Deep Representation Learning for Action Recognition

A dissertation presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy
in
Computer Science

at Massey University, Auckland, New Zealand

Jun Ren

2019

To Shuangshuang Lu
for her love and support

Abstract

This research focuses on deep representation learning for human action recognition based on the emerging deep learning techniques using RGB and skeleton data. The output of such deep learning techniques is a parameterised hierarchical model, representing the learnt knowledge from the training dataset. It is similar to the knowledge stored in our brain, which is learned from our experience. Currently, the computer's ability to perform such abstraction is far behind human's level, perhaps due to the complex processing of spatio-temporal knowledge.

The discriminative spatio-temporal representation of human actions is the key for human action recognition systems. Different feature encoding approaches and different learning models may lead to quite different output performances, and at the present time there is no approach that can accurately model the cognitive processing for human actions. This thesis presents several novel approaches to allow computers to learn discriminative, compact and representative spatio-temporal features for human action recognition from multiple input features, aiming at enhancing the performance of an automated system for human action recognition.

The input features for the proposed approaches in this thesis are derived from signals that are captured by the depth camera, e.g., RGB video and skeleton data. In this thesis, I developed several geometric features, and proposed the following models for action recognition: CVR-CNN, SKB-TCN, Multi-Stream CNN and STN. These proposed models are inspired by the visual attention mechanisms that are inherently present in human beings. In addition, I discussed the performance of the geometric features that I developed along with the proposed models.

Superior experimental results for the proposed geometric features and models are obtained and verified on several benchmarking human action recognition datasets. In the case of the most challenging benchmarking dataset, NTU RGB+D, the accuracy of the results obtained surpassed the performance of the existing RNN-based and ST-GCN models. This study provides a deeper understanding of the spatio-temporal representation of human actions and it has significant implications to explain the inner workings of the deep learning models in learning patterns from time series data. The findings of these proposed models can set forth a solid foundation for further developments, and for the guidance of future human action-related studies.

Acknowledgements

From my perspective undertaking this PhD study has truly been a life-changing journey for me. It would never have been possible without the strong support and supervision that I received from many people. First of all, I would like to take this opportunity to express my deep gratitude to my supervisor, Dr. Reyes Napoleon, for his strong support and invaluable guidance, which ensured my research stayed on the right track. In our fruitful discussions he has always contributed new insights while also providing me with the freedom to pursue my own ideas. With his deep and broad knowledge of science, Dr. Napoleon has taught me a lot about academic research.

I would also like to extend my thanks to my co-supervisors Dr. Andre Barczak, A/Prof. Chris Scogings and Prof. Mingzhe Liu for advising me during my stays in the robotics lab at Massey University and during our regular meetings. I feel very fortunate to have such an excellent academic environment to finish my PhD study. I greatly appreciate the long-term support and help from Prof. Mingzhe Liu during my Master and PhD study.

Also, I would like to thank all my friends and colleagues for making my time as a student very enjoyable and memorable in New Zealand. Special thanks go to Julia Ma, Rahila Umer, Tiffany Tang and Yukio Fukuzawa for their help during my PhD study. Finally, my special appreciation is dedicated to my wonderful wife, Shuangshuang Lu, and to my parents for their unconditional love and endless support that has made all this possible.

Contents

Abstract iii

Acknowledgements v

Contents vi

List of Publications xi

List of Figures xii

List of Tables xvi

List of Acronyms xviii

1 Introduction 1

1.1 Human action recognition 1

1.2 Problem statement 2

1.3 Motivation 3

1.4 Contributions 5

1.5 Outline of the dissertation 7

2 Literature Review for Human Action Recognition **9**

2.1 An overview of action recognition 9

2.2 Traditional approaches for action recognition 14

2.2.1 RGB Video-based action recognition 14

2.2.2 Depth video-based action recognition 17

2.2.3 RGB+D video-based action recognition 20

2.3 Deep learning-based action recognition 21

2.4 Summary 26

3 Skeleton-to-Image based Human Action Recognition **28**

3.1 Motivation 28

3.2 Related work 29

3.2.1 Skeleton-based primitive geometric features 30

3.2.2 Regularized CNN models 30

3.3 The algorithms 31

3.3.1 General architecture 31

3.3.2 Novel input representation of actions 34

3.3.3 Encoding primitive geometric features 35

3.3.4 Correctness-Vigilant Regularized CNN Model 41

3.4 Empirical testing and analysis 45

3.4.1 Datasets 45

3.4.2 Implementation details 46

3.4.3 Results and Comparisons 46

3.5 Summary and contributions 54

4 Spatio-temporal Kernel based Temporal Convolutional Network for Human Action Recognition **55**

4.1 Motivation . . . 55

4.2 Related work . . . 57

4.2.1 Geometric relational features based on 3D skeleton . . . 57

4.2.2 Local signal encoding strategies for action recognition . . . 59

4.2.3 Skeleton-based action recognition . . . 61

4.3 The algorithms . . . 62

4.3.1 Extension of geometric relational features . . . 62

4.3.2 General architecture . . . 66

4.3.3 Pipeline for action recognition . . . 69

4.3.4 Local temporal contextual feature extraction . . . 71

4.3.5 Variation of the proposed model . . . 73

4.4 Empirical testing and analysis . . . 73

4.4.1 Datasets . . . 73

4.4.2 Implementation Details . . . 74

4.4.3 Results and comparisons . . . 75

4.4.4 Results of UT-Kinect dataset . . . 76

4.4.5 Results of SBU-Kinect Interaction dataset . . . 78

4.4.6 Results of UTD-MHAD dataset . . . 79

4.4.7 Ablation Study . . . 81

4.5 Summary and contributions . . . 82

5 Action Recognition with CNN Features using LSTM-C RNN Model **84**

5.1 Motivation . . . 84

5.2 Related work . . . 86

5.3 The algorithms . . . 87

5.3.1 General architecture . . . 87

5.3.2 Workflow of the CNN-LSTM-C framework . . . 89

5.4 Empirical testing and analysis . . . 94

5.4.1 Datasets . . . 94

5.4.2 Implementation Details . . . 95

5.4.3 Results and comparisons . . . 95

5.5 Summary and contributions . . . 100

6 Multi-stream CNN model for Human Action Recognition **102**

6.1 Motivation . . . 102

6.2 Related work . . . 103

6.3 The algorithm . . . 104

6.3.1 General architecture . . . 104

6.3.2 Multi-stream CNN model . . . 106

6.3.3 Interaction actions . . . 107

6.4 Empirical testing and analysis . . . 108

6.4.1 Dataset . . . 108

6.4.2 Results of SBU-Kinect Interaction database . . . 109

6.4.3 Results of NTU RGB+D database . . . 112

6.4.4 Results of UTD MHAD database 112

6.5 Summary and contributions 114

7 Human Action Recognition based on Skeleton Transformer Network 116

7.1 Motivation 116

7.2 Related work 117

7.3 The algorithm 119

7.3.1 General architecture 119

7.3.2 Skeleton transformer network 121

7.3.3 Training optimization 122

7.4 Empirical testing and analysis 123

7.4.1 Dataset 123

7.4.2 Results of UTD MHAD database 124

7.4.3 Results of Northwestern UCLA database 125

7.4.4 Results of NTU RGB+D database 126

7.4.5 Visualization 129

7.5 Summary and contributions 130

8 Conclusion and perspectives 132

8.1 Key contributions 132

8.2 Future work 134

Bibliography 136

List of Publications

The work presented in this thesis is related with following publications.

1). Ren, J., Reyes, N., Barczak, A., Scogings, C. & Liu, M. (2018). Toward three-dimensional human action recognition using a convolutional neural network with correctness-vigilant regularizer. Journal of Electronic Imaging, 27(4), 043040.

2). J. Ren, R. Napoleon, B. Andre, S. Chris, M. Liu & J. Ma. (2018). Robust Skeleton-based Action Recognition through Hierarchical Aggregation of Local and Global Spatio-temporal Features. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 901-906). IEEE.

3). J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings & M. Liu.(2018). An Investigation of Skeleton-Based Optical Flow-Guided Features for 3D Action Recognition Using a Multi-Stream CNN Model. In 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC) (pp. 199-203). IEEE.

4). J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings & M. Liu.(2018). Towards 3D Human Action Recognition Using a Distilled CNN Model. In 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP) (pp. 7-12). IEEE.

List of Figures

2.1 List of available databases for Human action recognition (HAR) 12

2.2 MEI and MHI based approach [1] 15

2.3 Detection of spatio-temporal interest points [4], [59] 15

2.4 The workflow of feature extraction process of Dense Trajectory [33] 16

2.5 The workflow of Histogram extraction from Oriented 4D Normals [64] 17

2.6 Example of BRSF [68] 18

2.7 Pose-Based action recognition model [69] 19

2.8 Framework of ST-GCN model [77] 20

2.9 Transfer learning for the RGB Dataset (a) and BHIM model (b) [80], [81] . . . 21

2.10 The architecture for 3D-CNN model [85] 22

2.11 The framework for LSTM-RNN model [86] 23

2.12 Stacked convolutional ISA network [89] 23

2.13 The framework for two-stream model [88] 24

2.14 Current most popular video analysis framework [99] 25

2.15 Unified framework for video analysis [97] 25

2.16 Unsupervised video representation learning model [105] 26

3.1 The flowchart of the proposed CVR-CNN framework 32

3.2 Architecture of CVR-CNN Model 32

3.3 Example of typical layout of skeleton data (a), motion features (b) and body-part representation strategy 34

3.4 The process of conversion from skeleton data to temporal kinematic image representation 35

3.5 Variation of kinetic energy of joint "ankle" for two actions: "kick" and "run slowly" 39

3.6 Image representation of ten different PGFs for four actions in NTU RGB+D. The code for calculating the above 10 features can be accessed in this link: https://gitlab.com/jren2019/skb_cvr_cnn/blob/master/calSOF.m 39

3.7 Rotation and interpolation of features: PGF2 (left figure) and PGF3 (right figure) 40

3.8 Regularized cross-entropy 44

3.9 Convergence rate curves of training for different input features on UTD-MHAD 48

3.10 Confusion matrix for feature "PGF4+PGF-M" on UTD-MHAD 49

3.11 Confusion matrix for feature "PGF4+PGF-M" on Northwestern UCLA [128] . . 49

3.12 Confusion matrix for three different features on UTD-MHAD (Left: without correctness-vigilant regularizer, Right: with correctness-vigilant regularizer) 50

3.13 Convergence rate curve of training for different features on Northwestern UCLA 52

3.14 Convergence rate curves of training for best results on NTU RGB+D 53

3.15 Confusion matrix for best result on NTU RGB+D 53

4.1 Actions instances with similar variation of skeleton sequences 59

4.2 Extended geometric relational features. a) motion features derived from joint coordinates; b) Distance between two joints; c) Angle between adjacent limbs; d) Angle between joint-joint-lines; e) Angle between joint-joint-line and plane; f) Angles between plane and plane; g) Distance between two joints; h) Distance between joint and joint-joint-line; i) Distance between joint and plane . . . 63

4.3 The workflow of the proposed SKB-TCN framework 67

4.4 Information flow of one LSTM node 70

4.5 The architecture of spatio-temporal kernel. 72

4.6 Convergence rate curve for different geometric relational features for UT-Kinect 77

4.7 Convergence curve for best three combinations on UTD-MHAD 80

4.8 Convergence rate curve for the proposed model with(left) and without(right) spatio-temporal kernel 81

4.9 Performance of variant model with spatio-temporal kernel 82

5.1 The workflow of the proposed CNN-LSTM-C framework 88

5.2 Proposed framework for video-based action recognition 90

5.3 The architecture of the LSTM-C model 92

5.4 Samples of experimental action recognition datasets 94

5.5 Performance of LSTM-C on UCF 11 dataset 96

5.6 Convergence curves for LSTM and LSTM-C with $memory\_size = 30$ on UCF 11 97

5.7 Convergence curves for RGB video-based action recognition for UCF-Sports . 97

5.8 An action instance of UTD MHAD dataset: a) RGB image, b) original skeleton data (provided by the database, (x, y, z) for 20 joints), c) estimated noisy 3d skeleton data, d) estimated 2D skeleton data 98

5.9 Accuracy of RGB video-based action recognition for UTD-MHAD dataset . . 100

6.1 Workflow of the proposed Multi-stream CNN model 104

6.2 Overview of the proposed multi-stream CNN model for fusing geometric and kinematic features. The input streams are combinations of the aforementioned features, such as "joint coordinates", "motion features" and "energy features". 105

6.3 Variants of proposed multi-stream CNN model for fusing geometric features 107

6.4 Visualization for two actions with swapping the positions of two subjects . . 109

6.5 Confusion matrix of five-fold testing with multi-stream CNN model 111

6.6 Convergence curve of Multi-stream CNN model on NTU RGB+D 111

6.7 Convergence rate curves for different fusion strategies on UTD-MHAD (SOF2+SOF3) 115

7.1 Framework of the STN model 119

7.2 Variant of STN model to accommodate actions performed by multiple actors. The transformed coordinates are hypothetical examples, a visualisation of the actual results are provided in Fig.7.9 120

7.3 Confusion matrix of Skeleton Transformer Network (STN) on UTD-MHAD . 124

7.4 Convergence curve of STN model on UTD-MHAD 124

7.5 Confusion matrix of STN model on Northwestern UCLA 125

7.6 Training and testing accuracy and loss for STN on Northwestern UCLA . . . 126

7.7 Confusion matrix of best result with Cross-View evaluation strategy 128

7.8 Convergence rate curve for training and testing on NTU RGB+D 129

7.9 Image visualization of raw skeleton sequence and the transformed sequences. The number alongside each frame indicates the action type. 130

