN between original and augmented data [144]



(a) Baby crawling and Mopping

(b) Banlance beam and Parallel bars

(c) Turn and Walk

(d) Wave and Shake Hands

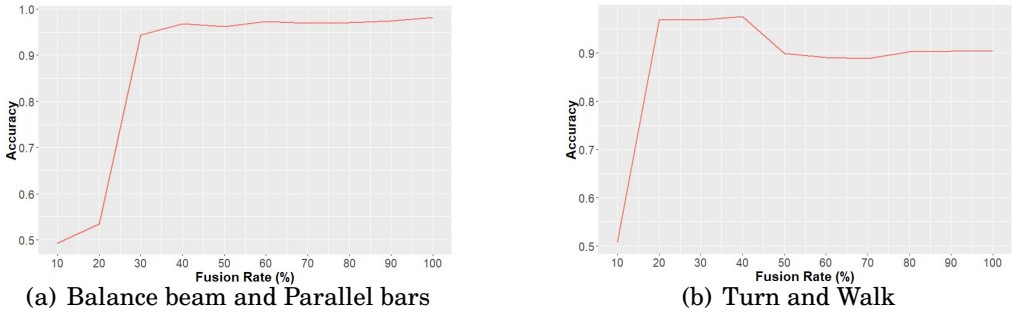
Figure 4.10: Accuracy changes based on the data obtained different GAN iterations [144]

tions, the image quality will be increased, which will affect the classification results.

It can be seen from Fig. 4.10 that the images created at 10000 iterations are almost same for every class, which cannot improve the classification. The image quality fluctuates between 20000 and 100000 iterations, and stabilizes after 100000 iterations for "Baby crawlin" and "Mopping". The highest performance for "Turn" and "Walk" is reached at 110000 iterations, after which it decreases slightly, which may be due to the loss of afterimage information from high quality images.

Performance on different fusion rates, different fusion rates will also affect the classification results. In this evaluation, 2000 frames were used from the original dataset, then gradually fuse the generated frames from 200 frames to 2000 frames. To ensure frame quality, the generated frames were applied from 100000 iterations.

Figure. 4.11 shows that performance will be increased after 400 frames have been added to the dataset and will stabilize after 600 frames have been added for "Balance Beam" and "Parallel Bars". In contrast, "Turn" and "Walk" reaches the highest performance after 800 frames have been added, then decreases at 1000 frames and stabilizes after the addition of 1000 frames.



(a) Balance beam and Parallel bars (b) Turn and Walk

Figure 4.11: Accuracy changes based on different fusion rates [144]

4.5 Discussion

The proposed data augmentation framework from similar gesture action recognition can improve the 2DCNN classfication performance due to the training data and bias have been enlarged. However, video-based action recognition now shift to 3DCNN methods which may extract more temporal features than the 2DCNN.

For the 3DCNN classification, the performance improved not as higher as 2DCNN. This is because the 3DCNN is designed for processing the features on continues frames and learn the correlation between frames. The future work could be using GAN to generate predicted videos which may improve the 3DCNN performance after the data augmentation.

4.6 Summary

In this chapter, an action data augmentation framework (ADAF) with a GAN features generator is proposed, which can enlarge the differences between similar class. The results on the baseline CNN are evaluated, which proves that the framework boost the performance of the classifier. Most existing human action recognition methods failed to fully leverage the differences and the internal connections of the similar gesture actions and the proposed ADAF overcome the challenges. The experimental results on three datasets demonstrated that our frameworks ADAF outperform the baseline CNN and can adapt to other methods using other CNNs. The future work will focus on how to use data augmentation to improve the performance of new 3DCNN-based methods such as video augmentation.

CHAPTER 5

GAN-BASED APPROACHES IN OTHER DOMAINS

In this chapter, the idea from the computer vision area is shifted to other domains to evaluate if the approaches can benefit other research areas. Autoencoder for image embedding is used to apply deep learning models to calculate feature vectors for images and return an enhanced data table with image descriptors. Similar to image embedding, graph embedding transfers the graphs to a vector or a set of vectors, which can be used to be retrieved in later tasks such as link prediction and clustering. The feature-dependent graph convolutional autoencoder is proposed to process the graph embedding. By applying the adversarial training method using GAN, the proposed frameworks not only exploit structural characteristics and node features, but also reconstructs both structural characteristics and node features, which naturally possess the interaction between these two sources of information while learning the embedding.

Major parts of this chapter have been accepted in the paper titled "Feature-Dependent Graph Convolutional Autoencoders with Adversarial Training Methods", Wu et al. [145].

5.1 Introduction

Graph techniques are widely applied to a variety of real-world scenarios, such as transportation, academic citation networks and social networks. Various data analysis tasks rely on analyzing graph data, for example, node or graph classification [59] [91], node clustering [126], and link prediction [140]. However, traditional machine learning techniques for graph data suffer from several challenges including high computational complexity,

low parallelizability, and inapplicability[23]. Recently graph embedding has become a vital solution to tackle these challenges.

The key idea of graph embedding is to learn the data distribution with a continuous and compact feature matrix which includes the original vertex content, network structure and side-information. Therefore, the traditional methods like linear classifier can be used to demonstrate the tasks such as link prediction, clustering and classification for various graph analytic purposes. [158][12].

Based on the underlying implementation, graph embedding algorithms can be categorized into three categories: matrix factorization algorithms, probabilistic model algorithms and deep learning algorithms. Matrix factorization-based algorithms used to compress the graph structure information as an adjacency matrix and extract the embedded graph information by decomposing the matrix, such as HOPE [87] and M-NMF [136]. Qiu et al. [98] proved that many probabilistic algorithms can be interpreted as matrix factorization-based methods.

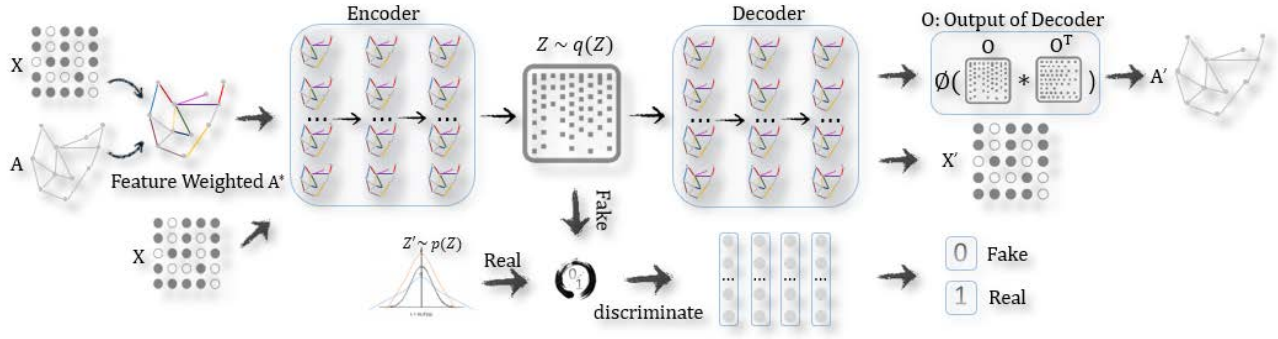


Figure 5.1: The proposed framework for graph embedding [145]

Probabilistic model-based algorithms aim to extract different patterns such as local neighborhood connectivities, global structural equivalence, and other various order proximities for learning graph embedding. These algorithms [96][118][41] are more effective and flexible for the large-scale graph data, compared to classical algorithms like spectral clustering [119].

Furthermore, the graph proximities and model positive point-wise mutual information (PPMI) can be preserved by the deep learning-based approaches. Methods such as SDNE [127] and DNGR [13] apply autoencoder-based frameworks for graph embedding, and Wang et al. proposed MGAE [126] for the clustering task, which learn the representation by leveraging a marginalized single-layer autoencoder to learn the representation.

Probabilistic methods mainly concentrate on preserving the structure relationship; by contrast, matrix factorization and deep learning methods aim to minimize the reconstruction error. However, they are all unregularized approaches, which mostly ignore the data distribution of the latent codes. Specifically, unregularized methods only learn the compressed identity mapping and ignore the structure of the graph [75], which could lead to unsatisfactory embedding when the graph data is sparse and noisy. To address this issue, Makhzani et al. [75] attempt to regularize the latent codes by enforcing them to follow certain prior distributions. Moreover, generative adversarial-based approaches have also been employed to learn robust latent representations under the adversarial training scheme [30] [99]. Pan et al. [89] proposed a novel adversarial framework with two variants, ARGA and ARVGA, which enforce the embedding to follow the prior distribution, while minimizing the reconstruction errors of the graph structure. However, their work does not reveal the inner-interdependency between structural characteristics and node features when processing the original graph into the framework. Additionally, the node features are not reconstructed in the decode stage, which cannot leverage the natural interaction between the node features and the topological structure of the graph when learning graph embedding.

In this chapter, first, an approach to preserve the natural interdependency between the structural characteristics and node features of a graph into a feature-dependent graph matrix (FGM) is proposed; then, given the FGM of a graph, a novel graph encoder-decoder framework (GED) and its variational version (VGED) are designed for graph embedding with a specially designed decoder which not only reconstruct the structural information, but also reconstruct content information in the decode stage. The special decoding scheme of the GED/VGED fully exploits the natural interaction during the training procedure while possessing the diffusion of the node features over the graph. The proposed GED offers high flexibility in relation to the choice of the encoder and decoder, such as the GCN, the graph attention network (GAT) or even using the inner production operation. In this work, GCN has been employed as the layers for the encoder and decoder, expecting to simultaneously expose the inner relationship between the structural characteristics and node attributes from two different angles: 1) the feature-dependent graph matrix reveals interdependency between the topological information and node content by normalizing and centralizing all the content of the graph and re-building the topological matrix based on both their content-distance between every two vertexes and their original linkages; and 2) while GCN is a variation of Laplacian smoothing, which propagates the features of a vertex to its neighbors. An adversarial

training scheme has been applied to enforce the embedding to follow a prior distribution which can enhance the robustness of the graph embedding.