List of Tables

3.1 Hyperparameters for the Correctness-Vigilant Regularizer Convolutional Neural Networks (CVR-CNN) model 34

3.2 Performance comparison for CVR-CNN model 47

4.1 Representation of proposed geometric relational features 66

4.2 Hyperparameters tuned in our proposed model 69

4.3 Performance comparison of SKB-TCN model and related models on UT-Kinect, SBU-Kinect and UTD-MHAD dataset. 75

4.4 Performance of different combination with the derived motion and energy features on UT-Kinect dataset 78

4.5 Recognition accuracy based on the derived features with different configurations 78

4.6 Performance of different combination with the derived motion and energy features on SBU-Kinect. 79

4.7 Performance of different combination with the derived motion and energy features on UTD-MHAD. 80

5.1 Hyperparameters settings in our proposed model 88

5.2 Comparison of LSTM-C and related models on UCF11 and UCF-Sports. . . . 96

5.3 Performance comparison of video-based approach on UTD-MHAD dataset . 99

6.1 Performance comparison of Multi-CNN and related models on SBU-Kinect . 110

6.2 Performance comparison of Multi-CNN and related models on NTU RGB+D. 112

6.3 Performance comparison of Multi-CNN and related models on UTD-MHAD. 113

6.4 Results of different feature combinations on UTD-MHAD 114

7.1 Performance comparison of STN and related models on UTD-MHAD. 123

7.2 Performance comparison of STN and related models on Northwestern-UCLA 125

7.3 Performance comparison of STN and related models on NTU RGB+D 127

List of Acronyms

BHIM Bilinear Heterogeneous Information Machine

BoF Bag of Features

BoP Bag-of-Points

BoW Bag of Words

BRSF Binary Range-Sample Feature

CNN Convolutional Neural Networks

CVR-CNN Correctness-Vigilant Regularizer Convolutional Neural Networks

DBM Deep Boltzman Machine

DBN Deep Belief Network

DMW Dynamic Manifold Warping

DNN Deep Neural Networks

DPM Deformable Part-based Model

DTW Dynamic Time Warping

FC Fully Connected

FTP Fourier Temporal Pyramid

GRU Gated Recurrent Unit

HAR Human action recognition

HAU Human Action Understanding

HMM Hidden Markov Model

HOF Histogram of Optical Flow

HOG Histogram of oriented gradients

HOJ3D Histograms of 3D Joints

HON4D Histogram of Oriented 4D Normals

HRNN Hierarchical Recurrent Neural Networks

ISA Independent Subspace Analysis

LDA Latent Dirichlet Allocation

LDS Linear Dynamical Systems

LOP Local Occupancy Patterns

LRCN Long-term Recurrent Convolutional Network

LSTM Long-short Term Memory

LSTM-C Long-short Term Memory with Constituent nodes

LSTM-RNN Long-short Term Memory Recurrent Neural Network

MBH Motion Boundary Histogram

MEI Motion Energy Images

MHI Motion History Images

MKL Multiple Kernel Learning

PGF Primitive Geometric Features

RNN Recurrent Neural Networks

SFA Slow Feature Analysis

SKB-TCN Spatio-temporal Kernel Based Temporal Convolutional Network

SOFs Skeleton based Optical-flow guided Features

SSVM Structural Support Vector Machine

STN Skeleton Transformer Network

STP Spatial-Temporal Pyramids

SVM Support Vector Machine

TSN Temporal Segment Networks

VLAD Vector of locally Aggregated Descriptor

WFMMNN Weighted Fuzzy Min-Max Neural Network

Chapter 1

Introduction

1.1 Human action recognition

In the computer vision research community, Human Action Recognition (HAR), which is the key for Human Action Understanding (HAU), has been one of the most important research lines because of its wide spectrum of applications, e.g., patient monitoring, smart surveillance and sport video analysis and so forth. The ultimate objective of HAR is to determine the label of the action for a person or a group from a stream of videos and its context information. As reported in [1], currently, there is no one general theoretical framework that is available that can be employed to ideally model the evolution of human actions. HAR is associated with a wide variety of challenges, such as scaling, occlusion and clutter in the spatial domain. The extra overhead posed by the changing illuminations, dynamic situations, cluttered background, and so forth turn HAR into a more complicated task. Moreover, HAR is also complicated by the variations between different actors [2], because different actors usually present quite different appearance for the same action.

In the past years, HAR has been extensively investigated, even though there are still challenges for realistic applications. The earlier works related to HAR are based on videos and images, due to the easily accessible videos and images, so various algorithms were proposed for videos in different scenarios. With the advancement in depth camera technology, such as the Kinect from Microsoft, it became possible to acquire depth information from the environment directly, and this has resulted in depth images. Consequently, human action recognition based on computer vision can be divided into RGB video-based approach and depth video-based approach. Moreover, with the more recent advancement of pose estimation, skeleton-based action recognition has become another alternative for this task.

Therefore, the depth video-based action recognition includes two efficient research lines, namely, depth image-based action recognition and skeleton-based action recognition. For the former approach, most research adopted similar approaches that were designed for RGB video-based action recognition. More recently, the latter approach has become one of the most efficient methods for action recognition, because of the accurate and compact representational ability of the skeleton data, which is promising in realistic applications, e.g., robotics vision and game control. In summary, the rich spatial structure and the dynamics of the RGB video and depth video are used to distinguish different actions. In terms of image-based action recognition, as another research line of action recognition, it usually relies on image segmentation techniques or object detection techniques. This approach is effective for actions that are not sensitive to motion dynamics, and the body pose is the key clue for classifying the actions correctly from the static images. Although various successful models have been developed for HAR, there is still room for improvement, particularly for realistic applications. Therefore, more efficient alternative models should be developed for robust action recognition. Human beings are capable of learning and recognizing different actions, due to the brain's powerful signal processing ability, particularly the attention mechanisms. However, in computer vision, mimicking the function of the human brain is one of the most challenging tasks.

1.2 Problem statement

This thesis aims to identify efficient representation learning approaches for HAR from multiple features that are easily accessible in our daily life, including RGB video and human skeleton data. The question then becomes how to encode the skeleton sequence efficiently for human action recognition? Other questions include the following: are there any more discriminative features that are available to distinguish human actions in complex scenes, and also, how to extract and fuse multiple features to increase the recognition accuracy of HAR systems? The visual features acquired from the environment include redundant information, how to refine the input features and extract significant representative information about the actions so as to augment the discriminative ability of the action recognition systems? Is there any advanced approach to suppress the interference of the

irrelevant noise to train a more robust model for skeleton-based action recognition systems? Although various effective spatial and temporal hand-crafted features demonstrate superior performance on existing datasets, another open question that still exists is how to utilize the deep learning techniques to augment the training dataset. This research proposes several approaches to address the questions mentioned above.

1.3 Motivation

The task of HAR research mainly includes the following five parts: feature extraction from video or skeleton sequence, feature selection, feature encoding, feature fusion and classifier. These five components are deeply investigated in the domain of computer vision research, and among these steps efficient feature encoding (feature representation) is considered as the most important step for subsequent operations. Conventional approaches to encode hand-crafted features for creating spatio-temporal representation are based on Spatial-Temporal Pyramids (STP) methods [3]. In traditional RGB video-based methods, the predefined feature detector and descriptor are utilized to extract and represent interest regions in video frames. In terms of the classical feature detector, it includes the Harris detector [4], the Cuboid detector [5] and the Hessian detector [6]. Regarding the classical feature descriptor, it includes the Cuboid descriptor [5], the HOG3D descriptor [7], the HOGHOF descriptor [8] and the ESURF descriptor [9]. Even though these feature detectors and descriptors are commonly used in image and video processing; they are not specifically designed for HAR. These methods convert spatio-temporal variations into static features and then use a spatio-temporal volume to represent the actions contained in one period of the video, which unavoidably lose some information. Due to the limited representation ability of the hand-crafted features, it is very challenging to further improve the performance. Apart from the limitation of the recognition accuracy, because traditional video-based methods require more computational cost for feature extraction, they are not suitable for large-scale realistic action recognition systems.

As an emerging topic in the area of academia as well as in industry, deep learning is a promising approach for some challenging artificial intelligence problems. However, up to now, there are still no an official definitions for deep learning, which can be generally treated

as an approach to compose the representations of the data in a hierarchical manner. The Error Backpropagation Algorithm is the core idea for the training of neural network models, such as Multilayer Perceptron, Deep Belief Network (DBN), Deep Boltzman Machine (DBM), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN), and so forth [10]. According to the most recent literature, the CNNs and the RNNs are the two most widely used neural networks among the numerous deep learning models. The CNN model is the product of combining the theory of signal process and neural perception. It uses the convolution operation and the pooling operation to reduce the training parameters for large-scale neural network models. The convolution operation can be treated as a series of filters in the signal processing area. While the RNN model can be treated as a kind of time-series filters, since it is good at processing time series problems. Deep learning models are often referred to these two models or their combinations of them.

The first prototype of the current popular CNN model was proposed by Fukushima in 1980. It was used to detect patterns from the input data [11]. Due to the computational limitation, the convolutional neural network could not be applied widely at that time. The rapid rise of advanced neural networks is due to the significant improvement of AlexNet in 2012, on the ImageNet competition by Alex Krizhevsky et al. After that, the CNN model, as a biologically-inspired model, has extensively been used in solving computer vision problems, and then it achieved a breakthrough in many computer vision problems, including object localization, object detection, object segmentation, large scale video classification and human action recognition [12]–[15].

The application of RNNs was limited in earlier times because of its training problems, such as gradient vanishing and gradient explosion. Jurgen Schmidhuber and his student Sepp Hochreiter proposed the Long-short Term Memory (LSTM) model, which uses three gates to control the information transmission in the RNNs, to provide an efficient solution for the gradient vanishing and explosion problem. After that, Jurgen Schmidhuber and his student Alex Graves extended the LSTM model into handwriting digit recognition and speech recognition successfully [16], [17], which advanced the research of RNN significantly. The training of RNNs relies on the Back Propagation Through Time (BPTT) [18], which updates the parameter iteratively by unfolding the RNN along the time axis.

The data acquired in real-world applications for action recognition often consists of spatial

structure information and temporal dynamics. The former reflects the state of the object at one specific time step, while the latter reflects the evolution of the spatial structures. Both the CNNs and the RNNs model can be applied to process the static spatial structure information and the time-series signals. For example, static images mainly contain the static spatial information, so the CNNs model is widely used to solve images classification problems, while we can also convert the spatial information in a static image into sequential information, and then use the RNNs model to solve the image-related problems [19].

Motivated by the excellent representation learning ability of the deep learning techniques, especially the CNN model and the RNN model, this thesis devised several models for learning efficient representation from multiple features that are available for action recognition. The efficient extraction of the feature representation will be useful for a wide spectrum of applications. In this research study, we verified our proposed models based on several challenging and publicly available datasets that are designed for human action recognition.

1.4 Contributions

The aim of this thesis is to recognize actions from multiple input features with the latest emerging deep learning techniques. The first two parts of our work investigated the hand-crafted features that are derived from the skeleton sequence with our proposed CNN model and RNN model extensively. The third part of this work introduces a new attention mechanism for the Long-short Term Memory Recurrent Neural Network (LSTM-RNN) and utilizes this proposed model to recognize actions from videos with the CNN-based feature as input. We investigated the effect of different memory size for our Long-short Term Memory with Constituent nodes (LSTM-C) model for HAR. The fourth part of this thesis attempts to combine multiple geometric features for HAR with a novel multi-stream CNN model. Then, we attempt to use the STN model to transform the input sequence so as to augment the training dataset in the last part, which will in turn improve the performance on the test dataset. The proposed model demonstrates superior performance compared to the related models reported in the latest literature, and the performance of the STN model on the largest database called NTU RGB+D [20] surpasses the performance of the existing methods by 1.66% (CV) and 1.42%(CS). To sum up, the following contributions can be identified from this thesis:

- Motivated by the video-based optical flow, the Skeleton based Optical-flow guided Features (SOFs) are introduced in this thesis and several discriminative geometric and kinematic features are proposed. For this contribution, traditional skeleton-based features are investigated and novel motion and energy features are developed. We verified the proposed features with two most popular deep learning models, the CNNs and the RNNs. Based on the baseline system, we optimized these two models, respectively, by proposing the correctness-vigilant regularizer for the convolutional neural network and developing spatio-temporal kernel based temporal convolutional neural network. With the proposed correctness-vigilant regularizer, we can speed up the training process and output a more robust model, achieving a better recognition accuracy on the testing dataset.
- A spatio-temporal kernel-based temporal convolutional network is utilized to address the limitation of the CNN-based models, which involves a conversion of the skeleton to the static image that unavoidably loses some temporal information. With the proposed spatio-temporal kernel, we can aggregate sequential features locally and globally in a hierarchical way.
- A novel attention mechanism for video-based action recognition is proposed, which can integrate the spatial feature from the static images and the temporal dynamics between the image frames together to formulate the final optimized representation for the entire video. Image-based CNN-features are used to represent the spatial features, which are fed into our proposed LSTM-C model, and then the LSTM-C model is utilized to selectively extract the key frames from the input sequence to formulate the final representation. Experimental results of LSTM-C model on several video benchmarking datasets indicate that the LSTM-C model can efficiently extract the key features from the input visual features, which make the output model outperform the baseline systems.
- A multi-stream CNN framework is proposed for fusing the skeleton based geometric relational features. Multiple independent CNN models are employed to extract discriminative features from different features separately, and then a fusion operation is utilised to fuse the extracted features. The multi-CNN model can obtain the state-of-the-art result on the UTD-MHAD dataset in terms of recognition accuracy.