The experiment results on three typical benchmark datasets show that the proposed GED outperforms the state-of-the-art research works. The contributions are summarized as follows:

- Two novel encoder-decoder frameworks (GED and VGED) are proposed for graph embedding, which simultaneously encode topological structure and content associated with nodes while exploiting the natural interdependency between two sources of information;
- The specially designed decoder of the proposed framework reconstructs both the topology and its relevant content information, which fully leverage the interaction between the different sources of information of a graph when learning the embedding;
- Experiments conducted on three benchmark graph datasets validate that the proposed GED approaches outperform its state-of-the-art peers on clustering and link prediction tasks.

5.2 Related Work

From the perspective of information analysis, there are two types of graph embedding methods namely topological-embedding-based methods and content-exploration-based embedding methods.

Topological-embedding approaches assume that only structural characteristics are accessible, and aim to preserve the structural characteristics maximumly. The DeepWalk model [96], proposed by Perozzi et al. in 2014 which using a group of random walks to learn the node embedding. Other probabilistic models such as LINE [118] and node2vec [41] have been developed to address similar problems. Moreover, a number of matrix factorization-based methods such as M-NMF [136] and HOPE [87] have been developed to learn the latent codes by representing a graph as a topological matrix mathematically. Furthermore, deep learning models have been introduced into the graph research area. Some of the models focus on preserving the first and second order of proximities [127], and the others apply variants of autoencoders to reconstruct the positive pointwise mutual information (PPMI) [13].

Furthermore, content-enhanced embedding approaches assume node features are accessible and apply both structural characteristics and the node features simultaneously. TADW [153] was designed to decompose the adjacency matrix to obtain the node embeddings. TriDNR [90] use a tri-party neural network to capture the structural characteristics, node features, and label information. In social networks, an approximated kernel mapping scheme, namely UPP-SNE [157], which enhance user embedding learning by using the user profiles.

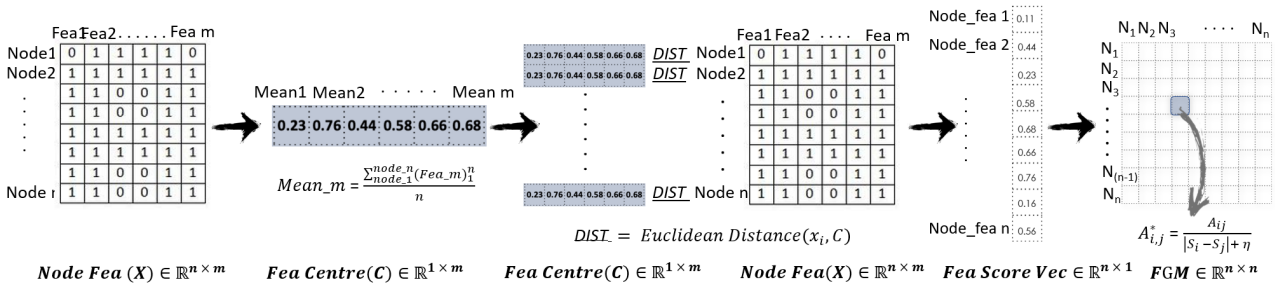


Figure 5.2: The framework of feature-dependent graph matrix [145]

Makhzani et al. [75] merged the adversarial mechanism into the autoencoder to learn the latent representation, namely the adversarial autoencoder (AAE) for general data. Dai et al. [24] proposed a framework with an adversarial training scheme for graph data. However, their approach only works on the structural characteristics. In contrast, a GCN has been applied to encode both the structural characteristics and node features into the low-dimensional graph embedding, and reconstruct the encoded information in the decoder stage. Such GED naturally collect and process the diffusion of the node features over the graph, which could improve the quality of the graph embedding.

Pan et al.[89] proposed an adversarially regularized graph autoencoder (ARGA) and the variational version (ARVGA) to address the issue of largely ignoring the latent code of the embedding. The key to their work is to apply the generative adversarial network (GAN) [40] to regularize the learned embedding, where the generator generates the graph embedding, and the discriminator determines whether the samples are from the prior distribution or the generator. Numerous adversarial algorithms such as those in [99] and [30] have been proposed, because of their effectiveness in unsupervised works such as the image or video classification and recognition.

The aforementioned approaches simultaneously encode both the structural characteristics and node features without considering the natural interdependency between these two perspectives of information. Additionally, these algorithms are incapable of

reconstructing the node features in the decode stage, hence they fail to comprehensively reconstruct the graph during the encoding-decoding procedure. In this paper, both the structural characteristics and the node features have been encoded into FGM to preserve the natural interdependency between these topological and content information, hence they the unique decoding scheme reconstructs both the structural characteristics and node features, which could greatly enhance the quality of the learned embedding.

5.3 Problem Definition

A graph can be defined as $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$, where $v_{i=1,\cdots,n}$ are vertices $\in \mathbf{V}$, and the edges can be represented as $e_{i,j} = (v_i, v_j) \in \mathbf{E}$. An adjacency matrix is used to represent the topological information of a graph. $\mathbf{A}_{i,j} = 1$ if the vertices have the edge $e_{i,j} \in \mathbf{E}$, otherwise $\mathbf{A}_{i,j} = 0$. $x_i \in \mathbf{X}$ denotes the content features of the node v_i.

Given a graph $\mathbf{G}$, the nodes $v_i \in \mathbf{V}$ are mapped into low-dimensional vectors $z_i \in \mathbb{R}^m$ which can be formatted as: $f : (\mathbf{A}, \mathbf{X}) \rightarrow \mathbf{Z}$, where $z_i^\top$ represents the i-th row if the matrix $\mathbf{Z} \in \mathbb{R}^{n \times m}$ where m and n indicates the dimension of the latent code and the number of hidden neurons respectively. The embedding matrix $\mathbf{Z}$ preserves the topological information from the structure $\mathbf{A}$ and the node features from $\mathbf{X}$.

5.4 Framework

The graph convolution encoder-decoders framework. The upper tier is a graph convolutional autoencoder taking a feature-dependent graph matrix $\mathbf{A}^*$ as input and attempts to reconstruct both structural characteristics $\mathbf{A}$ and node features $\mathbf{X}$ from the learned embedding $\mathbf{Z}$. A feature matrix $\mathbf{X}$ is additionally applied into the Encoder to further enhance the content associated with the graph into the embedding. The lower tier is an adversarial network, and the discriminator is trained to determine if a sample is generated by the embedding or from a prior distribution are shown in Fig 5.1

For given a graph $\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$, the objective of the proposed GCN encoder and decoder framework is to learn a robust embedding with both topological information and node features. To achieve this, a feature-dependent graph matrix is constructed to reveal the natural interdependency between topological structure $\mathbf{A}$ and node features $\mathbf{X}$, while a specially designed GCN decoder reconstructs both $\mathbf{A}$ and $\mathbf{X}$ to fully utilize the interaction between these two sources of information when learning the graph embedding.

- **Feature-dependent graph matrix**. Fig 5.2 shows the framework of the feature-dependent graph matrix. The feature center indicates the mean of each column in the node feature matrix **X**. Furthermore, the distance between the feature center and each node's features in **X** can be retrieved through the Euclidean distance, which can be persevered in a feature score vector. The final feature-dependent graph matrix is constructed based on the node_feature scores in the feature score vector.

- **Graph Convolutional Autoencoder**. The autoencoder learn the latent codes **Z** via the input of the feature-dependent graph matrix $\mathbf{A}^*$ and the node features **X**, by further reconstructing both the structural characteristics and node contents from the embedding.

- **Adversarial Network**. The embedding will follow the prior distribution through a well-trained adversarial module. The discriminator determines the embedding $z_i \in \mathbf{Z}$ is from the prior distribution or the encoder.

5.5 Algorithm

The GED is developed to embed a graph **G**, and map the nodes $v_i \in \mathbf{V}$ in a low-dimensional space. The proposed framework consists of three components: (1) feature-dependent graph matrix($\mathbf{A}^*$), (2) autoencoder, and (3) adversarial regulation.

5.5.1 Feature-dependent graph matrix ($\mathbf{A}^*$)

For given node features $\mathbf{X} \in \mathbb{R}^{n \times m}$, feature centre ($\mathbf{C} \in \mathbb{R}^{1 \times m}$) of the **X** represents the centred value of each feature in **X**. Each element in **C** can be calculated by Eq 5.1.

$$\mathbf{Mean\_m} = \frac{\sum_{node\_1}^{node\_n} (\mathbf{Fea\_m})_1^n}{n} \tag{5.1}$$

The feature score vector can be calculated via Eucli($\bullet$) = $\sqrt{\sum_{i=1}^{n} (v_i - \mathbf{C})^2}$ which is the Euclidean Distance between each node_feature vector v_i in **X** and **C**. Eq 5.2 shows the calculation of each fea_score.

$$fea\_score = Eucli(v_i, \mathbf{C}) \tag{5.2}$$

The final feature-dependent graph matrix $\mathbf{A}^*$ is constructed by fea_score of $\mathbf{S}$ between every two vertexes and their original linkages preserved in adjacency matrix $\mathbf{A}$. Specifically, if n_i and n_j indicate two nodes with an edge in $\mathbf{A}$ and s_i and $s_j \in \mathbf{S}$ represent the feature scores associated with node n_i and n_j, then the value of feature-dependent graph matrix $\mathbf{A^*}_{i,j}$ can be calculated as Eq 5.3.

$$\mathbf{A^*}_{i,j} = \frac{\mathbf{A}_{i,j}}{|s_i - s_j| + \eta} \tag{5.3}$$

where η is a very small number to ensure the denominator is not zero. After this, every value $a^*_{ij} \in \mathbf{A}^*$ was normalized to be between 0 and 1 through Eq 5.4.

$$normalized(a^*_{ij}) = \frac{a^*_{ij} - min(\mathbf{A}^*)}{max(\mathbf{A}^*) - min(\mathbf{A}^*)} \tag{5.4}$$

Graph Convolutional Encoder Model $\mathcal{G}(\mathbf{X},\mathbf{A}^*)$. A variant GCN has been developed [58] for both the encoder and decoder to represent the structural characteristics and node features in the framework. The idea of GCN extends from the spectral domain to the graph embedding rom the operational perspective, and the information can be learned by a convolution function $f(\mathbf{Z}^{(l)},\mathbf{A}^*|W^{(l)})$ from the transformation perspective.:

$$\mathbf{Z}^{(l+1)} = f(\mathbf{Z}^{(l)},\mathbf{A}^*|\mathbf{W}^{(l)}) \tag{5.5}$$

given $\mathbf{Z}^l$ is the input for the GCN, and the output is $\mathbf{Z}^{(l+1)}$. The embedding of the graph is represented as $\mathbf{Z}^0 = \mathbf{X} \in \mathbb{R}^{n \times m}$ in this paper, the number of features and the number of hidden neurons can be represented as m and n, respectively. The GCN attempts to learn the filter parameters of the $\mathbf{W}^{(l)}$ matrix. If $f(\mathbf{Z}^{(l)},\mathbf{A}^*|\mathbf{W}^{(l)})$ can be well defined, the deep convolutional neural networks can be constructed effectively.

The proposed GCN layers can be represented by $f(\mathbf{Z}^{(l)},\mathbf{A}^*|\mathbf{W}^{(l)})$ as below:

$$f(\mathbf{Z}^{(l)},\mathbf{A}^*|\mathbf{W}^{(l)}) = \phi(\widetilde{\mathbf{D}}^{-\frac{1}{2}}\widetilde{\mathbf{A}^*}\widetilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(l)}\mathbf{W}^{(l)}) \tag{5.6}$$

where ϕ indicates the activation function such as sigmoid() or Relu() function. $\widetilde{\mathbf{A}^*} = \mathbf{A}^* + \mathbf{I}$; $\widetilde{\mathbf{D}}_{ii} = \sum_j \widetilde{\mathbf{A}^*}_{i,j}$ and $\mathbf{I}$ is the identity matrix of $\mathbf{A}^*$. To sum up, the graph encoder $\mathcal{G}(\mathbf{X},\mathbf{A}^*)$ has a two-layer GCN structure.

The graph encoder can be mathematically expressed as follows:

$$\mathbf{Z}^{(1)} = f_{Relu}(\mathbf{X}, \mathbf{A}^* | \mathbf{W}^{(0)}); \tag{5.7}$$

$$Z^{(2)} = f_{linear}(\mathbf{Z}^{(1)}, \mathbf{A}^* | \mathbf{W}^{(1)}); \tag{5.8}$$

The first layer employs a Relu(•) as the activation function, while a linear function is used for the second layer. The encoder of the proposed GCN $\mathcal{G}(\mathbf{Z}, \mathbf{A}^*) = q(\mathbf{Z}|\mathbf{X}, \mathbf{A}^*)$ process the FGM ($\mathbf{A}^*$) which naturally contains the interdependency between the topological and content information , and the feature matrix $\mathbf{X}$ is additionally applied to further enhance the content associated with the graph into the embedding $q(\mathbf{Z}|\mathbf{X}, \mathbf{A}^*)$.

An inference model is used to define a variational encoder as follows:

$$q(\mathbf{Z}|\mathbf{X}, \mathbf{A}^*) = \prod_{i=1}^{n} q(z_i|\mathbf{X}, \mathbf{A}^*), \tag{5.9}$$

The average vectors z_i can constitute a matrix $\mu = Z^{(2)}$.

Decoder Model. The objective of our decoder model in the autoencoder is to decompress the latent codes and reconstruct both the structural characteristics and node features. Given a graph, the training data, consists of both topological and content information, therefore, the GCN decoder is naturally supposed to reconstruct both information, otherwise the information flow could be incomplete during the encoding-decoding process. The features amount of each vertex in the graph determine the dimensions of the second GCN layer, thus the second layer output will be in the range of $\mathbf{O} \in R^{n \times f}$ $Ni\mathbf{X}$. According to the conditions above, the final loss consists of two losses from the topology decoder and the feature decoder, respectively.

Topological Structure Decoder $d_t(\mathbf{Z})$. In the topological structure decoder, the original adjacency matrix $\mathbf{A}$ can be indirectly reconstructed from the embedding $\mathbf{Z}$ through the decompression operation of the decoder layers. Given an edge $e_{i,j} = (v_i, v_j) \in \mathbf{E}$, the model of the edge probability can be represented as $\mathbf{A}' = \phi(\mathbf{O}, \mathbf{O}^\top)$, where ϕ is an activation function like the sigmoid function as Eq 5.13 and $\mathbf{O}$ is the output from the GCN decoder layers.

$$\mathbf{Z}_D = f_{\text{linear}}(\mathbf{Z}, \mathbf{A} | \mathbf{W}_D^{(1)}). \tag{5.10}$$

$$\mathbf{O} = f_{\text{linear}}(\mathbf{Z}_D, \mathbf{A} | \mathbf{W}_D^{(2)}). \tag{5.11}$$

Algorithm 2 GED and VGED Algorithm

Require:
$\mathbf{G} = \{\mathbf{V}, \mathbf{E}, \mathbf{X}\}$;
T: iterations for updating;
K: steps for iterating discriminator;
m: the dimension of the latent variable
A^*: feature-dependent graph matrix
Ensure: $\mathbf{Z} \in R^{n \times m}$

1: **for** iterator = 1,2,3, $\cdots\cdots$, T **do**
2: Generate latent variables codes $\mathbf{Z}$ via Eq.(5.8);
3: **for** k = 1,2, $\cdots\cdots$, K **do**
4: Sample m entities $\{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(m)}\}$ from latent codes $\mathbf{Z}$
5: Sample m entities $\{\mathbf{a}^{(1)}, \ldots, \mathbf{a}^{(m)}\}$ from the prior distribution p_z
6: Update the discriminator with its stochastic gradient:

$$\bigtriangledown \frac{1}{m} \sum_{i=1}^{m} [\log \mathscr{D}(\mathbf{a}^i) + \log (1 - \mathscr{D}(\mathbf{z}^{(i)}))]$$

7: **end for**
8: Update the graph autoencoder with its stochastic gradient by Eq. (5.16) for GED or Eq. (5.17) for VGED;
9: **end for**
10: **return** $\mathbf{Z} \in R^{n \times m}$ =0

where the output of the encoder can be represented as $\mathbf{Z}$, and the first and second layer of the decoder output can be denoted as $\mathbf{Z}_D$ and $\mathbf{O}$. The number of hidden neurons is equal to the horizontal dimensions of $\mathbf{O}$.

$p(\mathbf{A}'|\mathbf{O})$ is the topological decoder where the links between nodes are available. Precisely, the inner production operation is used to train a link prediction layer to reconstruct the topology in the FGM:

$$p(\mathbf{A}'|\mathbf{O}) = \prod_{i=1}^{n} \prod_{j=1}^{n} p(\mathbf{A}'_{i,j}|\mathbf{o}_i, \mathbf{o}_j); \tag{5.12}$$

$$p(\mathbf{A}'_{i,j} = 1|\mathbf{o}_i, \mathbf{o}_j) = \text{sigmoid}(\mathbf{o}_i^\top, \mathbf{o}_j), \tag{5.13}$$

here the prediction $\mathbf{A}'$ should be close to the original top logical structure in FGM of the graph.

The reconstruction error can be calculated as follows:

$$\mathscr{L}_{A^*} = E_{q(\mathbf{O}|(\mathbf{X},\mathbf{A}^*))}[\log p(\mathbf{A}^*|\mathbf{O})] \tag{5.14}$$

Feature Decoder $d_f(Z)$. Then the node features reconstruction error can be computed as following equation:

$$\mathscr{L}_X = E_{q(\mathbf{Z}|(\mathbf{X},\mathbf{A}^*))}[\log p(\mathbf{X}|\mathbf{Z})]. \tag{5.15}$$

The final reconstruction error consists of topological decoder error and feature decoder error:

$$\mathscr{L}_0 = \mathscr{L}_{A^*} + \mathscr{L}_X. \tag{5.16}$$

Optimization. For the autoencoder optimization, the overall reconstruction error will be minimized by the equation which described in Eq 5.16 using gradient descent. The variational lower bound was optimized as follows for the variational encoder:

$$\mathscr{L}_1 = E_{q(\mathbf{Z}|(\mathbf{X},\mathbf{A}^*))}[\log\, p(\mathbf{A}^*,\mathbf{X}|\mathbf{Z})] - \mathbf{KL}[q(\mathbf{Z}|\mathbf{X},\mathbf{A}^*) \parallel p(\mathbf{Z})] \tag{5.17}$$

where the Kullback-Leibler divergence (also known as relative entropy) can be represented as $\mathbf{KL}[q(\bullet)||p(\bullet)]$ which is widely used to measure how much the distribution $q(\bullet)$ is different from $p(\bullet)$. In this chapter, $p(\bullet)$ is the prior distribution such as uniform distribution and Gaussian distribution. $p(\mathbf{Z}) = \prod_i p(\mathbf{z}_i) = \prod_i \mathscr{N}(\mathbf{z}_i|0,\mathbf{I})$ was used in this work.

5.5.2 Adversarial Mode $\mathscr{D}(Z)$

By applying the generative adversarial network (GAN) model, a graph embedding has been forced to match a prior distribution, where the latent distribution of the embedding can be regularized. Both generator G and discriminator D are standard multi-layer perceptron (MLP) and a single dimension sigmoid function is the output layer. The discriminator distinguishes whether the embedding is from the prior distribution p_z (real) or generator $\mathscr{G}(X,A*)$ (fake). The embedding will be regularized by minimizing the cross-entropy cost during the training process. The cost equation can be represented as follows:

$$-\frac{1}{2}E_{\mathbf{z}\sim p_z}\log\mathscr{D}(\mathbf{Z}) - \frac{1}{2}E_{\mathbf{X}}\log(1-\mathscr{D}(\mathscr{G}(\mathbf{X},\mathbf{A}))), \tag{5.18}$$

In this chapter, p_z denotes a simple Gaussian distribution.