- Finally, a STN model is proposed to filter and at the same time enhance the sequential input features in an end-to-end manner to improve the performance of the output model on the testing dataset. Our results in Fig.7.9 provide some visualization of the original input and transformed features. With the transformed sequence as the input of the classifier, the output results on several benchmarking datasets demonstrated the superiority of the STN model. The proposed framework is flexible to accommodate actions that are performed by multiple actors.

1.5 Outline of the dissertation

The other chapters in this dissertation are structured as follows:

- **Chapter 2** presents a comprehensive review for the current state of the action recognition field. The evolution of the human action recognition research in the past decades is presented to provide a general introduction to traditional and modern approaches for human action recognition, including various input features, feature extraction, representation learning and classification techniques that have been proposed by other researchers.
- **Chapter 3** proposes the skeleton-based motion and energy features, which we call primitive geometric features (PGFs), for action recognition. For feature encoding, we converted the extracted features from the skeleton sequence into static images in order to take advantage of the CNNs in extracting spatial patterns from the images. With the aim to speed up the training process, we devised a novel loss regularizer - correctness-vigilant regularizer (CVR).
- **Chapter 4** follows the proposed idea in Chapter 3, and presents several geometric relational features, collectively called the skeleton-based optical flow guided features (SOFs). A new spatio-temporal kernel is proposed to extract better representation for the input geometric relational features. Different from Chapter 3, the geometric relational features are fed into a RNN model in Chapter 4. An attention mechanism based on the RNN model will be explained in this Chapter. In addition, in order to ensure the attention mechanism is more flexible, the window-size and stride are introduced in this mechanism.

- **Chapter 5** presents a novel model for the video-based HAR, which uses the extracted features with the VGG model from the segmented video frames as input. With the aim to alleviate the influence of the changes of illumination, subject appearance and backgrounds, the pre-trained CNN model is adopted to extract the images-based spatial features. As observed in the previous exploration, we note that it becomes much more difficult to improve the accuracy taken video as the input even with a powerful model and computation that are available. We proposed a novel framework, which can extract the skeleton data from the video directly to improve the recognition accuracy.
- **Chapter 6** aims to explore the potential of the proposed geometric features further. We devised a multi-stream CNN model to fuse different features. In order to determine the most efficient features for the multi-CNN model, we explored the performance of different combinations of input features for the proposed multi-stream CNN framework extensively on the UTD-MHAD dataset.
- **Chapter 7** presents a STN model, which can transform the input features to augment the training dataset, so as to enhance the robustness of the trained model on the testing dataset. The proposed skeleton transformer functions as a redundant information cleaner, which can remove the irrelevant information and select out the important features from the original input sequences to formulate a more discriminative sequence.
- **Chapter 8** summarizes and concludes this thesis, highlights the contributions, states the limitations and points out the future research options.

Chapter 2

Literature Review for Human Action Recognition

This thesis aims to address the representation learning problems in the HAR-based system using multiple features, namely, the skeleton and the RGB videos, by utilizing the powerful feature extraction ability of the CNN models and the RNN models. In this Chapter, we will discuss and review Action Recognition and Deep Learning since these are the two topics that are most related to this thesis.

2.1 An overview of action recognition

In general, the goal of human action recognition based on computer vision is to enable machines to understand people's actions in the visual scene through the camera. We can generally categorize the approaches for action recognition into four categories with respect to the types of input data used, such as RGB video-based HAR, still image-based HAR, image sequence with depth information-based HAR and skeleton-based HAR.

Among these approaches, the RGB video-based HAR is the most popular approach for traditional action recognition. It is promising in intelligent surveillance, robotic vision, content-based video retrieval and other related areas. Recently, some researchers have started to implement action recognition using still images [21], because they believe that humans can recognize the actions from only one picture. They intend to develop a powerful algorithm to enable the machine to recognize the action types from the still image. In reality, the human actions are mainly characterized by the state of the human body and the objects the human interacts with in three dimensions. The traditional RGB video-based approaches

project the human movements information on the plane that is perpendicular to the axis of the camera, which will lose the 3D information and as a result it will be more difficult to recognize complex actions. On the contrary, the latest developed depth-camera can compensate for the lost information by capturing the depth information. This promoted the progress of the depth-image sequence-based action recognition. After applying the highly accurate pose estimation algorithm into the depth camera, we can estimate the human skeleton sequence in 3D space from those depth images obtained by the depth camera directly. The human skeleton sequences are more likely to accurately reflect the variation of body posture in the 3D space and therefore action recognition based on the human skeleton has become a new hot spot [22]. Although there are still no official definitions for action recognition, we often refer to action recognition to classify the action types from the sequential data. In the skeleton-based approaches, the skeleton sequences are extracted first, and then those skeleton sequences are used to represent the human actions. Currently, most research studies mainly focus on mining the space and time-varying characteristics of the skeleton sequence and this approach is easier to accept intuitively. Still image-based action recognition can be treated as understanding the scene or reconstructing the scene through a still image, which primarily relies on the human posture and the surrounding environment to judge the action of the subjects in the still image. This is not trivial because it discards the dynamic information of the actions, but it can be an effective auxiliary method for action recognition.

Action recognition (a.k.a. Activity Classification), a general concept, includes gesture recognition, action recognition, interaction recognition and group activities recognition [23]. Gesture recognition mainly distinguishes different types of gestures by the movement of human body parts, such as "stretch out hands", "kick legs" and some other kinds of simple gestures. This task is relatively simple and the difficulty of this task is to obtain the representation of the dynamic characteristics of the movement for different gestures. The action recognition is to determine the action label of one person or a group of people from a video. The target actions for this task are the combination of different types of gestures of different body parts, such as running, waving and jumping. These actions rely more on the time-varying characteristics of the relative motion of different limbs. The interactive action mainly refers to the actions between different people and actions as well as between people

and objects. In order to enable the computer to recognize interactive actions, we not only need to analyse the movement of different body parts, we also need to consider the state between the actor and the objects. For example, in terms of kicking a ball and kicking, the patterns of these two kinds of movements are similar, and the distinctive feature to distinguish these two movements is whether there is an interactive action between the people's leg and the football. The group activity recognition can be divided into two sub-categories generally, order and non-order group activity. For example, there are some regular patterns for a military parade as opposed to the people on the street, in which there are no common patterns. Both of these two kinds of group action recognition rely on the general state of the group and the context environment information of these subjects. For ordered group actions, the regular patterns for ordered group actions are useful for the recognizing process, while there is no regular pattern for non-order group actions. In other words, the group action recognition process is a scene understanding process in most cases, and we can judge the group actions from one image of the groups whereas if we want to recognize an individual action of a group, we can convert the group action recognition into individual action recognition. We need to segment each of the interested people from every frame and get the motion sequences of the interested people. Generally the above four categories can be grouped into two major categories, i.e., action recognition (referred to as individual action recognition) and activity recognition (group action recognition). Sometimes, we use action recognition to represent all of these categories.

Considering the real-world applications of action recognition, we can categorise the action recognition problems into action recognition from a single view and action recognition from multiple views. The former tries to classify different types of action through the spatial movement of the target in the vision scene from one point of view, while the latter one tries to improve the accuracy and robustness of the classification through analysis of the relationship of the action features from different cameras. The action recognition research at the early-stage was mainly based on small datasets that were collected in some controlled settings, with fixed background and illumination, to model the movement of a single front target. Nevertheless, in reality, the scene's background is much more complex, so action recognition of such a complex scene is much more challenging. There are many databases available for action recognition, such as WEIZMANN [24], KTH [25], WEIZMANN [26],

UTD-MHAD [27], NTU RGB+D [20] and all of these public datasets fuel the action recognition research rocket. Fig. 2.1 shows the chronology of the development of these databases and their characteristics. It is evident that they are becoming more and more complex, and the scale of actions included in the database is becoming greater and the amount of the data is also increasing. In the future, we foresee that action recognition will be implemented based on a large-scale dataset from complex scenes.

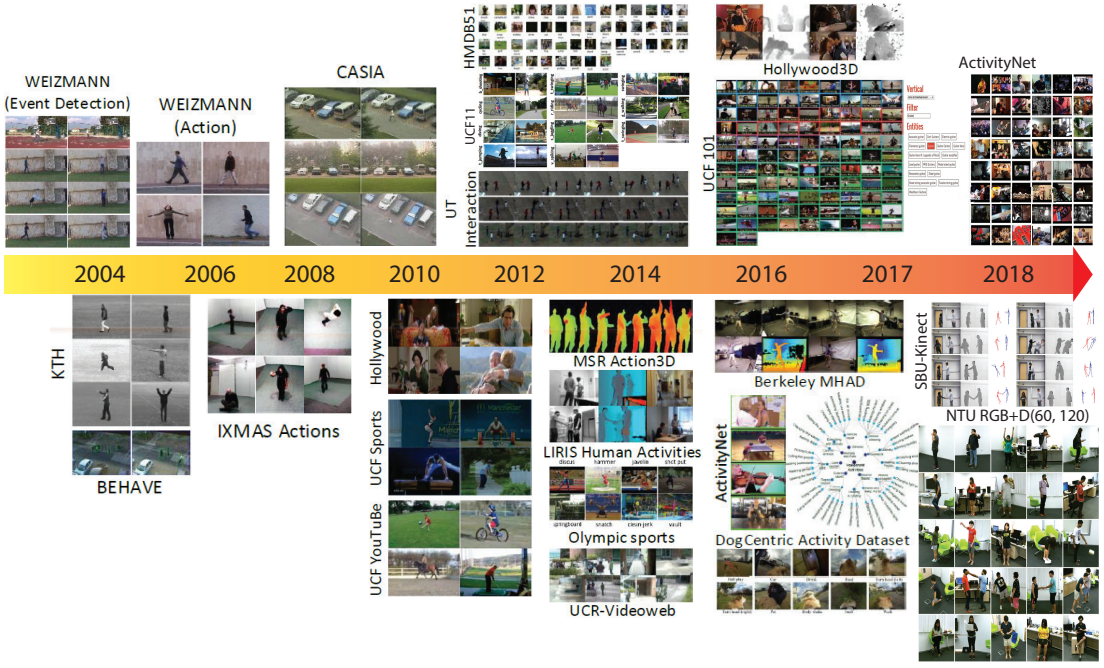


Figure 2.1: List of available databases for HAR

An action recognition system is usually composed of the following five steps: feature extraction, feature selection, feature coding, feature fusion and classification. Feature extraction is the fundamental step for action recognition. In the traditional video feature extraction process, the system usually applies 3D filters to detect the keypoints in the time-spatial cube first, and then we use manually designed multiscale filters to extract the surrounding texture features of these keypoints, and finally we get the representation from the features of keypoints' surrounding texture. The filters used to detect time-spatial keypoints are called detectors. Classic detectors include the Cuboid Detector [28], the Harris3D detector [4] and the Hessian Detector [6]. The group filters that are used to extract texture features around the keypoints are called Descriptors. Popular Descriptors are the Cuboid [28], the HOG3D [29], the SIFT3D [30], the HOG/HOF [31] and the ESURF [6]. The

low-level feature extraction process from video is independent from the target problems, and the feature selection process needs to consider the specific situation for different tasks to extract discriminative feature in order to get discriminative representations. Dense Trajectory [32]–[34] is one of the best methods, and Slow Features [35] have also presented excellent performance in many kinds of action recognition tasks. Another research area associated with the traditional action recognition methods is feature coding. The classical coding methods include Bag of Words/Features (BoW/BoF) [32]–[34], Sparse Coding [36], [37], local Soft Assignment Coding [38], Fisher Vector Coding [39], [40] and its non-probability version, VLAD [41]. The purpose of feature coding is to map those extracted features to a new hypothesized feature space non-linearly in order to improve its distinguishability. In general, while the single features have limited ability to distinguish different actions, the multi-features based action recognition combines some different kinds of hand-crafted features in order to utilize more information of the video. In addition, feature fusion is also a common method to select representative information extracted from videos with different methods and reduce the dimension of the representation [42]–[44]. The SVM model is one efficient classifier for action recognition [45]–[47], and the contribution of the SVM-based approaches is that researchers have attempted to propose novel kernel functions or to use MKL to improve the performance of the SVM classifier. Some researchers have also investigated SSVM [48] to solve the action recognition problems in a multi-view scenario.

In summary, we can generally categorize action recognition into traditional approaches and the deep learning-based method, and there are still many problems that need to be addressed for this task. Deep learning, an end-to-end learning model, has received too much attention from academia and industry because this model can put all those pre-processing phases into one model. While the traditional approach is to implement every phase independently, which made it challenging to control the system's learning performance. In the next sections, we will review these two approaches extensively.

2.2 Traditional approaches for action recognition

There is a rich tradition in computer vision for image sequences studying [49]. Traditional action recognition mainly used hand-crafted features to model the action's evolution. Based on a review of the literature, the following three major types of input features, e.g. RGB Video, Depth Video and RGB+D video, are the most efficient features for action recognition.

2.2.1 RGB Video-based action recognition

In the early stage of action recognition, [50] proposed to implement RGB video-based action recognition by matching Motion History Images (MHI) and Motion Energy Images (MEI), which describe the movements by their appearance. Fig.2.2 shows the appearance of two movements projected on a plane from a specified view (view-based approach) along with time, which not only demonstrated the motion posture of the people and the variations of posture along the time axis. Even though this kind of representation was designed to recognize the motion directly, which is efficient in some simple environments, it cannot work properly in complex scenes and the performance is degraded due to the various backgrounds, illumination and other environmental factors. Action recognition in complex scenes usually relies on the time-spatial keypoints detection [4], [6], [28] to obtain the interest region of a video, and then it gets the texture feature around these keypoints by using some feature descriptors [6], [29]–[31], and finally it builds the classification model based on the statistical representation. Schuldt et al. [51] obtained the histogram of vision vocabularies with the Bag of Words (BoW) model based on the local hand-crafted features, and then the obtained histogram is fed into the subsequent model to build a classifier. This method was popular and led to the current version of the research in action recognition with the BoW model. However, the pure BoW-based model got the global representation from the video and discarded the distribution information of the visual features in the time spatial-space, which is important for action recognition. Therefore, Kovashka and Grauman [52] proposed a method to obtain the local time-spatial feature representation with the BoW model based on the raw features that were obtained in the previous stage, and then they were able to get the context feature representation with the time-spatial pyramid model [53]–[55]. Although, this kind of model can extract context features from the videos, it did not reflect the motion features of the

moving individuals. In order to address this problem, Wang and Mori [56] introduced the Deformable Part-based Model (DPM) into the video feature extraction process to model the action based on the local and global motion characteristic. Xie et al. [57] also modelled the action based on the DPM model. They described the action using a sequence of postures of the actors, which relies on the extracted postures from each frame. Similarly, Tian et al. [58] also built a similar spatio-temporal part-based model to model the human actions.

Figure 2.2: MEI and MHI based approach [1]

The BoW model mainly relies on the spatio-temporal representation that is obtained from the surrounding regions of the keypoints. This kind of model is efficient when the scene is simple, as graph (a) of Fig.2.3 shows, the spatio-temporal key points are mainly situated on the foreground. Yet, as shown in graph (b) of Fig.2.3, if the background becomes complex, it is not easy to find a perfect detector to direct the keypoints to track the moving object in the foreground, since it needs to be further processed by using dimension reduction algorithms.

a b

Figure 2.3: Detection of spatio-temporal interest points [4], [59]

Zhang and Tao [60] extracted the time-spatial cube along with the boundary of the moving objects in the video, and then they obtained the slow feature representation from these cubes

with the Slow Feature Analysis (SFA) and used it as the input of the classifier. This method is characterized by obtaining the efficient features with motion boundary detection.

Figure 2.4: The workflow of feature extraction process of Dense Trajectory [33]

After a comprehensive review of the existing approaches, we can find that the most effective feature extraction method is to extract features along with the Dense Trajectories (DT), which was proposed by Wang et al. [33], [34]. Messing et al. [61] first proposed to represent the motion information with the motion trajectory of the keypoints. This method first detects the keypoints using the Harris3D detector, and then obtains the trajectory with the Kanade-Lucas-Tomasi (KLT) algorithms [62]. However, the accuracy of the keypoint detection is sensitive to the background, and the representation ability is limited. Based on the previous work, Wang et al.[33] proposed to select the discriminative features along with the trajectory of the keypoints, which was also called improved Dense Trajectory (iDT). As Fig.2.4 shows, they first applied dense sampling in each spatial scale, then they tracked these sample points and optimized the trajectory, and finally they utilised the BoW model to obtain the features around the trajectory point. After this, in order to deal with the jitter problems in camera, Wang et al. [21], [34] adopted a similar method with [60] to obtain the feature representation along with the boundary of the moving object. They also used the Motion Boundary Histogram (MBH) to represent the foreground motion information, they then fused the trajectories-based HOG and HOF features. These DT-based features outperformed the traditional methods because they extracted the trajectory features from different spatio-temporal pyramids.

The camera viewpoint keeps changing in realistic applications, while most of the current research studies are utilising databases that were captured from a single view. The videos of these databases have recorded the information of vision scenes projected on the plane that are perpendicular with the axis of the camera, and this cannot reflect all of the information in the

scene (especially those moving objects in the foreground) because of the cluttered background and occlusion. With the advance of action recognition in a single camera viewpoint, some researchers began to implement action recognition under multi-cameras settings. For this task, the main purpose is to extract the motion features that are independent from viewpoint. Souvenir and Babbs [63] extracted the view-invariant feature of people by using R-transform and Manifold Learning. The common features of the multi-view action recognition methods is to extract invariant representations for different viewpoints to improve the robustness of the model to view-variant.

2.2.2 Depth video-based action recognition

The depth camera can acquire depth information that was neglected by traditional 2D cameras, which can describe the movement of the target object precisely in 3D space. With the advance and prevalence of low-cost low-power depth cameras, action recognition based on the depth camera gained significant attention from researchers and from industries. The depth video-based action recognition includes depth image-based action recognition, skeleton sequence (estimated from depth image sequence) based action recognition and combination of both to implement action recognition. In this section, we will review these three methods.

Figure 2.5: The workflow of Histogram extraction from Oriented 4D Normals [64]

For the depth video-based action recognition, most of the research studies have been conducted by transferring the traditional methods for RGB video to depth video. Li et al. [65] proposed a Bag-of-Points (BoP) model, which utilise Action Graph to model the dynamics of actions in the temporal domain. Xia et al. [66] built HOJ3D based on skeleton joints, and quantized the HOJ3D feature using BoW after LDA transformation and then modelled the sequences by HMM model. The work of Oreifej and Liu [64] extracted the normal vector of the

depth sequences in 4D space (time, depth and 2D coordinate space) to obtain HON4D, which is the final representation feature that is fed into the classifier. The process is illustrated in Fig.2.5. Song et al. [67] extracted the Body Surface Context texture features through the 3D point cloud. This method is computation-intensive and Lu et al. [68] improved this algorithm by proposing "π tests" based BRSF model. As Fig.2.6 shows, the depth image is divided into three layers, labelled as green, red, and blue, representing background, action and cluttered region respectively. The pixel-pair 1 to 6 indicates the same background, cluttered region, the same action and background, the same background and cluttered, and same cluttered and action.

Figure 2.6: Example of BRSF [68]

Traditional depth-image based action recognition models rely on the specifically designed feature descriptors to extract the spatial and time-varying characteristic from the 3D cloud, which belongs to the probability method, while the current skeleton-based action recognition models mainly use the physical parameters to represent the local features of the movements. Traditional skeleton-based model can be categorized into two categories: 1) extract the posture and time-varying information from the skeleton sequences; 2) extract the key posture to implement template matching. We will introduce these two methods in the next paragraphs.

Aiming at extracting the spatial and time-varying information from the skeleton data efficiently, Wang et al. [69] proposed a pose-based model, as shown in Fig.2.7. In this model, they estimate the coordinates of the joints of the human body with an improved posture

estimation algorithm. Once the coordinates of the joints are available, a pose dictionary is created with the five parts coordinates of the skeleton. Then, the spatio-temporal representations can be extracted based on the created dictionary. Chaudhry et al. [70] applied LDS to extract the time-varying information from the skeleton data. Vemulapalli et al. [71] mapped the human skeleton-based 3D geometry features into a Lie Group with the mapping parameter between the rotation and local or global coordination of the skeleton joints. Then, they used the Dynamic Time Warping (DTW) algorithm and the Fourier Temporal Pyramid (FTP) to simulate the evolution of the human movements. While all of the models mentioned above attempt to extract the local representation from the skeletal data, which cannot represent the evolution process of action globally, while this is significantly important for HAR.

Figure 2.7: Pose-Based action recognition model [69]

Action recognition is essentially a time-series analysis problem. Lv and Nevatia [72] used the HMM model to model the action evolution process globally. Wu and Shao [73] used the DNN model to estimate the transmission probability of the HMM state, and then to predict the action types. Gong et al. [74] used a Kernelized Temporal Cut model to align the skeletal data with the movement, and the DMW to measure the similarity of the actions. The performance of using the HMM model to simulate the evolution process of actions is limited by two difficult problems: 1) align and segment the input sequence data; 2) estimate the transition probability. The DMW model involves a great amount of computation, which is challenging to implement in real-time applications.

Even though human skeletal data can describe the variations of human posture accurately, but it discards the appearance information from the depth image, which consequently

degrades the performance of the complex action recognition, such as interactive actions. In order to address these problems, Wang et al. [75] used joints as keypoints and extracted the texture features surrounding these joints, which are called the Local Occupancy Patterns (LOP). And then they utilised the FTP to estimate the skeletal data sequence and used the SVM model as the classifier to implement classification. Yang and Tian [76] have also carried out some similar work. In conclusion, the combination of the depth images and skeletal data to achieve action recognition is an efficient and popular approach. Treating joints as keypoints and extracting spatio-temporal features from depth images actually is a feature-selection process. Another recent work [77] proposed using graph convolutional neural network to extract the spatio-temporal features from the skeleton sequence. This model is illustrated in Fig.2.8. The significant limitation of this model is that they need to design the graph specifically for different tasks, even so, some recent advances have been achieved based on this model.

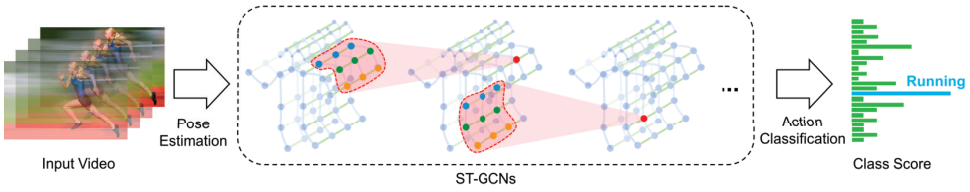


Figure 2.8: Framework of ST-GCN model [77]

2.2.3 RGB+D video-based action recognition

Due to the advent of the cost-effective Kinect sensor, researchers have started to devote a lot of attention to recognizing actions using the RGB-D data [59], [64], [75], [78], [79]. With the additional depth information, it provides an efficient method to remove the background and simplify the intra-class motion variations. Some researchers proposed some semi-supervised learning methods for action recognition, which has made it possible to exploit the large amount of conventional RGB data [80]. They utilized the RGB-D data as the source database and learned the correlations between the RGB data and the depth data, which can be transferred to the target RGB video database.

The RGB-D based action recognition is a typical feature fusion problem, because it relies on the features that have been extracted from both the RGB images and the depth images.

Figure 2.9: Transfer learning for the RGB Dataset (a) and BHIM model (b) [80], [81]

Chaaraoui et al. [82] extracted the skeleton and 2D silhouette of people, and then represented the human actions by fusing these features. Lin et al. [83] treat depth information and skeleton data as an auxiliary method for RGB video-based action recognition, which reconstructs the action information from this auxiliary information to adapt to a different database. Jia et al. [80] utilised the Low-Rank Transfer Learning method to model the subspace of the Depth and RGB data. As shown in graph (a) of Fig.2.9, they transfer the depth information into the RGB video through the Cross-Modality Regularizer, introducing different modality features to solve the RGB-based action recognition problems. Kong and Fu [81] proposed the BHIM model, shown in graph (b) of Fig.2.9 , to projects the RGB and depth features onto a common public subspace. Yu et al.[84] devised a new LFF descriptor based on the gradient field of the RGB and the depth sequences.

2.3 Deep learning-based action recognition

The previous section reviewed the traditional methods for action recognition. However recently, the focus of this research relies less on traditional methods and more on the deep learning-based methods. This shift has been triggered not only by the availability of the computational resources, it is also by the achievement obtained by Krizhevsky in the ImageNet Competition in 2012. The deep learning-based action recognition approach mainly depends on the convolution and recurrent neural network to extract the spatial and temporal dynamic features of the image or skeleton sequences.

Kim et al. [85] first introduced deep learning into the action recognition research. He obtained the volume of the front object through segmenting the front object from the background, and then extracted the action descriptors by using the Gabor filter. The extracted descriptors are then fed into a 3D-CNN model to extract distinctive feature further for the subsequent classifier, WFMMNN model. Fig.2.10 demonstrates the framework of this model and the structure of the WFMMNN. The contribution of this work is that the author firstly proposed the 3D-CNN model and then used WFMMNN as the classifier.

Figure 2.10: The architecture for 3D-CNN model [85]

After this, Baccouche et al. [86] utilised the LSTM-RNN to recognize actions, using BoW to obtain the representation of each frame, and then fed these representations of the image into the LSTM-RNN model, in which the temporal representation is extracted. As shown in Fig.2.11, we can get the final decision from the last frame. Even though this model is simple, it is the first time to extract spatial and temporal features simultaneously. This model became one of the most popular approaches in action recognition and video analysis, and most of the approaches that were developed later are based on this model. It is worth mentioning that many ideas and algorithms have been developed based on this work in the past years and to this day it still has an influence on the current research, with the dramatic changes that have happened to the data, models, algorithms and computing power.

In 2011, Baccouche et al. [87] improved the model proposed by themselves in 2010, which use 3D-CNN model to extract local spatial representation and use LSTM-RNN model to extract temporal dynamic information from sequences of spatial representation. This is the first time to implement action recognition based on original video sequence through Neural Network, which is a popular model to realize action recognition and another popular one is the Two-Stream [88] model. In the same year, Le et al. [89] proposed ISA, as shown in Fig.2.12, which

Figure 2.11: The framework for LSTM-RNN model [86]

is a combination of the traditional methods and deep learning.