Adversarial Graph Autoencoder Model. The training procedure of adversarial training scheme applied on encoder and Discriminator $\mathscr{D}(\mathbf{Z})$ are listed in the following equation:

$$\min_{\mathscr{G}}\max_{\mathscr{D}} E_{\mathbf{z}\sim p_z}[\log\mathscr{D}(\mathbf{Z})] + E_{\mathbf{x}\sim p(\mathbf{x})}[\log(1-\mathscr{D}(\mathscr{G}(\mathbf{X},\mathbf{A}*)))] \tag{5.19}$$

where $\mathscr{G}(\mathbf{X},\mathbf{A}*)$ denotes the generator and $\mathscr{D}(\mathbf{Z})$ is the discriminator explained above.

5.5.3 Algorithm Explanation

Algorithm 2 demonstrates the workflow of the proposed GED and VGED. Given a graph **G** with vertex set V, edges E and node feature matrix X, latent codes **Z** is generated by the GCN encoder in step 2. Then the generated **Z** will be sampled in step 4 as well as the prior distribution p_z in step 5. Step 6 update the discriminator according to the cross-entropy loss. After K iterations, the GCN encoder generate embedding and the discriminator will discriminate the embedding is ture or fake. The stochastic gradient will be used to update encoder.The **GED** can be trained by the updated Eq. (5.16), or the **VGED** can be trained with Eq. (5.17). Finally, the graph embedding $\mathbf{Z} \in R^{n \times m}$ will be returned in step 9.

5.6 Experimental results

The proposed approaches are evaluated with two unsupervised graph analytic tasks: linkage prediction and node clustering on three scientific publications datasets. Table 5.1 details three benchmark graph datasets used in the experiments. The scientific publications and citation relationships can be represented as nodes and edges respectively. The unique words in each scientific article are the features.

Table 5.1: Graph Datasets [145]

Dataset	Nodes	Linkages	Total Words	Features
Cora	2,708	5,429	3,880,564	1,433
Citeseer	3,327	4,732	12,274,336	3,703
PubMed	19,717	44,338	9,858,500	500

5.6.1 Experimental results on link prediction

Baseline methods. The state-of-art algorithms are choosen as the baseline methods in comparing with the performance on the link prediction task with proposed GED and GED:

- **ARGA**[89]: ARGA uses the adversarially regularized autoencoder algorithm to learn the embedding.
- **ARVGA**[89]: ARVGA is a variational ARGA to represent the graph.

- **GAE**[59]: is an unsupervised approach which learns meaningful latent embedding from both the structural characteristics and node features.
- **VGAE**[59]: is a variational GAE for network representation with a network structure and features information.
- **DeepWalk**[96]: learns the features from the captured structural characteristics independent of the label distribution.
- **Spectral Clustering**[119]: is a classification framework to learn the description of a plausible affiliation of users based on latent social dimensions.
- **GED** Our proposed algorithm, which naturally merges structural characteristics and node features into a feature-dependent graph matrix, and reconstructs topological and node features simultaneously.
- **VGED** Our proposed algorithm, which is a variational version of GED.

Evaluation metrics. The metrics used in this chapter are average precision (AP) and AUC score (the area under a receiver operating characteristic curve). The final score is the mean values with the standard errors after running each experiment 20 times. Each dataset has been set into a training set (85%), testing set (10%) and validation set (5%), for the purpose of training, verification and optimization respectively.

Parameter Settings. The autoencoder models are trained for 50 iterations and apply Adam algorithm as optimization algorithm for the clustering task, meanwhile. 200 iterations were set for the link prediction. The learning rate is 0.005 and the learning rate of discriminator are set at 0.001. In view of the large size of the PubMed, the iterations are set as 500 for clustering and 2000 for link prediction. All the experiments includ a 32-neuron embedding layer and 32-neuron hidden layer in the encoder. The number of neurons in the decoder are dynamic. Specifically, the number of the neurons are equal to the the number of the features associated with each vertex in the graph, thus the node feature matrix can be reconstructed in the decode stage. Lastly, the discriminator comprises two hidden layers of 32-neurons and 64-neurons respectively. For comparison, the parameter settings are provided in the corresponding chapter.

Experimental results. Table 5.2 shows the experiment results on the link prediction task. All the baselines and proposed models have been run 20 times and report the mean and standard deviation of their performances. The results show that our GED and VGED achieve an outstanding performance: both AUC score and AP are as high as 93% on all

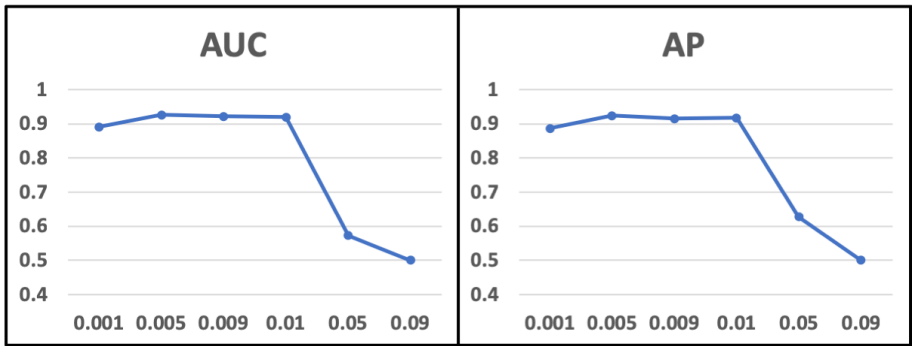
Table 5.2: Results for Link Prediction. GAE* and VGAE* are variants of GAE and VGAE, which only explore topological structure, i.e., $\mathbf{X} = \mathbf{I}$. [145]

Approaches	Cora		Citeseer		PubMed	
	AUC	AP	AUC	AP	AUC	AP
SC	84.6 ± 0.01	88.5 ± 0.00	80.5 ± 0.01	85.0 ± 0.01	84.2 ± 0.02	87.8 ± 0.01
DW	83.1 ± 0.01	85.0 ± 0.00	80.5 ± 0.02	83.6 ± 0.01	84.4 ± 0.00	84.1 ± 0.00
GAE*	84.3 ± 0.02	88.1 ± 0.01	78.7 ± 0.02	84.1 ± 0.02	82.2 ± 0.01	87.4 ± 0.00
VGAE*	84.0 ± 0.02	87.7 ± 0.01	78.9 ± 0.03	84.1 ± 0.02	82.7 ± 0.01	87.5 ± 0.01
GAE	91.0 ± 0.02	92.0 ± 0.03	89.5 ± 0.04	89.9 ± 0.05	96.4 ± 0.00	96.5 ± 0.00
VGAE	91.4 ± 0.01	92.6 ± 0.01	90.8 ± 0.02	92.0 ± 0.02	94.4 ± 0.02	94.7 ± 0.02
ARGA	92.4 ± 0.003	93.2 ± 0.003	91.9 ± 0.003	93.0± 0.003	96.8 ± 0.001	97.1 ± 0.001
ARVGA	92.4 ± 0.004	92.6 ± 0.004	92.4 ± 0.003	93.0 ± 0.003	96.5± 0.001	96.8± 0.001
GED	93.1 ± 0.003	93.1 ± 0.003	93.3 ± 0.003	**93.4**± 0.003	**97.1** ± 0.001	**97.2** ± 0.001
VGED	**94.1** ± 0.003	**94.2** ± 0.003	**94.6** ± 0.003	93.3± 0.003	96.9 ± 0.001	97.0 ± 0.001

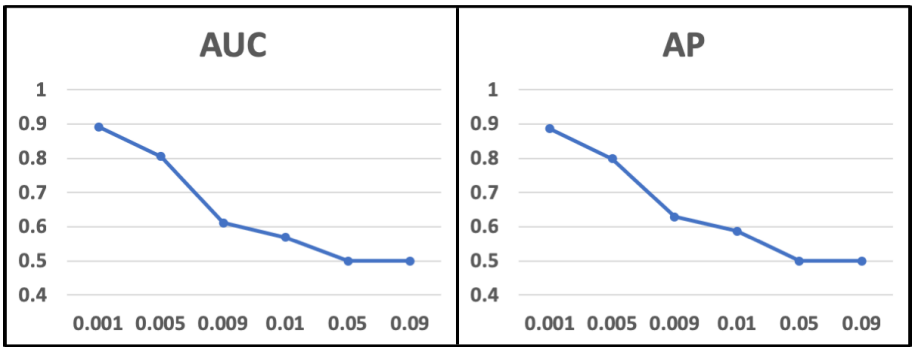
the three benchmark datasets. Compared to the listed baseline models, AP score has increased around 2.6% compared with VGAE incorporating the node features, and 11.1% higher than the VGAE without the node features. The proposed GED also lead 15.6% improvement compared with DeepWalk and 10.7% improvement compared with Spectral Clustering on the PubMed dataset, respectively. Which means the features may effect the prediction of the future connections in these datasets. The results of GED and VGED outperformed TADW. The AUC of our model GED achieves 92.6%, 93.3% and 97.1% and VGED achieves 94.1%, 94.6% and 96.9% in Cora, Citeseer and PubMed respectively. This is because TADW fails to fully exploit the inner-interdependency between the structural characteristics and node features, bacause TADW is a shallow linear model. The GED and VGED also significantly outperform both ARGA and ARVGA both of which only reconstruct the structural characteristics. The experiments show that the FGM and reconstructing both topological and content information are crucial for improving graph embedding.

Parameter Sensitivity. Learning rate from 0.001 to 0.9 is varied for evaluating the parameter sensitivity and Fig 5.3 shows the report results.

The report in both Fig 5.3 AUC and AP show similar trends: when the learning rate is increased from 0.001 to 0.005, the performance reaches its peak, then it stabilises until the learning rate has been increased to 0.01. The performance dramatically drops after 0.01 from around 90% to 50%. Similarly, the performance of the learning rate drops from the discriminator learning rate 0.001 to 0.09. This is because the learning rate determines the speed of the weight updating, however, the large learning rate will result



(a) Learning rate



(b) Discriminator learning rate

Figure 5.3: Average performance on (a) learning rate and (b) discriminator learning rate on the Cora dataset for AUC and AP. [145]

in the results exceeding the optimal value.

It is worth mentioning that when the learning rate is set between 0.005 to 0.01 and the discriminator learning rate is set at 0.001, the framework achieved its best performance.

5.6.2 Experimental results on node clustering

The K-means clustering algorithm is applied on the learned embedding, for the node clustering task.