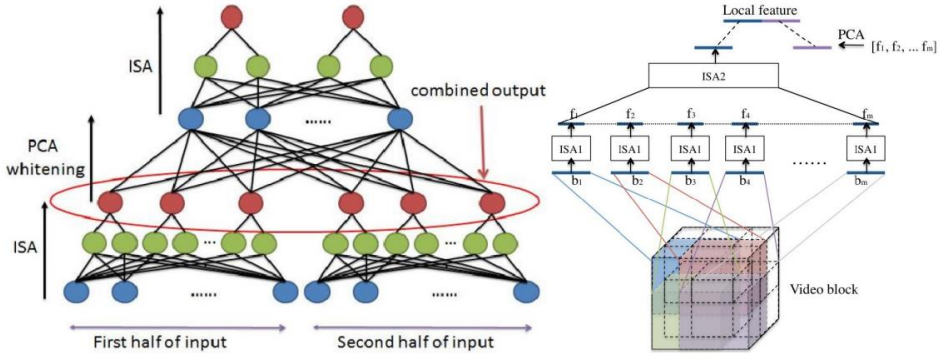


Figure 2.12: Stacked convolutional ISA network [89]

Alex Krizhevsky et al. raised the new current of research on the neural network, because the significant achievement was attained on ImageNet Competition in 2012. After that, many researches were conducted in artificial intelligence using CNN [12]–[14], [88], [90]–[98], which promoted the development of DNN. Karpathy et al. [91] first conducts research on large-scale video analysis using CNN in 2014, they tested four different architectures of the CNN model, showing that the later fusion demonstrated better performance. In 2014, Simonyan and Zisserman proposed a Two-Stream model [88] on the NIPS workshop, which is considered as the most significant breakthrough in deep learning for video analysis. Fig.2.13 shows the architecture of this model. This model proposes adopting two convolutional neural networks to take advantage of the complementary information of these two sub-models. These two sub-models first extract the static spatial information from the vision scenes (spatial stream) and the dynamic information from the temporal stream, which is reflected by the optical flow. The extracted features were then fused in the last decision layer. This model is pre-trained on the large image dataset, which is used to address the

limitation of the low resource problem of the training data. This is a common strategy for all applications that are based on the video analysis and action recognition.

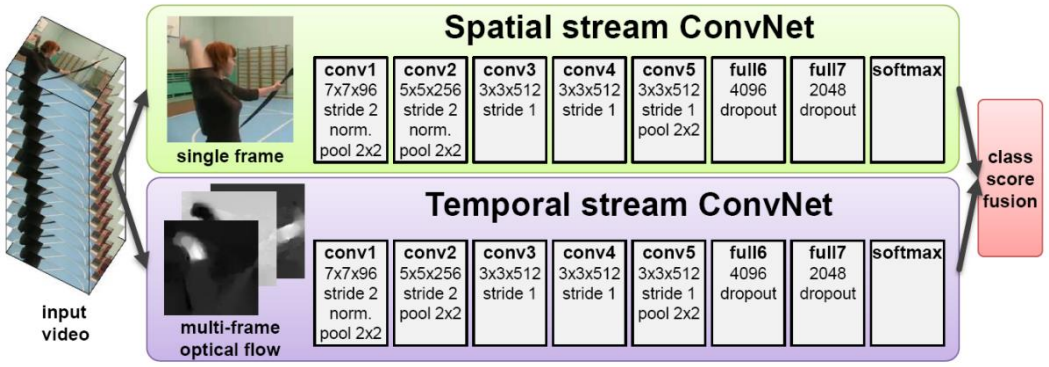


Figure 2.13: The framework for two-stream model [88]

The Long-term Recurrent Convolutional Network (LRCN) [99] and the Temporal Segment Networks (TSN) [100] are two popular deep learning models that were proposed for video and action classification. The LRCN model firstly used the CNN model to obtain the spatial representation of the video frames, and then utilized the LSTM-RNN to model the time-varying dynamic information between the sequential frames. In general, this model is similar to the works of [87], the biggest difference is that [87] trains the 3D-CNN and the LSTM-RNN separately. However, [101] proved that this kind of model could not improve the performance of the system efficiently, but it increased the model's complexity and the number of parameters. After this work, various models were proposed based on this model. Wang et al. [102] combined the idea of [32]–[34], selecting features along with the trajectory by using the pooling operation of CNN to obtain efficient feature representation, and then combined it with the improved Dense Trajectories features [32] to conduct the classification. In another study [103], the authors proposed to address the universal limitation of existing models in the literature with a novel model, which can accommodate variant length of video sequences. Wu et al. [101] combined the merits of the LRCN and the Two-Stream model. They firstly obtained the representation of images and optical sequences by employing a pre-trained model, then they utilize the LSTM-RNN model to obtain the global representation of the sequence and finally they built a regularized feature fusion network to fuse the features to implement classification. The TSN model, as shown in Fig.2.14, is also a two-stream model to take care of the spatial and temporal features that are extracted from

the video snippets.

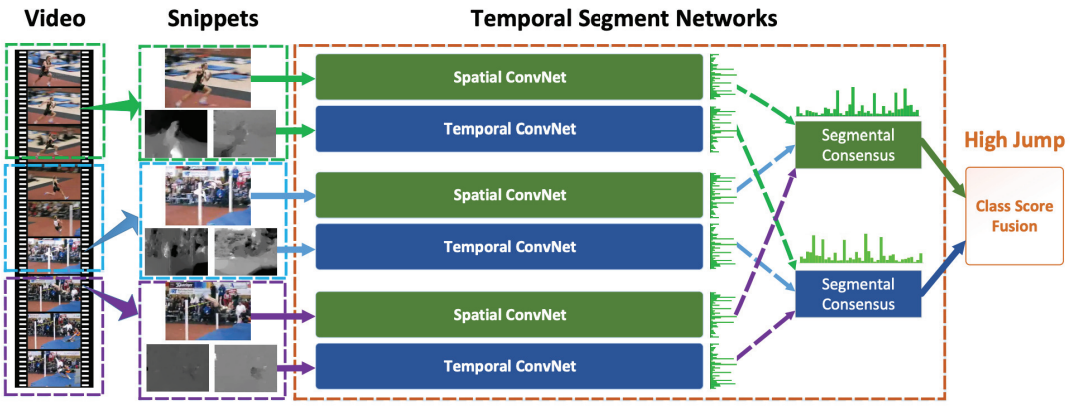


Figure 2.14: Current most popular video analysis framework [99]

Wang et al.[69] borrowed the idea of the two-stream model to solve the action recognition problem, but they converted the representation of multi frames into a video representation of fixed length by utilising the temporal pyramid pooling operation based on 3D-CNN. Another universal model, as shown in Fig.2.15, was designed by Du et al. [97]. This model is applicable for action, action similarity labeling, dynamic scene recognition and object recognition task. Even so, this model showed its limitation on the extraction of dynamic representations for actions.

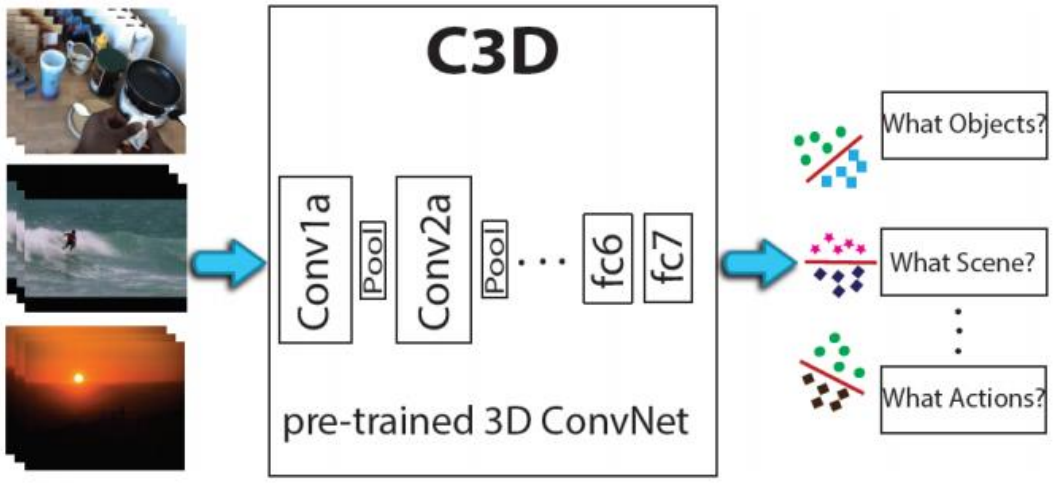


Figure 2.15: Unified framework for video analysis [97]

Lastly, there are some other deep learning models that are proposed for action recognition. For example, Sun et al. [104] replaced the ISA model in [89] with the SFA model, and built an

unsupervised deep learning network with a slow feature fusion strategy for action recognition. Srivastavs et al. [105] provided an unsupervised model , which can map different length of input sequences to a fixed length of output sequences, which can also be used to reconstruct and predict the sequence data. The architecture of this model is shown in Fig.2.16.

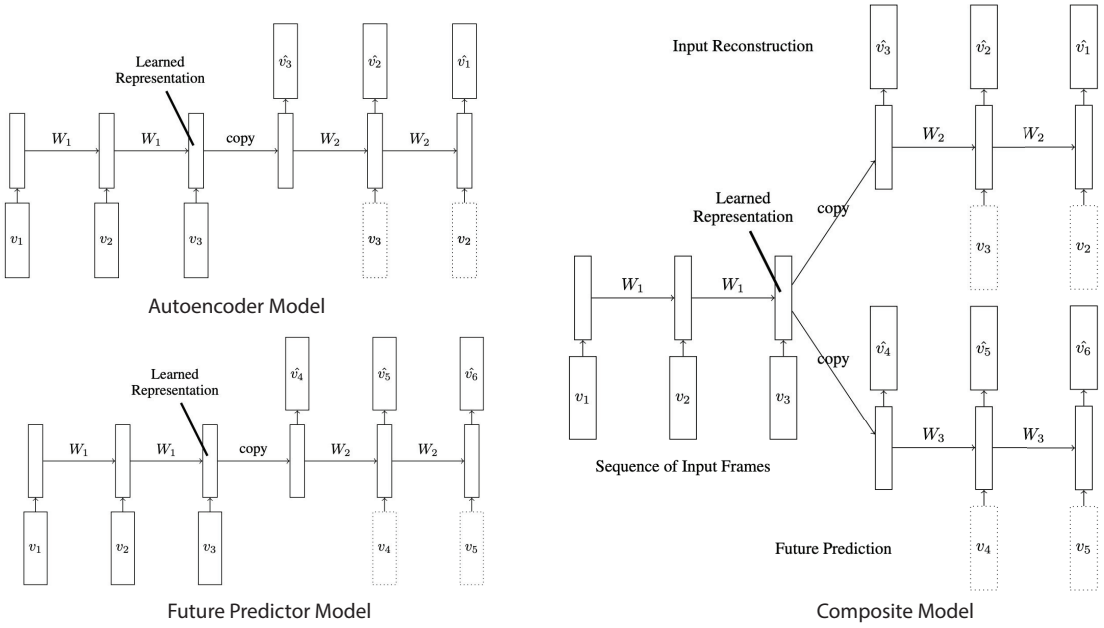


Figure 2.16: Unsupervised video representation learning model [105]

In order to train an robust action recognition model from the video stream continuously, Hasan and Roychowdhury [106] proposed a framework based on the Sparse Autoencoder and Active Learning. This framework can select distinctive features and utilise the unlabelled samples to increase the accuracy of the pre-trained model further. Motivated by this, we proposed to use the LSTM model to transform the input sequence adaptively so as to augment the training samples and improve the performance of the skeleton-based action recognition.

2.4 Summary

A thorough and deep review of the current state of action recognition research studies is presented in this Chapter. We generally categorized the research about action recognition into three research lines according to the type of input data used, such as skeleton, RGB video

and depth video. Since there is no one perfect feature extraction method for different input data, and given that the skeleton data has already contained the depth information, we therefore focus more on skeleton and RGB video-based features and explore the spatio-temporal feature extraction methods from the perspective of these two modality features.

Chapter 3

Skeleton-to-Image based Human Action Recognition

In the machine learning research, the input data, as the lifeblood of machine learning, is expected to provide rich and discriminative patterns for classifiers. Inspired by the image classification with CNN model, this Chapter presents a novel approach for converting the skeleton-based features into static images to carry out action classification.

3.1 Motivation

The Human Action Recognition (HAR) is one promising approach for doing human-computer interaction research, as it is highly vital in addressing the demands of modern society, such as automatic video surveillance for security, patient monitoring for recovery, content-based video retrieval, and so forth. In line with this, deep learning systems are fast becoming the defacto standard for object recognition, video understanding and pattern recognition due to their inherent powerful feature learning ability from a vast amount of data. It makes sense to capitalise on its great success and to further improve it for the complex task of action recognition. Heuristic based approaches for action recognition have attracted an increasingly large and diverse group of researchers. Among the many input features used for action recognition, the 3D skeleton sequence stands out because of recent advancements in pose estimation algorithms [23], [107]. Recently, skeleton-based approaches have significantly progressed using deep learning-based models, since they can help to provide highly accurate spatio-temporal information for action recognition, as compared to RGB or depth videos. They are also computationally inexpensive as compared to traditional appearance-based

approaches, lending themselves more amenable for real-world applications [75].

CNN models have gained a very good reputation as one of the most efficient approaches for solving a wide variety of challenging tasks, e.g., image/video classification [108], ASR [90] and NLP [109]. It has been reported in the literature that CNN models have been successfully applied in HAR tasks, taking skeleton data as input [110], [111]. Motivated by existing regularizers that only consider the correct prediction of the model [112], in this Chapter, we attempted to further improve the discriminatory ability of the CNN model by introducing a new correctness-vigilant regularizer that accounts for both the correct and wrong predictions in the training iterations to speed up the training process. To the best of our knowledge, the proposed output regularizer serves as pioneering work which treats both the correct and incorrect prediction probabilities as two extra supervisory signals in the loss layer. We systematically investigated the efficacy of the proposed model on several popular human action recognition datasets. The empirical results prove that the proposed output regularizer works well with the cross-entropy loss function. Motivated by the feature extraction techniques used in speech recognition [90], we proposed to concatenate the proposed primitive geometric relational features, including the motion and energy features, which are derived from the skeleton sequence, together with the original joint coordinates. Considering the inherent advantages of the CNN models in extracting the spatial features from images, we then encoded the proposed features into the color images that we call the temporal kinematic images, carrying vital motion features. After that, we trained our proposed Correctness-Vigilant Regularized CNN (CVR-CNN) model based on the converted images to classify the actions.

3.2 Related work

In terms of HAR using 3D skeleton data, it has been investigated with different methods over the past several decades [22]. Here, we only review some of the closely related literature with our approach, including the existing geometric features and the regularization techniques that are used together with the CNN model.

3.2.1 Skeleton-based primitive geometric features

The human body movement can be represented as the movements of skeletons that are formed by a hierarchy of joints. In the public databases, typical layouts of the skeletal representation of a human body are usually composed of 15, 20 or 25 joints. [111] concatenated together all the joints coordinates to represent one action by one static image, casting the action recognition problem into an image classification problem. However, they did not consider the relationship of the joints between the different frames explicitly, which severely limits the performance of the CNN model because they cannot extract effective features that can reflect the spatial relationship between the joints. In [113], the authors selected a sequence of informative joints to represent spatio-temporal features in the preprocess stage, which may also lead to losing information. In the existing literature, we observe that all of the parametric representation, e.g., the angle, position, and orientation [66], [114], are used by classifiers directly. Another study [115] proposed several pose-related features, such as distance between joints, distance between the joints and the planes, angle between different limbs. Moreover, [71] proposed using "Trisarea" to describe the geometric correspondences between the joints, which utilize the area of the predefined triangles to represent the geometric features. However, much of the prior research derives the geometric features from only one single frame, therefore this thesis proposes a simple yet highly intuitive set of geometric features, called PGFs, to extend the input representations that were fed into the CNN model. The proposed features, the PGFs, not only consider the relationship between the joints in a single frame but also consider the variation of the joint coordinates through time, by introducing the representative motion and energy features to enrich the texture pattern of the converted temporal kinematic images [116].

3.2.2 Regularized CNN models

Due to its convincing performance in spatial feature extraction, the CNN model has been widely used for image classification, which learns the correlation between the local pixels efficiently by using a different scale of convolutional kernels. However, action recognition using the CNN model based on the skeleton data is still a challenging problem, mainly because of the significant variations of sequences, pertaining to the same action. Therefore, most of the previous research, over a long period of time, attempted to extract discriminative

features using the CNN model with regularization techniques [117]. Many of the techniques were proposed to alleviate the issue of overfitting, such as early stopping, L1/L2 regularization [118], weight decay, dropout [119], and batch normalization [120], model averaging [121], and data augmentation, and so on. These techniques, along with other forms of regularization, almost act on the hidden activations or weights of a neural network. There are also some works explored adding a regularizer on the output layer [122], [123], but all of them only took advantage of the correct prediction probability produced by the softmax layer. Few studies have been conducted to exploit the wrong prediction probability. In our opinion, the wrong predicted labels are also indicative of the knowledge learned by the neural network, which can provide more insight about the training process and can be used to update the corresponding weights more appropriately. To some extent, this augments the supervisory training signals, influencing the formation of the learned features, and making them more discriminative.

Generally, three key facts can be observed from these related works. Firstly, the spatial and temporal representations of the input skeleton sequence are critically important for the performance of the action recognition systems. However, the spatial and temporal representations are of a different nature, and how to represent these two features ideally is still an open question. In this Chapter, we attempted to extend the input representations and use the methods in the image classification area to extract these two critical features simultaneously. Secondly, some actions are characterized by certain joints, and the features that were extracted from these joints will be more discriminative than others. Thirdly, due to the significant variations in the skeleton sequences, the robustness of the model is very important for the efficacy of the classifier on the test dataset.

3.3 The algorithms

3.3.1 General architecture

An ideal human action recognition system favors feature representation that is invariant to different types of human physiques and anthropometric differences between individuals. The features used in this Chapter can be found in Section 3.3.3. Here, we present the workflow of

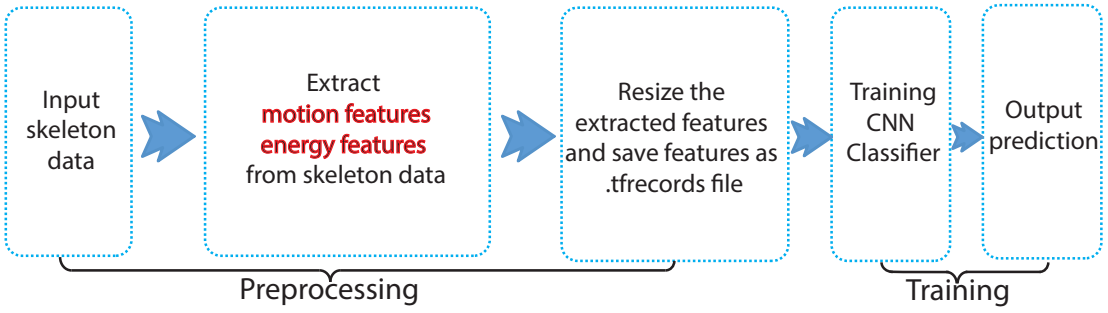


Figure 3.1: The flowchart of the proposed CVR-CNN framework

our proposed model to learn global features from these primitive geometric relational features (represented by the converted static images, refer to section 3.3.3). The workflow of the system is described in Fig.3.1.

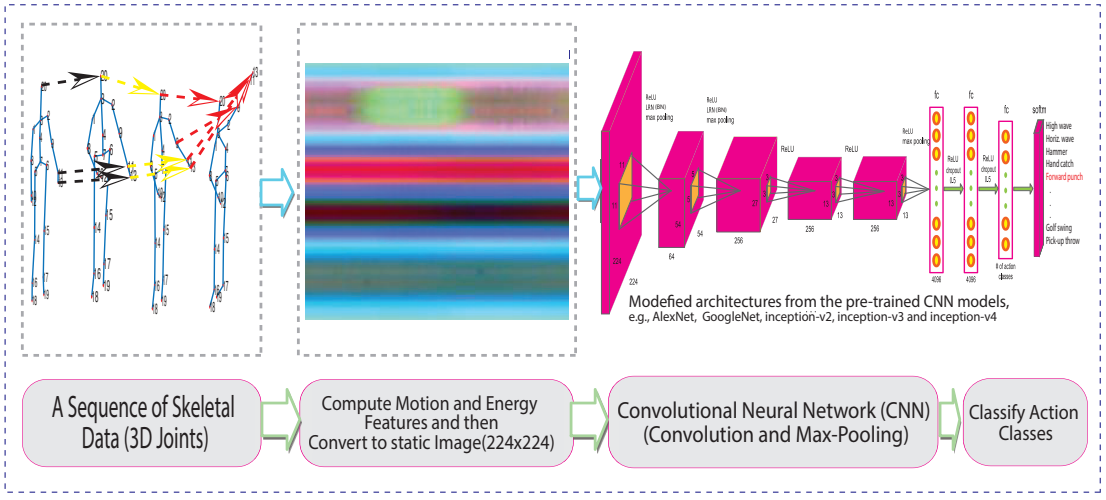


Figure 3.2: Architecture of CVR-CNN Model

Our proposed framework is shown in Fig.3.2, the skeletal data are first converted into static color texture images. In this proposed framework, we have incorporated several advancements. For example, we added helpful features (e.g., displacement information of each joint for the current frame and the previous frame) that describe the motion between the skeletal joints. This idea was inspired by a technique from speech recognition that amplifies the spatio-temporal cues. The color texture images explicitly involve both the joint coordinates, as well as the motion and kinematic energy along with the time domain. Moreover, aiming at speeding up the training process, our method involves a novel output

regularization technique for the CNN model, which is called CVR-CNN, to help extract more discriminative features.

The pseudocode for the proposed pipeline is shown in following code block:

```
videos # given training and testing video/skeleton sequences

num_instance = len(videos)

for i in range(num_instance):
   skeleton  # given skeleton inputs.

   # Motion calculation
   for j in range(len(skeleton)):
      Motion = skeleton(j+1) - skeleton(j)

   # Energy calculation
   Energy = square(Motion)
   Image = [skeleton, Motion, Energy]   # converted image matrix

Train_path, Test_path  # given the train and test path
input = loading_data(Train_path, Test_path)

# Define the proposed model
outputs = PretrainedVGG(input)
outputs_prob = Dense(num_classes)(outputs)

# Training
for i in range(epoch):
   label, features = Load_training_batch(batch_size, Train_path)
   outputs_prob = train_one_batch(loss, accuracy, lable, features)
   predict = argmax(outputs_prob)

   # Calculate metrics for the model
   loss = softmax_cross_entropy(label, predict) # Equation 3.5
   loss_pos = positive_cross_entropy(label, predict)  # Equation 3.9
   loss_neg = negative_cross_entropy(label, predict) # Equation 3.10
   final_loss = weighted(loss, loss_pos, loss_neg)  # Equation 3.14

   # update parameters
   final_loss.backward()
   train_accuracy = reduce_mean(correct_pred equal lable)

   # Testing
   if i % 5 = 0:
      label, features = Load_testing_batch(batch_size,Test_path)
      test(loss, accuracy, lable, features)

      # Calculate testing metrics
      test_accuracy = reduce_mean(correct_pred equal lable)
```

Code block 1: *Convert primitive geometric relational features into images and train CVR-CNN model*

The hyperparameters that can be tuned in the proposed framework are listed in Table 3.1.

Table 3.1: Hyperparameters for the CVR-CNN model

parameter	Description	Default value
Image-size	size of imput images	224×224
lr	start learning rate	0.001
Epoch	number of epoches for training	50
batch_size	number of training instance for each batch	64
Input	premitive geometric relational features (motion, energy)	
outputnodes	equal to the target classes of actions	

3.3.2 Novel input representation of actions

Motivated by [90] and [124], we propose a novel type of features to encode the skeletal motion contained in a skeleton sequence. The main limitation of the traditional optical flow for RGB video is the expensive computational costs, while the limitation of the existing approaches [66], [114] based on the skeleton data is that they mainly consider the relationship between the joints of a human body in a single frame. In order to encode the temporal information, our approach takes the representative motion and kinematic energy that are derived from the consecutive frames into consideration, which explicitly inputs the motion patterns, represented as PGFs, of each joint and body part into the subsequent classifier. The devised geometric and kinematic features are demonstrated in graph (a) and graph (b) of Fig. 3.3, the details of these features will be presented in the following sections.

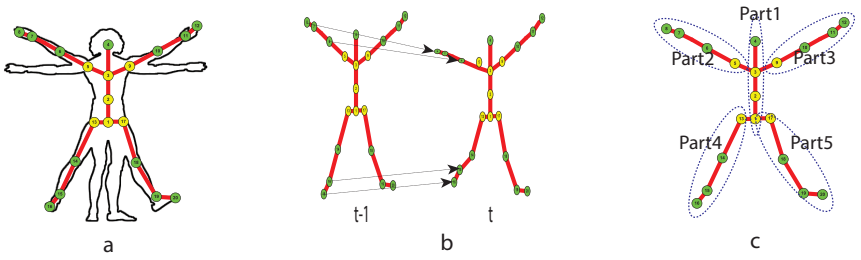


Figure 3.3: Example of typical layout of skeleton data (a), motion features (b) and body-part representation strategy

3.3.3 Encoding primitive geometric features

Du et al., [111] firstly proposed to encode a sequence of joint coordinates into a static image. However, due to the noise data obtained by the pose estimation algorithms, the converted static image, in turn, became noisy in a complex environment. This method emphasized the salient change of some of the joints. Taking as an example, an action "wave left hand", we need to pay more attention to the left hand, and not to all of the frequent slight movements of the other joints. As a result, we propose to combine the joint coordinates, their motion and energy features as the input representation, which simultaneously encode the spatio-temporal features that exist in the skeleton sequence into one static color image.