Baseline methods. The embedding approaches and clustering approaches baselines which designed for clustering are compared:

- **DNGR**[13]: is able to learn a weighted graph embedding.
- **TADW**[153]: associates DeepWalk to factorizes a matrix and solve the close form of the matrix.

- **RMSC**[149]: is a Markov chain to transfer the noise between the multi-view transition probability matrices.
- **Graph Encoder**[120]: takes the sparse autoencoder as the building block, then generates the non-linear graph embedding.
- **RTM**[15]: explicitly ties the documents and the content.
- **K-means**: is an old classic method which can be the basis of many other clustering algorithms.

The K-means, Graph Encoder, and DNGR only exploit the topological structures of the graph, while the rest of the algorithms apply both topological structures and node features for the graph clustering task.

Evaluation metrics. Five metrics are used to evaluate the clustering results: normalized mutual information (NMI), accuracy (ACC), precision, F-score (F1) and average and index (ARI), which are defined by Xia et al. [149].

Table 5.3: Clustering Results on Cora [145]

Cora	Acc	NMI	F1	Precision	ARI
K-means	0.492	0.321	0.368	0.369	0.230
Spectral	0.367	0.127	0.318	0.193	0.031
GraphEncoder	0.325	0.109	0.298	0.182	0.006
DeepWalk	0.484	0.327	0.392	0.361	0.243
DNGR	0.419	0.318	0.340	0.266	0.142
RTM	0.440	0.230	0.307	0.332	0.169
RMSC	0.407	0.255	0.331	0.227	0.090
TADW	0.560	0.441	0.481	0.396	0.332
GAE	0.596	0.429	0.595	0.596	0.347
VGAE	0.609	0.436	0.609	0.609	0.346
ARGA	0.640	0.449	0.619	0.646	0.352
ARVGA	0.638	0.450	0.627	0.624	0.374
GED	0.679	0.462	0.660	0.654	0.437
VGED	**0.695**	**0.515**	**0.695**	**0.696**	**0.473**

Experimental results. Table 5.3, Table 5.4, and Table 5.5 listed results of the clustering task on the three benchmark datasets Cora, Citeseer and PubMed, respectively. The results show that the proposed GED and VGED outperform on all the metrics compared with baseline models. For instance, on Citeseer, the accuracy of VGED has

Table 5.4: Clustering Results on Citeseer [145]

Citeseer	Acc	NMI	F1	Precision	ARI
K-means	0.540	0.305	0.409	0.405	0.279
Spectral	0.239	0.056	0.299	0.179	0.010
GraphEncoder	0.225	0.033	0.301	0.179	0.010
DeepWalk	0.337	0.088	0.270	0.248	0.092
DNGR	0.326	0.180	0.300	0.200	0.044
RTM	0.451	0.239	0.342	0.349	0.203
RMSC	0.295	0.139	0.320	0.204	0.049
TADW	0.455	0.291	0.414	0.312	0.228
GAE	0.408	0.176	0.372	0.418	0.124
VGAE	0.344	0.156	0.308	0.349	0.093
ARGA	0.573	0.350	0.546	0.573	0.341
ARVGA	0.544	0.261	0.529	0.549	0.245
GED	0.555	0.277	0.540	**0.565**	0.253
VGED	**0.581**	**0.338**	**0.581**	0.537	**0.302**

Table 5.5: Clustering Results on PubMed [145]

PubMed	Acc	NMI	F1	Precision	ARI
K-means	0.393	0.001	0.192	0.574	0.001
Spectral	0.396	0.037	0.273	0.493	0.002
GraphEncoder	0.	0.0	0.	0.	0.
DeepWalk	0.679	0.273	0.672	0.693	0.302
DNGR	0.453	0.151	0.462	0.631	0.051
RTM	0.571	0.192	0.445	0.459	0.143
RMSC	0.0	0.0	0.0	0.0	0.0
TADW	0.353	0.001	0.331	0.333	0.001
GAE	0.675	0.279	0.662	0.687	0.273
VGAE	0.632	0.231	0.636	0.633	0.217
ARGA	0.665	0.302	0.637	0.673	0.302
ARVGA	0.687	0.293	0.672	0.688	0.311
GED	0.678	**0.311**	0.640	0.692	0.313
VGED	**0.692**	0.303	**0.681**	**0.701**	**0.320**

been improved around 158.2% compared with GraphEncoder and 7.6% compared with K-means ; in addition, the F1 score of VGED increased the 15.2% performance compared with DeepWalk and 40.3% compared with TADW. The improvement of the results between GED and ARGA further proves the superiority of our FGM and node feature reconstruction. Furthermore, the clustering results from DeepWalk only consider graph information from single perspective, which lead the low performance compared to the methods which consider both the structural characteristics and the node features. By applying FGM and reconstructing the node features, our algorithms outperform the state-of-art algorithms ARGA and ARVGA, compared with ARGE, VGED which increases the accuracy to 8.5%, 1.3% and 4% in Cora, Citeseer and PubMed, respectively.

5.7 Summary

In this chapter, the GAN and autoencoder from computer vision were shifted to the graph research area. An adversarial graph encoder-decoder framework is developed with a specially designed decoder to simultaneously reconstruct the structural characteristics and content information associated with nodes when learning graph embedding. A novel method to construct both topological information and node features into a feature-dependent graph matrix (FGM) is developed while preserving the interdependency between these two sources of information. It is worth to indicate that most existing graph embedding approaches fail to fully leverage the natural connection between graph structure and node features and are not able to reconstruct the node features during the training procedure, which may result in an unsatisfactory embedding. The proposed frameworks and graph construction method have smoothly overcome these challenges. The experiment results on three real-world graph datasets demonstrate that the proposed frameworks, GED and VGED with FGM as the input outperform their peers in link prediction and node clustering tasks.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter concludes the thesis and provides some direction for future work. The aim of the thesis is to explore the similar gesture recognition problems which decrease the performance of the CNNs. The aim the study is also to come up with an automatic and robust human action recognition framework. I have adopted/proposed several computer vision and deep learning approaches to build intelligent solutions for understanding the human actions.

In the first part of the thesis, I adopted the hierarchical classification approach for similar gesture action recognition, which can apply multi-stage classification for similar gesture action recognition. Thus the classifier will not interfered by other classes. Following the hierarchical classification approach, an adversarial action data augmentation framework was used to generate video frames which can improve the classification. Moreover, the generated frames includes more features than the original frames which can enlarge the differences and bias for different classes.

In the second part of the thesis, I adopted the GAN approaches to generate the video frames, which can enlarge the training set and bias between classes. The proposed end-to-end framework can identify similar gesture classes and generate the frames for these classes automatically. By augmenting the generated frames with original frames, the classification performance was improved.

In the third part of the thesis, I adopted the GAN-based approaches to the graph embedding, which can merge the structural characteristics and node features according to their interdependency and reconstructs both structural characteristics and node

features. Experiments where conducted on three real-world graph datasets such as Cora, Citeseer and PubMed to evaluate the proposed framework and algorithms, and the results outperform baseline methods on both link prediction and graph clustering tasks.

6.1 Summary of the thesis

- Chapter 2 presents an extensive literature review about state-of-the-art methods published related to human action recognition using computer vision and deep learning approaches. The human action datasets covers two aspects of single-view datasets and multi-view datasets. It was found in the literature that a large body of work has been published using hand-crafted features to represent the human actions pros and cons of these approaches discussed in greater detail. In single-view datasets, there is only one camera capturing the human actions, while the information is incomplete, while a multi-view dataset has multiple view angles so that the occlusion has been solved in multi-view datasets. Additionally some of the approaches could also be used to generate features for different classifiers. Most early traditional machine learning works are problem dependent, which apply the texture descriptors on the extracted handcraft motion features.

- Chapter 3 presents a new approach to handle the actions with similar gestures to improve the overall accuracy of a gesture recognition system. Analysis showed that a major reason for low performance is due to the confusion among the similar gestures. Hence, we focus on resolving the confusion among the class with similar gestures, in the current work. A generic hierarchical classification model is proposed in this work, which can be applied to any datasets/real-world application involving gesture recognition.

- Chapter 4 presents an action data augmentation framework with a GAN features generator, which can enlarge the differences between similar class. Firstly, the original video frames were set as the input of the GAN, then generate the new frames for each action videos; followed by the data augmentation process, combining the original frames and generated frames; the final stage sends the augmented frames as the CNN input and obtains the result. The results on CNN-based methods have been evaluated, which proves the framework indeed boost the performance of the classifier.

- Chapter 5 presents novel encoder-decoder frameworks (GED and VGED) for graph embedding, which simultaneously encode topological structures and content associated with nodes while exploiting the natural interdependency between two sources of information. The specially designed decoder of the proposed framework reconstructs both the topology and its relevant content information, which fully leverage the interaction between the different sources of information of a graph when learning the embedding.

6.2 Future research

Various approaches for video-based human action recognition were investigated in this thesis with different aspects of human actions. Several areas are identified for further research which is summarised as follows.

- The GAN is used to generate still images which can significantly improve 2DCNN-based performance. However, the performance improvement using 3DCNN approaches are not siginificant enough, as the GAN generated frames loses the correlation with the original frames. This will be a require a video generating method to generate sequence video to improve the 3DCNN performance.
- Multi-human action recognition could be another direction which is a very challenging task.
- Another extension to the action recognition framework is to modify current state-of-the-art gesture estimation methods, which can extract more features for classification.
- Discovering more features between frames could improve the classification performance. Furthermore, new detection and recognition methods can be developed with higher performance and lower costs.

BIBLIOGRAPHY

[1] J. K. AGGARWAL AND M. S. RYOO, *Human activity analysis: A review*, ACM Computing Surveys (CSUR), 43 (2011), p. 16.

[2] A. ALI AND J. AGGARWAL, *Segmentation and recognition of continuous human activity*, in Proceedings IEEE Workshop on Detection and Recognition of Events in Video, IEEE, 2001, pp. 28–35.

[3] K. H. ALI AND T. WANG, *Learning features for action recognition and identity with deep belief networks*, in Audio, Language and Image Processing (ICALIP), 2014 International Conference on, IEEE, 2014, pp. 129–132.