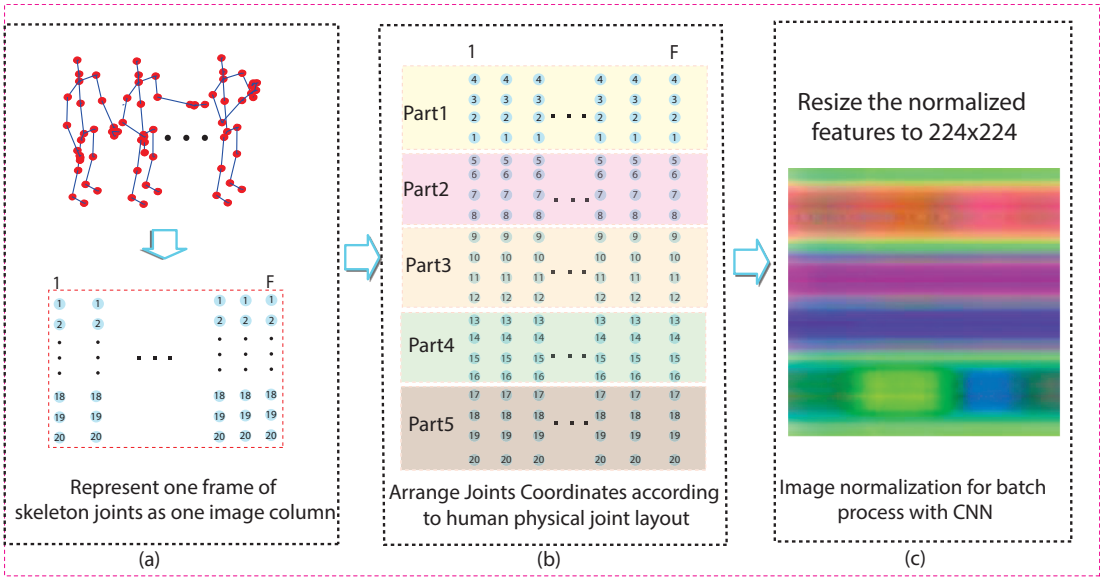


Figure 3.4: The process of conversion from skeleton data to temporal kinematic image representation

As shown in Fig.3.4, for a specific skeleton sequence I, which consists of F frames, and each frame includes N joints, the n_{th} joint of the f_{th} frame can be formulated as $J^f_n = (J^f_{n,x}, J^f_{n,y}, J^f_{n,z})$, where $f \in (1, ..., F)$ and $n \in (1, ..., N)$. From Fig.3.4(c), it can be seen that each coordinate is eventually mapped to a colour channel. The value of N and the accuracy of the joint coordinates are determined by the motion capture system, pose estimation algorithms or the depth information obtained by the depth camera. Fig.3.3(a) shows a popular joint configuration. In order to simplify the problem, we adopted one typical

layout of skeleton, with 20 joints in each frame, to demonstrate the calculation of our proposed primitive geometric features. Pseudo code block 2 provide more details for this process.

```
# arrange the skeleton data according to the predefined joint order (body part representation strategy, shown in Fig.3.3(c))
skeleton_sequence
skeleton_sequence = rearrange_joints(skeleton_sequence, Joint_Order)

# rotate the skeleton sequence relative to the right and left hips( optional step)
# RHip,LHip indicate the joint coordinate of the right hip and left hip
skeleton_sequence = rotate_ske(skeleton_sequence,RHip,LHip)

# interpolate points between adjacent joints(optional step)
skeleton_sequence = interpolate_ske(skeleton_sequence,RHip,LHip)

# normalized skeleton sequence
skeleton_sequence

image = []

for frame in range(len(skeleton_sequence)):
        for joint in range(N) # N = len(frame), number of joints
                x, y, z = skeleton_sequence[frame][joint]
                r, g, b = x, y, z
                image[frame][joint] = r, g, b

output_image = imresize(image,[224,224]);
```

Code block 2: *Represent geometric features with static RGB images*

We followed the configuration of [111], and divided the human body into five parts, such as Trunk, LeftArm, RightArm, LeftLeg, RightLeg, as illustrated in graph (c) of Fig.3.3. As shown in Equation 3.1, $\vec{Part(i)}_M$ indicates the displacement of joints in body part i from current frame to the next frame, which can be extended to the angle-based features and the distance-based features that will be described in detail in Chapter 4. In this Chapter, we only utilize the basic joint coordinates as the geometric features and their derived motion and energy features. The motion-based features indicate the variations of the basic joint coordinates between the consecutive frames, whereas, the energy-based features roughly accounts for the representative energy exerted by the actor within a fixed time period. We will detail the calculations of primitive geometric features and their derived motion and energy features in the following sections. The converted static color image can be denoted as the following matrix, I:

$$I = \left| \begin{array}{c||c||c||c} 1 & 2 & \begin{array}{c}\cdots\cdots\\ \\ \cdots\cdots\\ \\ \cdots\cdots\\ \\ \cdots\cdots\\ \\ \cdots\cdots\\ \\ \cdots\cdots\\ \\ \cdots\cdots\end{array} & F-1 \\ Part1_J & Part1_J & & Part1_J \\ \cdots & \cdots & & \cdots \\ Part5_J & Part5_J & & Part5_J \\ \vec{Part1}_M & \vec{Part1}_M & & \vec{Part1}_M \\ \cdots & \cdots & & \cdots \\ \vec{Part5}_M & \vec{Part5}_M & & \vec{Part5}_M \\ \vec{Part1}_E & \vec{Part1}_E & & \vec{Part1}_E \\ \cdots & \cdots & & \cdots \\ \vec{Part5}_E & \vec{Part5}_E & & \vec{Part5}_E \end{array} \right| \tag{3.1}$$

where each colour pixel corresponds to a coordinate in the skeleton representation. $\vec{Part}(i)_J$ corresponds to joint coordinates; $\vec{Part}(i)_M$ corresponds to motion-based features; $\vec{Part}(i)_E$ corresponds to energy-based features.

3.3.3.1 Representative motion

Human action representations do not only depend on the position of the limbs at the current *timestep* but they also rely on the previous positions at *timestep-1*. Normally, if we want to describe the state of a moving object, we usually present the speed and the direction of that object. For example, if we describe a running human, we need to give both the speed (km/h) and direction (e.g., left to right). Similarly, in this Chapter, as we describe a human's action, we also need to describe both the velocity and the motion direction of the joints, because different joints have a different velocity as a human takes different actions, and the direction will also vary. For example, we define the direction of the motion of the joints by measuring the displacement between the corresponding joints in consecutive frames, which can be calculated by the following formula:

$$\phi_{n,f} = (J^f_{n,x} - J^{f-1}_{n,x}, J^f_{n,y} - J^{f-1}_{n,y}, J^f_{n,z} - J^{f-1}_{n,z}) \tag{3.2}$$

where $\phi_{n,f}$ indicates the n_{th} joints' direction in the f_{th} frame relative to the previous frame, and $(J^f_{n,x}, J^f_{n,y}, J^f_{n,z})$ is the joint's coordinate of the n_{th} joint in f_{th} frame.

3.3.3.2 Representative kinetic energy

Human actions are characterized not only by the motion direction, they are also characterized by the energy, or speed when the human finishes one specific action. For instance, we define the kinetic energy for all of the joints by using the corresponding joints' coordinates in the consecutive frames with the following formula:

$$\begin{aligned}\mathcal{E}_n^f &= k(v_n^f)^2 = \frac{1}{\Delta t^2} k \left| J_n^f - J_n^{f-\Delta f} \right|^2 \\ &= \frac{1}{\Delta t^2} k \sum_{p=x,y,z} (J_{n,p}^f - J_{n,p}^{f-\Delta f})^2 \\ &= \frac{1}{\Delta t^2} k \Big\{ (J_{n,x}^f - J_{n,x}^{f-\Delta f})^2 + (J_{n,y}^f - J_{n,y}^{f-\Delta f})^2 \\ &+ (J_{n,z}^f - J_{n,z}^{f-\Delta f})^2 \Big\}\end{aligned} \tag{3.3}$$

where $\mathcal{E}_n^f$ is the representative kinetic energy, k can be treated as the weight of the person, which is a constant coefficient, for simplicity we set $k = 1$. $\Delta t = \Delta f \times (\frac{1}{30})$, where Δf is an integer number, $1/30$ is the frequency, which is different for each dataset. Different body parts will have different energies when humans perform different actions. For example, Fig.3.5 shows the variation of one joints' kinetic energy, e.g., the Ankle, for two typical actions: "Kick" and "Run slowly".

After computing all the aforementioned geometric features, we then converted the obtained matrix I, refer to formula (3.1) , into a color texture image using a colormap function. In this work, we used the simplest colormap function: $P_n^f = 255 * \frac{p_n^f - min(p)}{max(p) - min(p)}$. Lastly, we resized the converted images into fixed-size of images, e.g., 224×224 images, which will be used as input of our subsequent CNN model. This operation is essential because the shape of the input image should conform to the input requirements of the VGG model. The whole process is demonstrated in Fig.3.4 and several converted images of different features are shown in Fig.3.6. Each of the features is explained in the following paragraphs.

As shown in Fig.3.6, ten different Primitive Geometric Features (PGFs) for four different actions are extracted from the skeleton sequence data. PGF1 represents the skeleton images

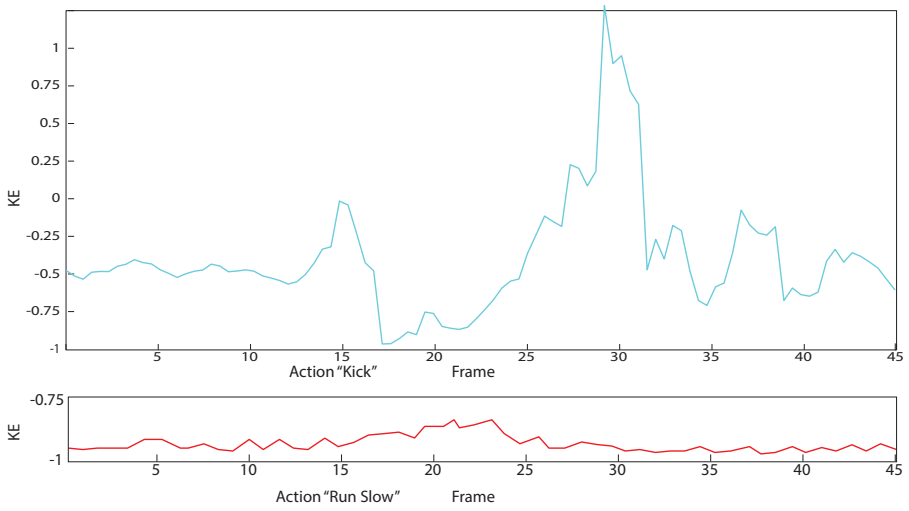


Figure 3.5: Variation of kinetic energy of joint "ankle" for two actions: "kick" and "run slowly"

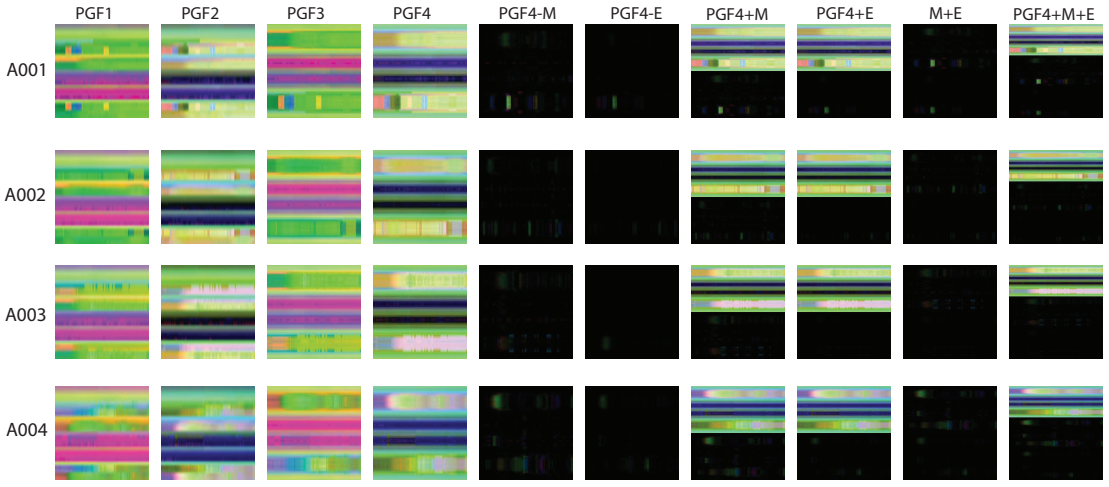


Figure 3.6: Image representation of ten different PGFs for four actions in NTU RGB+D. The code for calculating the above 10 features can be accessed in this link: https://gitlab.com/jren2019/skb_cvr_cnn/blob/master/calSOF.m

converted from the original skeleton data with normalization. PGF2 represents the skeleton images converted from skeleton preprocessing with a global rotation matrix, in order to eliminate the effects of view variations. Shown as graph (a) of Fig.3.7, we transformed the joint coordinates of a skeleton sequence into a new space, which use the 'hip center' as the origin. The z axis of the new space is the same as the orientation of the torso. The direction of