[4] A. ANTONIOU, A. STORKEY, AND H. EDWARDS, *Data augmentation generative adversarial networks*, arXiv preprint arXiv:1711.04340, (2017).

[5] M. ARJOVSKY, S. CHINTALA, AND L. BOTTOU, *Wasserstein gan*, arXiv preprint arXiv:1701.07875, (2017).

[6] M. BACCOUCHE, F. MAMALET, C. WOLF, C. GARCIA, AND A. BASKURT, *Sequential deep learning for human action recognition*, in International Workshop on Human Behavior Understanding, Springer, 2011, pp. 29–39.

[7] N. BALLAS, L. YAO, C. PAL, AND A. COURVILLE, *Delving deeper into convolutional networks for learning video representations*, International Conference of Learning Representations, (2016).

[8] B. BANERJEE AND V. MURINO, *Efficient pooling of image based cnn features for action recognition in videos*, in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE, 2017, pp. 2637–2641.

[9] M. BLANK, L. GORELICK, E. SHECHTMAN, M. IRANI, AND R. BASRI, *Actions as space-time shapes*, in Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2, IEEE, 2005, pp. 1395–1402.

[10] A. F. BOBICK AND J. W. DAVIS, *The recognition of human movement using temporal templates*, IEEE Transactions on pattern analysis and machine intelligence, 23 (2001), pp. 257–267.

[11] C. BOWLES, L. CHEN, R. GUERRERO, P. BENTLEY, R. GUNN, A. HAMMERS, D. A. DICKIE, M. V. HERNÁNDEZ, J. WARDLAW, AND D. RUECKERT, *Gan augmentation: Augmenting training data using generative adversarial networks*, arXiv preprint arXiv:1810.10863, (2018).

[12] H. CAI, V. W. ZHENG, AND K. CHANG, *A comprehensive survey of graph embedding: problems, techniques and applications*, IEEE Transactions on Knowledge and Data Engineering, (2018).

[13] S. CAO, W. LU, AND Q. XU, *Deep neural networks for learning graph representations.*, in AAAI, 2016, pp. 1145–1152.

[14] J. CARREIRA AND A. ZISSERMAN, *Quo vadis, action recognition? a new model and the kinetics dataset*, in proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[15] J. CHANG AND D. BLEI, *Relational topic models for document networks*, in Artificial Intelligence and Statistics, 2009, pp. 81–88.

[16] L.-C. CHEN, G. PAPANDREOU, I. KOKKINOS, K. MURPHY, AND A. L. YUILLE, *Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, IEEE transactions on pattern analysis and machine intelligence, 40 (2018), pp. 834–848.

[17] M.-Y. CHEN AND A. HAUPTMANN, *Mosift: Recognizing human actions in surveillance videos*, CMU-CS-09-161, (2009).

[18] Y. CHEN, Z. LI, X. GUO, Y. ZHAO, AND A. CAI, *A spatio-temporal interest point detector based on vorticity for action recognition*, in Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on, IEEE, 2013, pp. 1–6.

[19] G. CHÉRON, I. LAPTEV, AND C. SCHMID, *P-cnn: Pose-based cnn features for action recognition*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3218–3226.

[20] F. CHOLLET ET AL., *Keras*.
https://github.com/fchollet/keras, 2015.

[21] K.-P. CHOU, M. PRASAD, D. WU, N. SHARMA, D.-L. LI, Y.-F. LIN, M. BLUMENSTEIN, W.-C. LIN, AND C.-T. LIN, *Robust feature-based automated multi-view human action recognition system*, IEEE Access, 6 (2018), pp. 15283–15296.

[22] R. COLLOBERT AND J. WESTON, *A unified architecture for natural language processing: Deep neural networks with multitask learning*, in Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167.

[23] P. CUI, X. WANG, J. PEI, AND W. ZHU, *A survey on network embedding*, IEEE Transactions on Knowledge and Data Engineering, (2018).

[24] Q. DAI, Q. LI, J. TANG, AND D. WANG, *Adversarial network embedding*, arXiv preprint arXiv:1711.07838, (2017).

[25] G. DENINA, B. BHANU, H. T. NGUYEN, C. DING, A. KAMAL, C. RAVISHANKAR, A. ROY-CHOWDHURY, A. IVERS, AND B. VARDA, *Videoweb dataset for multi-camera activities and non-verbal communication*, in Distributed Video Sensor Networks, Springer, 2011, pp. 335–347.

[26] K. G. DERPANIS, M. SIZINTSEV, K. J. CANNONS, AND R. P. WILDES, *Action spotting and recognition based on a spatiotemporal orientation analysis*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 527–540.

[27] A. DIBA, M. FAYYAZ, V. SHARMA, A. H. KARAMI, M. M. ARZANI, R. YOUSEFZADEH, AND L. VAN GOOL, *Temporal 3d convnets: New architecture and transfer learning for video classification*, arXiv preprint arXiv:1711.08200, (2017).

[28] P. DOLLÁR, V. RABAUD, G. COTTRELL, AND S. BELONGIE, *Behavior recognition via sparse spatio-temporal features*, in 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, IEEE, 2005, pp. 65–72.

[29] J. DONAHUE, L. ANNE HENDRICKS, S. GUADARRAMA, M. ROHRBACH, S. VENUGOPALAN, K. SAENKO, AND T. DARRELL, *Long-term recurrent convolutional*

networks for visual recognition and description, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2625–2634.

[30] J. DONAHUE, P. KRÄHENBÜHL, AND T. DARRELL, *Adversarial feature learning*, arXiv preprint arXiv:1605.09782, (2016).

[31] A. EWEIWI, S. CHEEMA, C. THURAU, AND C. BAUCKHAGE, *Temporal key poses for human action recognition*, in 2011 IEEE international conference on computer vision workshops (ICCV Workshops), IEEE, 2011, pp. 1310–1317.

[32] A. FATHI AND G. MORI, *Action recognition by learning mid-level motion features*, in 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[33] C. FEICHTENHOFER, A. PINZ, AND A. ZISSERMAN, *Convolutional two-stream network fusion for video action recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1933–1941.

[34] B. FERNANDO, E. GAVVES, J. M. ORAMAS, A. GHODRATI, AND T. TUYTELAARS, *Modeling video evolution for action recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5378–5387.

[35] J. GALL, A. YAO, N. RAZAVI, L. VAN GOOL, AND V. LEMPITSKY, *Hough forests for object detection, tracking, and action recognition*, IEEE transactions on pattern analysis and machine intelligence, 33 (2011), pp. 2188–2202.

[36] Z. GAO, M.-Y. CHEN, A. G. HAUPTMANN, AND A. CAI, *Comparing evaluation protocols on the kth dataset*, in International Workshop on Human Behavior Understanding, Springer, 2010, pp. 88–100.

[37] A. GILBERT, J. ILLINGWORTH, AND R. BOWDEN, *Action recognition using mined hierarchical compound features*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33 (2011), pp. 883–897.

[38] S. S. GIRIJA, *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, (2016).

[39] N. GKALELIS, H. KIM, A. HILTON, N. NIKOLAIDIS, AND I. PITAS, *The i3dpost multi-view and 3d human action/interaction database*, in Visual Media Production, 2009. CVMP'09. Conference for, IEEE, 2009, pp. 159–168.

[40] I. GOODFELLOW, J. POUGET-ABADIE, M. MIRZA, B. XU, D. WARDE-FARLEY, S. OZAIR, A. COURVILLE, AND Y. BENGIO, *Generative adversarial nets*, in Advances in neural information processing systems, 2014, pp. 2672–2680.

[41] A. GROVER AND J. LESKOVEC, *node2vec: Scalable feature learning for networks*, in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 855–864.

[42] A. GRUSHIN, D. D. MONNER, J. A. REGGIA, AND A. MISHRA, *Robust human action recognition via long short-term memory*, in Neural Networks (IJCNN), The 2013 International Joint Conference on, IEEE, 2013, pp. 1–8.

[43] K. GUO, P. ISHWAR, AND J. KONRAD, *Action recognition from video using feature covariance matrices*, IEEE Transactions on Image Processing, 22 (2013), pp. 2479–2494.

[44] K. HARA, H. KATAOKA, AND Y. SATOH, *Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?*, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 6546–6555.

[45] S. HAUBERG, O. FREIFELD, A. B. L. LARSEN, J. FISHER, AND L. HANSEN, *Dreaming more data: Class-dependent distributions over diffeomorphisms for learned data augmentation*, in Artificial Intelligence and Statistics, 2016, pp. 342–350.

[46] D. HE, Z. ZHOU, C. GAN, F. LI, X. LIU, Y. LI, L. WANG, AND S. WEN, *Stnet: Local and global spatial-temporal modeling for action recognition*, arXiv preprint arXiv:1811.01549, (2018).

[47] K. HE, X. ZHANG, S. REN, AND J. SUN, *Deep residual learning for image recognition*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[48] G. E. HINTON AND R. R. SALAKHUTDINOV, *Reducing the dimensionality of data with neural networks*, science, 313 (2006), pp. 504–507.

[49] K. HUANG, D. TAO, Y. YUAN, X. LI, AND T. TAN, *View-independent behavior analysis*, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 39 (2009), pp. 1028–1035.

[50] Q. HUANG, S. SUN, AND F. WANG, *A compact pairwise trajectory representation for action recognition*, in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE, 2017, pp. 1767–1771.

[51] A. IOSIFIDIS, A. TEFAS, AND I. PITAS, *Neural representation and learning for multi-view human action recognition*, in The 2012 International Joint Conference on Neural Networks (IJCNN), IEEE, 2012, pp. 1–6.

[52] A. A. J. JOHNSON AND F. LI, *Perceptual losses for real-time style transfer and super-resolution*, in In European Conference on Computer Vision (ECCV), 2016.

[53] S. JI, W. XU, M. YANG, AND K. YU, *3d convolutional neural networks for human action recognition*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 221–231.

[54] Z. JIANG, Z. LIN, AND L. DAVIS, *Recognizing human actions by learning and matching shape-motion prototype trees*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 533–547.

[55] J. JOHNSON, A. ALAHI, AND L. FEI-FEI, *Perceptual losses for real-time style transfer and super-resolution*, in European Conference on Computer Vision, Springer, 2016, pp. 694–711.

[56] W. KAY, J. CARREIRA, K. SIMONYAN, B. ZHANG, C. HILLIER, S. VIJAYANARASIMHAN, F. VIOLA, T. GREEN, T. BACK, P. NATSEV, ET AL., *The kinetics human action video dataset*, arXiv preprint arXiv:1705.06950, (2017).

[57] Y. KE, R. SUKTHANKAR, M. HEBERT, ET AL., *Efficient visual event detection using volumetric features.*, in ICCV, vol. 1, 2005, pp. 166–173.

[58] T. N. KIPF AND M. WELLING, *Semi-supervised classification with graph convolutional networks*, arXiv preprint arXiv:1609.02907, (2016).

[59] ——, *Variational graph auto-encoders*, arXiv preprint arXiv:1611.07308, (2016).

[60] A. KOLESNIKOV AND C. H. LAMPERT, *Seed, expand and constrain: Three principles for weakly-supervised image segmentation*, in In European Conference on Computer Vision (ECCV), 2016.

[61] H. KUEHNE, H. JHUANG, E. GARROTE, T. POGGIO, AND T. SERRE, *HMDB: a large video database for human motion recognition*, in Proceedings of the International Conference on Computer Vision (ICCV), 2011.

[62] H. KUEHNE, H. JHUANG, E. GARROTE, T. POGGIO, AND T. SERRE, *Hmdb: a large video database for human motion recognition*, in 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2556–2563.

[63] M. N. KUMAR AND D. MADHAVI, *Improved discriminative model for view-invariant human action recognition*, Int. J. Comput. Sci. Eng. Technol., 4 (2013), pp. 1263–1270.

[64] Z. LAN, Y. ZHU, A. G. HAUPTMANN, AND S. NEWSAM, *Deep local video feature for action recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1–7.

[65] I. LAPTEV, *On space-time interest points*, International Journal of Computer Vision, 64 (2005), pp. 107–123.

[66] I. LAPTEV, B. CAPUTO, ET AL., *Recognizing human actions: a local svm approach*, in null, IEEE, 2004, pp. 32–36.

[67] I. LAPTEV, M. MARSZALEK, C. SCHMID, AND B. ROZENFELD, *Learning realistic human actions from movies*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[68] Y. LECUN, Y. BENGIO, AND G. HINTON, *Deep learning*, Nature, 521 (2015), p. 436.

[69] H. LI, J. CHEN, Z. XU, H. CHEN, AND R. HU, *Multiple instance discriminative dictionary learning for action recognition*, in Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 2014–2018.

[70] J. LIU AND M. SHAH, *Learning human actions via information maximization*, in Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

[71] L. LIU, L. SHAO, X. ZHEN, AND X. LI, *Learning discriminative key poses for action recognition*, IEEE transactions on cybernetics, 43 (2013), pp. 1860–1870.

[72] D. G. LOWE, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision, 60 (2004), pp. 91–110.

[73] Y. LU, Y. LI, Y. SHEN, F. DING, X. WANG, J. HU, AND S. DING, *A human action recognition method based on tchebichef moment invariants and temporal templates*, in 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, vol. 2, IEEE, 2012, pp. 76–79.

[74] C.-Y. MA, M.-H. CHEN, Z. KIRA, AND G. ALREGIB, *Ts-lstm and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition*, Signal Processing: Image Communication, 71 (2019), pp. 76–87.

[75] A. MAKHZANI, J. SHLENS, N. JAITLY, I. GOODFELLOW, AND B. FREY, *Adversarial autoencoders*, arXiv preprint arXiv:1511.05644, (2015).

[76] M. MARSZAŁEK, I. LAPTEV, AND C. SCHMID, *Actions in context*, in CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition, IEEE Computer Society, 2009, pp. 2929–2936.

[77] M. MATHIEU, C. COUPRIE, AND Y. LECUN, *Deep multi-scale video prediction beyond mean square error*, International Conference of Learning Representations, (2016).

[78] H. MOBAHI, R. COLLOBERT, AND J. WESTON, *Deep learning from temporal coherence in video*, in Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 737–744.

[79] Z. D.-A. V. G. M. MOSTAFA S. IBRAHIM, SRIKANTH MURALIDHARAN, *A hierarchical deep temporal model for group activity recognition*, in CVPR, 2016.

[80] P. NATARAJAN AND R. NEVATIA, *Coupled hidden semi markov models for activity recognition*, in Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on, IEEE, 2007, pp. 10–10.

[81] S. NAZIR, M. H. YOUSAF, AND S. A. VELASTIN, *Evaluating a bag-of-visual features approach using spatio-temporal features for action recognition*, Computers & Electrical Engineering, 72 (2018), pp. 660–669.

[82] J. C. NIEBLES, C.-W. CHEN, AND L. FEI-FEI, *Modeling temporal structure of decomposable motion segments for activity classification*, in European conference on computer vision, Springer, 2010, pp. 392–405.

[83] J. C. NIEBLES AND L. FEI-FEI, *A hierarchical model of shape and appearance for human action classification*, in 2007 IEEE Conference on Computer Vision and Pattern Recognition, Citeseer, 2007, pp. 1–8.

[84] X.-X. NIU AND C. Y. SUEN, *A novel hybrid cnn–svm classifier for recognizing handwritten digits*, Pattern Recognition, 45 (2012), pp. 1318–1325.

[85] N. OLIVER, E. HORVITZ, AND A. GARG, *Layered representations for human activity recognition*, in Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on, IEEE, 2002, pp. 3–8.

[86] N. M. OLIVER, B. ROSARIO, AND A. P. PENTLAND, *A bayesian computer vision system for modeling human interactions*, IEEE transactions on pattern analysis and machine intelligence, 22 (2000), pp. 831–843.

[87] M. OU, P. CUI, J. PEI, Z. ZHANG, AND W. ZHU, *Asymmetric transitivity preserving graph embedding*, in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 1105–1114.

[88] P. PAN, Z. XU, Y. YANG, F. WU, AND Y. ZHUANG, *Hierarchical recurrent neural encoder for video representation with application to captioning*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1029–1038.

[89] S. PAN, R. HU, G. LONG, J. JIANG, L. YAO, AND C. ZHANG, *Adversarially regularized graph autoencoder for graph embedding.*, in IJCAI, 2018, pp. 2609–2615.

[90] S. PAN, J. WU, X. ZHU, C. ZHANG, AND Y. WANG, *Tri-party deep network representation*, Network, 11 (2016), p. 12.

[91] S. PAN, J. WU, X. ZHUY, C. ZHANG, AND P. S. YUZ, *Joint structure feature exploration and regularization for multi-task graph classification*, in Data Engineering (ICDE), 2016 IEEE 32nd International Conference on, IEEE, 2016, pp. 1474–1475.

[92] G. T. PAPADOPOULOS, A. AXENOPOULOS, AND P. DARAS, *Real-time skeleton-tracking-based human action recognition using kinect data*, in International Conference on Multimedia Modeling, Springer, 2014, pp. 473–483.

[93] S. N. PAUL AND Y. J. SINGH, *Survey on video analysis of human walking motion*, International Journal of Signal Processing, Image Processing and Pattern Recognition, 7 (2014), pp. 99–122.

[94] X. PENG AND C. SCHMID, *Multi-region two-stream r-cnn for action detection*, in European conference on computer vision, Springer, 2016, pp. 744–759.

[95] X. PENG, L. WANG, X. WANG, AND Y. QIAO, *Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice*, Computer Vision and Image Understanding, 150 (2016), pp. 109–125.

[96] B. PEROZZI, R. AL-RFOU, AND S. SKIENA, *Deepwalk: Online learning of social representations*, in Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 701–710.

[97] L. L. PRESTI AND M. LA CASCIA, *3d skeleton-based human action classification: A survey*, Pattern Recognition, 53 (2016), pp. 130–147.

[98] J. QIU, Y. DONG, H. MA, J. LI, K. WANG, AND J. TANG, *Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec*, in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, ACM, 2018, pp. 459–467.

[99] A. RADFORD, L. METZ, AND S. CHINTALA, *Unsupervised representation learning with deep convolutional generative adversarial networks*, arXiv preprint arXiv:1511.06434, (2015).

[100] H. RAGHEB, S. VELASTIN, P. REMAGNINO, AND T. ELLIS, *Human action recognition using robust power spectrum features*, in 2008 15th IEEE International Conference on Image Processing, IEEE, 2008, pp. 753–756.

[101] D. RAMANAN AND D. A. FORSYTH, *Automatic annotation of everyday movements*, in Advances in neural information processing systems, 2004, pp. 1547–1554.

[102] C. RAO AND M. SHAH, *View-invariance in action recognition*, in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, vol. 2, IEEE, 2001, pp. II–II.

[103] K. K. REDDY AND M. SHAH, *Recognizing 50 human action categories of web videos*, Machine Vision and Applications, 24 (2013), pp. 971–981.

[104] D. ROY, K. S. R. MURTY, AND C. K. MOHAN, *Action-vectors: Unsupervised movement modeling for action recognition*, in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE, 2017, pp. 1602–1606.

[105] A. SALEH, M. ABDEL-NASSER, M. A. GARCIA, AND D. PUIG, *Aggregating the temporal coherent descriptors in videos using multiple learning kernel for action recognition*, Pattern Recognition Letters, 105 (2018), pp. 4–12.

[106] M. SAQIB, S. D. KHAN, N. SHARMA, AND M. BLUMENSTEIN, *A study on detecting drones using deep convolutional neural networks*, in Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on, IEEE, 2017, pp. 1–5.

[107] C. SCHULDT, I. LAPTEV, AND B. CAPUTO, *Recognizing human actions: a local svm approach*, in Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3, IEEE, 2004, pp. 32–36.

[108] P. SCOVANNER, S. ALI, AND M. SHAH, *A 3-dimensional sift descriptor and its application to action recognition*, in Proceedings of the 15th ACM international conference on Multimedia, ACM, 2007, pp. 357–360.

[109] Y. SHEIKH, M. SHEIKH, AND M. SHAH, *Exploring the space of a human action*, in Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 1, IEEE, 2005, pp. 144–149.

[110] Y. SHI, W. ZENG, T. HUANG, AND Y. WANG, *Learning deep trajectory descriptor for action recognition in videos using deep neural networks*, in 2015 IEEE International Conference on Multimedia and Expo (ICME), IEEE, 2015, pp. 1–6.

[111] N. SHU, Q. TANG, AND H. LIU, *A bio-inspired approach modeling spiking neural networks of visual cortex for human action recognition*, in 2014 International Joint Conference on Neural Networks (IJCNN), IEEE, 2014, pp. 3450–3457.

[112] K. SIMONYAN AND A. ZISSERMAN, *Two-stream convolutional networks for action recognition in videos*, in Advances in neural information processing systems, 2014, pp. 568–576.

[113] S. SINGH, S. A. VELASTIN, AND H. RAGHEB, *Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods*, in 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, IEEE, 2010, pp. 48–55.

[114] K. SOOMRO AND A. R. ZAMIR, *Action recognition in realistic sports videos*, in Computer vision in sports, Springer, 2014, pp. 181–208.

[115] K. SOOMRO, A. R. ZAMIR, AND M. SHAH, *Ucf101: A dataset of 101 human actions classes from videos in the wild*, arXiv preprint arXiv:1212.0402, (2012).

[116] N. SRIVASTAVA, E. MANSIMOV, AND R. SALAKHUDINOV, *Unsupervised learning of video representations using lstms*, in International conference on machine learning, 2015, pp. 843–852.

[117] L. SUN, K. JIA, D.-Y. YEUNG, AND B. E. SHI, *Human action recognition using factorized spatio-temporal convolutional networks*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4597–4605.

[118] J. TANG, M. QU, M. WANG, M. ZHANG, J. YAN, AND Q. MEI, *Line: Large-scale information network embedding*, in Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 1067–1077.

[119] L. TANG AND H. LIU, *Leveraging social media networks for classification*, Data Mining and Knowledge Discovery, 23 (2011), pp. 447–478.

[120] F. TIAN, B. GAO, Q. CUI, E. CHEN, AND T.-Y. LIU, *Learning deep representations for graph clustering.*, in AAAI, 2014, pp. 1293–1299.

[121] D. TRAN, L. BOURDEV, R. FERGUS, L. TORRESANI, AND M. PALURI, *Learning spatiotemporal features with 3d convolutional networks*, in Proceedings of the IEEE international conference on computer vision, 2015, pp. 4489–4497.

[122] J. C. VAN GEMERT, M. JAIN, E. GATI, C. G. SNOEK, ET AL., *Apt: Action localization proposals from dense trajectories.*, in BMVC, vol. 2, 2015, p. 4.

[123] G. VAROL, I. LAPTEV, AND C. SCHMID, *Long-term temporal convolutions for action recognition*, IEEE transactions on pattern analysis and machine intelligence, 40 (2018), pp. 1510–1517.

[124] V. VEERIAH, N. ZHUANG, AND G.-J. QI, *Differential recurrent neural networks for action recognition*, in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4041–4049.

[125] S. A.-E.-H. A. G. K. M. VIGNESH RAMANATHAN, JONATHAN HUANG AND L. FEI-FEI, *Detecting events and key actors in multi-person videos*, in CVPR, 2016.

[126] C. WANG, S. PAN, G. LONG, X. ZHU, AND J. JIANG, *Mgae: Marginalized graph autoencoder for graph clustering*, in Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, ACM, 2017, pp. 889–898.

[127] D. WANG, P. CUI, AND W. ZHU, *Structural deep network embedding*, in Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2016, pp. 1225–1234.

[128] H. WANG, A. KLÄSER, C. SCHMID, AND C.-L. LIU, *Action recognition by dense trajectories*, in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 3169–3176.

[129] H. WANG, D. ONEATA, J. VERBEEK, AND C. SCHMID, *A robust and efficient video representation for action recognition*, International Journal of Computer Vision, 119 (2016), pp. 219–238.

[130] H. WANG AND C. SCHMID, *Action recognition with improved trajectories*, in Proceedings of the IEEE international conference on computer vision, 2013, pp. 3551–3558.

[131] H. WANG, M. M. ULLAH, A. KLASER, I. LAPTEV, AND C. SCHMID, *Evaluation of local spatio-temporal features for action recognition*, in BMVC 2009-British Machine Vision Conference, BMVA Press, 2009, pp. 124–1.

[132] L. WANG, Y. QIAO, AND X. TANG, *Action recognition with trajectory-pooled deep-convolutional descriptors*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4305–4314.

[133] ——, *Mofap: A multi-level representation for action recognition*, International Journal of Computer Vision, 119 (2016), pp. 254–271.

[134] L. WANG, Y. XIONG, Z. WANG, Y. QIAO, D. LIN, X. TANG, AND L. VAN GOOL, *Temporal segment networks: Towards good practices for deep action recognition*, in European Conference on Computer Vision, Springer, 2016, pp. 20–36.

[135] P. WANG, Y. CAO, C. SHEN, L. LIU, AND H. T. SHEN, *Temporal pyramid pooling-based convolutional neural network for action recognition*, IEEE Transactions on Circuits and Systems for Video Technology, 27 (2017), pp. 2613–2622.

[136] X. WANG, P. CUI, J. WANG, J. PEI, W. ZHU, AND S. YANG, *Community preserving network embedding.*, in AAAI, 2017, pp. 203–209.

[137] X. WANG, A. FARHADI, AND A. GUPTA, *Actions˜ transformations*, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2016, pp. 2658–2667.

[138] Y. WANG, K. HUANG, AND T. TAN, *Human activity recognition based on r transform*, in 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.

[139] Y. WANG, J. SONG, L. WANG, L. VAN GOOL, AND O. HILLIGES, *Two-stream sr-cnns for action recognition in videos.*, in BMVC, 2016.

[140] Z. WANG, C. CHEN, AND W. LI, *Predictive network representation learning for link prediction*, in Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2017, pp. 969–972.

[141] D. WEINLAND, R. RONFARD, AND E. BOYER, *Free viewpoint action recognition using motion history volumes*, Computer vision and image understanding, 104 (2006), pp. 249–257.

[142] G. WILLEMS, T. TUYTELAARS, AND L. VAN GOOL, *An efficient dense and scale-invariant spatio-temporal interest point detector*, in European conference on computer vision, Springer, 2008, pp. 650–663.

[143] I. H. WITTEN, E. FRANK, M. A. HALL, AND C. J. PAL, *Data Mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2016.

[144] D. WU, J. CHEN, N. SHARMA, S. PAN, G. LONG, AND M. BLUMENSTEIN, *Adversarial action data augmentation for similar gesture action recognition*, in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[145] D. WU, R. HU, Y. ZHENG, J. JIANG, N. SHARMA, AND M. BLUMENSTEIN, *Feature-dependent graph convolutional autoencoders with adversarial training methods*,

in 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.

[146] D. WU, N. SHARMA, AND M. BLUMENSTEIN, *Recent advances in video-based human action recognition using deep learning: a review*, in 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 2865–2872.

[147] ——, *An end-to-end hierarchical classification approach for similar gesture recognition*, in 2018 International Conference on Image and Vision Computing New Zealand (IVCNZ), IEEE, 2018, pp. 1–6.

[148] ——, *Similar gesture recognition using hierarchical classification approach in rgb videos*, in 2018 Digital Image Computing: Techniques and Applications (DICTA), IEEE, 2018, pp. 1–7.

[149] R. XIA, Y. PAN, L. DU, AND J. YIN, *Robust multi-view spectral clustering via low-rank and sparse decomposition.*, in AAAI, 2014, pp. 2149–2155.

[150] K. XU, X. JIANG, AND T. SUN, *Two-stream dictionary learning architecture for action recognition*, IEEE Transactions on Circuits and Systems for Video Technology, 27 (2017), pp. 567–576.

[151] J. YAMATO, J. OHYA, AND K. ISHII, *Recognizing human action in time-sequential images using hidden markov model*, in Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on, IEEE, 1992, pp. 379–385.

[152] S. YAN, J. S. SMITH, W. LU, AND B. ZHANG, *Hierarchical multi-scale attention networks for action recognition*, Signal Processing: Image Communication, 61 (2018), pp. 73–84.

[153] C. YANG, Z. LIU, D. ZHAO, M. SUN, AND E. Y. CHANG, *Network representation learning with rich text information.*, in IJCAI, 2015, pp. 2111–2117.

[154] Y. YANG, I. SALEEMI, AND M. SHAH, *Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions*, IEEE transactions on pattern analysis and machine intelligence, 35 (2013), pp. 1635–1648.

[155] A. YILMAZ AND M. SHAH, *Actions as objects: A novel action representation*, CVPR, 2005.

[156] J. YUE-HEI NG, M. HAUSKNECHT, S. VIJAYANARASIMHAN, O. VINYALS, R. MONGA, AND G. TODERICI, *Beyond short snippets: Deep networks for video classification*, in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 4694–4702.

[157] D. ZHANG, J. YIN, X. ZHU, AND C. ZHANG, *User profile preserving social network embedding*, in Proceedings of IJCAI, 2017, pp. 3378–3384.

[158] ——, *Network representation learning: a survey*, IEEE Transactions on Big Data, (2018).

[159] L. ZHANG, Y. FENG, X. XIANG, AND X. ZHEN, *Realistic human action recognition: When cnns meet lds*, in Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on, IEEE, 2017, pp. 1622–1626.

[160] H. ZHU, R. VIAL, AND S. LU, *Tornado: A spatio-temporal convolutional regression network for video action proposal*, in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5813–5821.

[161] J. ZHU, Z. ZHU, AND W. ZOU, *End-to-end video-level representation learning for action recognition*, in 2018 24th International Conference on Pattern Recognition (ICPR), IEEE, 2018, pp. 645–650.

[162] W. ZHU, J. HU, G. SUN, X. CAO, AND Y. QIAO, *A key volume mining deep framework for action recognition*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1991–1999.

[163] M. ZIAEEFARD AND R. BERGEVIN, *Semantic human activity recognition: A literature review*, Pattern Recognition, 48 (2015), pp. 2329–2345.