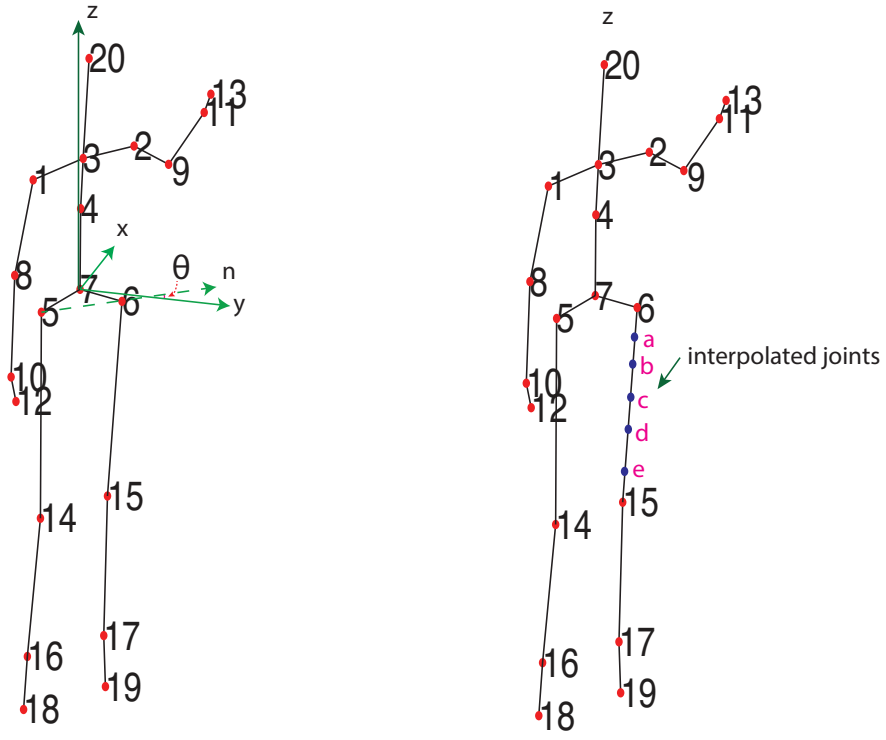


Figure 3.7: Rotation and interpolation of features: PGF2 (left figure) and PGF3 (right figure)

y is obtained by the following formula:

$$\begin{aligned} y &= \underset{y}{\operatorname{argmin}} \arccos(y, n) \\ &\text{s.t. } y \perp z \end{aligned} \tag{3.4}$$

where n is the direction pointing from the "left hip" to the "right hip". The x axis can be derived by $x = z \times y$. The converted skeleton sequence, which used the "hip center" as the origin, is view invariant. For PGF3, we introduced a novel approach to augment the converted image, interpolating several points between adjacent joint coordinates to increase the dimension of converted images. For example, shown as graph (b) of Fig.3.7, the limb connected with *Joint 6* and *Joint 15* are inserted 5 points using the interpolation algorithm. If we leave it to the image resize (scaling) function, and not perform the interpolation between the adjacent joints (by adding more points between joints), then the image resize function will basically perform interpolation on the unrelated joints; thus, introducing noise.

Specifically, if we arrange the joint coordinates as shown in Fig.3.7 and Fig.3.4, the image resize function will insert some pixel values between joint 1 and joint 5, joint 8 and joint 9, joint 12 and joint 13, joint 16 and joint 17. However, all these inserted pixel values can be treated as noise, because they cannot represent meaningful information of human pose; that is, there are no correlations between the inserted pixel values and the physical coordinates of the limbs.

For PGF4, we adopt both the rotation and interpolation approach to extract the skeletal images. The remaining 6 skeletal images are derived from PGF4, which are the result of the different combinations of the original feature, such as motion and energy features. Concretely, "PGF4-M" and "PGF4-E" indicates the motion and energy features based on the skeleton coordinates. "PGF4+PGF4-M" represents the concatenation of the extracted "PGF4" feature and "PGF4-M" feature together to represent the whole sequence. While the "PGF4+PGF-E" indicates the combination of the "PGF4" feature and the "PGF4-E" feature together to represent the skeleton sequence. Different discriminative characteristics of different actions are reflected by the texture pattern of the converted images and the temporal dynamics are reflected by the color patterns along the time axis.

3.3.4 Correctness-Vigilant Regularized CNN Model

As stated in Section 3.3.2, a vast wealth of techniques have been used to regularize the training of the Deep Neural Networks (DNN). [122] proposed perturbing the negative log-likelihood of the correct prediction by assigning a wrong label randomly for image recognition to produce a larger cross-entropy, which is a kind of method for generating a broader, more robust discriminative knowledge from the training patterns by using the hidden knowledge learned by the neural network. In this work, we propose using both the correct and wrong prediction probabilities simultaneously as two kinds of extra supervisory signals to accelerate the training convergence speed. Szegedy [125] argues that putting all probability on a single class in the training dataset is a symptom of overfitting. The proposed regularizer aims at mitigating this problem, making our network better at generalizing beyond the training data.

The output layer of the CNN model defines the score function $f : \mathbb{R}^D \rightarrow \mathbb{R}^K$ that maps the extracted features to the confidence scores for each class. The softmax or $1 - of - K$

encoding method is a standard scheme for the deep learning-based classification problems, where K indicates K distinct categories, at the top layer, to generate the confidence scores. For example, the database UTD-MHAD includes 27 possible distinct actions, so the softmax layer has 27 nodes denoted by p_i, where $i = 1, ..., K$. The value of p_i defines a discrete probability distribution, which satisfies $\sum_1^K p_i = 1$. The output of nodes in the penultimate layer can be represented as h, which is connected with the last softmax layer by w. Therefore, the sum of all the input into a softmax layer, as indicated by z, is:

$$z_i = \sum_k h_k W_{ki} \tag{3.5}$$

Then we have the softmax function, which gives the probability computed for class i:

$$p_i = \frac{e^{z_i}}{\sum_k^K e^{z_j}} \tag{3.6}$$

The predicted class $\hat{l}$ would then be:

$$\hat{l} = \underset{i}{\operatorname{argmax}}\, p_i = \underset{i}{\operatorname{argmax}}\, z_i \tag{3.7}$$

therefore, a neural network produces a conditional distribution $p_\theta(y|x)$ over Class y given an input x through a softmax function. The cross-entropy of this ground-truth distribution is commonly defined as:

$$C_0 = -\frac{1}{n} \sum_i \sum_j^K y_j \log(p_\theta(y_j|x_i)) \tag{3.8}$$

where K represents the number of classes for the target task, the y_j is the one-hot encoding of the target class label for each example, and n is the batch-size.

Apart from the above commonly used ground-truth cross-entropy, we now define the output-positive cross-entropy and the output-negative cross-entropy as formula 3.9 and 3.10. Equation C_1 corresponds to the positive log-likelihood of the correct prediction, where $y_j = y_i$, equation C_2 corresponds to the negative log-likelihood of all of the wrong

predictions, where $y_j \neq y_i$. In these two equations, y_i is the index of the target class label.

$$C_1 = -\frac{1}{n}\sum_{i}\sum_{j,y_j=y_i}^{K} p_\theta(y_j|x_i)\log(p_\theta(y_j|x_i)) \tag{3.9}$$

$$C_2 = -\frac{1}{n}\sum_{i}\sum_{j,y_j\neq y_i}^{K} p_\theta(y_j|x_i)\log(p_\theta(y_j|x_i)) \tag{3.10}$$

According to the information theory [126], the distribution of both the correct and incorrect predictions together influences the generalizability of the trained model. Therefore, we experimented with these two equations to comprise the regularization term used during the training process, which leads to our proposed objective function as:

$$\mathcal{L}_\prime(\theta) = C_0 + (\alpha C_1 + \beta C_2) \tag{3.11}$$

A common problem that occurs while training a classifier using the standard objective function, C_0, is that the confidence levels computed for the patterns in the training set are much larger than those found in the testing dataset. In order to increase the confidence levels of the classifiers in the training process and maintain the training stability, we integrate the proposed regularizers into the standard log-likelihood loss function. Additionally, in order to strengthen the supervisory signal whenever C1 or C2 considerably diminishes, we employ a thresholding function to the output regularizers. As shown in Fig.3.8, the regularization restricts the variation of positive and negative cross entropy in the shaded area, which can improve the accuracy and the generalizability of the model at the same time. In addition, this regularization is designed to help to speed up the training and help to extract more discriminative features from the training dataset.

Furthermore, in order to make the trained model more robust, we add another regularization item once the training process is near the final iterations. We assume that the ground-truth distribution of each class is a uniform distribution, $P = [P_1, ..., P_k, ..., P_K]$. Next, we randomly generate a number $w \in [0.1, 0.2]$, and then we get the uniform distribution as $W_k = \frac{w}{K}$. Therefore, to calculate the smoothed-out probability distribution of

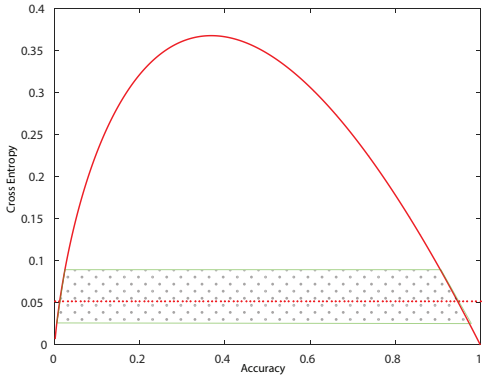


Figure 3.8: Regularized cross-entropy

one example belonging to different classes is:

$$\mathbf{P} = \left[P_1, ..., P_k, ..., P_K\right] = (1 - w) * \left[y_1 ... y_k ... y_K\right]^T + W_k \tag{3.12}$$

With this hypothesized distribution, we applied the discrete KL-divergence formula [127] to compute the loss, which can be represented as:

$$D(\mathbf{P}\|Q) = \sum P(i) \log(\frac{P(i)}{Q(i)}) \tag{3.13}$$

where $Q(i)$ is the output prediction of the model for the given input example to each class. Adding this item to the previous objective function leads to the loss function that can be represented as the following formula:

$$\begin{aligned} \mathcal{L}(\theta) &= \mathcal{L}_\prime(\theta) + \gamma D(P\|Q) \\ &= C_0 + \alpha C_1 + \beta C_2 + \gamma D(P\|Q) \end{aligned} \tag{3.14}$$

In general, this customized loss function helps to provide more insights into the generalization capability of the classifier during training. As a result, it can also help to boost its performance on the testing dataset. The weight parameters in 3.11 and 3.14, α, β, γ, indicate the weights that are used to balance the components of the loss function. f

3.4 Empirical testing and analysis

3.4.1 Datasets

We evaluate the proposed primitive geometric relational features and the regularized CNN model on several widely used action recognition datasets [20], [27], [128] . All the features were derived from the skeleton data provided by the database.

UTD-MHAD dataset: In terms of the UTD-MHAD dataset [27], we followed the evaluation configuration proposed by the author of this dataset, using 431 examples of the first 5 actors as the training dataset, and the remaining 430 examples of the remaining 5 actors are used for testing.

Northwestern-UCLA dataset: The Northestern-UCLA dataset [128] provides skeleton data of 10 action categories, 1494 action sequences in total, which are obtained by using 3 Microsoft Kinect v1 cameras simultaneously. It is designed to investigate the variation of different actions from different viewpoints.

HDM05 dataset: The HDM05 dataset is the largest skeleton dataset that was acquired by using Optical Motion Capture System, including 130 actions with 2337 videos in total. The actions in this dataset are performed by 5 non-professional actors, and each frame of the skeleton data consists of 31 joints. Due to the fact that this dataset is captured from the MoCap system, the produced skeleton includes fewer outliers compared to the outputs of pose estimation algorithms, and the joint coordinates in this dataset are much more accurate.

NTU RGB+D dataset: The NTU RGB+D dataset [20] serves as the latest and the most challenging dataset for HAR, which contains 60 action classes. All of the action instances included in this dataset can be generally divided into three categories, including the health-related actions, the daily actions and the mutual actions. Each action instance contains a sequence of 3D locations for 25 skeleton joints. There are 56880 skeleton sequences in total in this dataset, captured by 3 Kinect cameras, and an extra challenge is posed by the significant intra-class and view point changing for this dataset.

3.4.2 Implementation details

The pre-trained CNN models provide an efficient solution for low-resource problems, which can help to boost up the performance on small dataset. There are some well known pre-trained CNN models available, such as, the VGG [129], the Xception [130], and the Resnet [131]. Due to the expensive computational cost of the Xception and the Resnet model, we adopt to re-train the parameters of the pre-trained VGG model. For the training of the CNN model, the weights of the lower layer in the CNN are learned by backpropagating the gradients from the last output layer. In order to implement this, we need to differentiate the proposed regularization items with respect to the net input of the top layer. Here we denote $L_0 = C_1 + C_2$, which is the cross-entropy for all the classification results. Denoting the i_{th} logit by z_i, the partial derivative of L_0 with respect to z_i can be denoted as the following equation, see 3.15 below:

$$\begin{aligned} \frac{\partial L_0}{\partial z_i} &= -\frac{\partial \sum_{j=1}^{K} y_j \log y_j}{\partial z_i} \\ &= y_i(-\log y_i - C_0) \end{aligned} \tag{3.15}$$

From this point on, the backpropagation algorithm is same as the standard Softmax-based deep learning models. Notably, the gradient of our proposed item, the positive cross-entropy and the Negative cross-entropy, with respect to the logits can be accomplished by the deep learning toolkits (e.g., Tensorflow, Pytorch) automatically.

3.4.3 Results and Comparisons

The summarized results and a comparison with the related literature on four challenging datasets are listed in Table 3.2. We selected several widely reported algorithms in the recent literature as our baseline system, such as, [132], [133], [134], [124]. In contrast to our existing work, our proposed approach achieves competitive results on all four datasets. The performance on both UTD-MHAD dataset and Northwestern-UCLA dataset is improved by adding the motion features to 89.9% and 87.9% respectively.

Table 3.2: Performance comparison for CVR-CNN model

Features	UTD MHAD		Northwestern UCLA		HDM05		NTU RGB+D			
							CV		CS	
CNN + JTM [135]	85.8		-		-		75.2%		73.4%	
CNN + JDM [135]	-		-		-		82.3%		76.2%	
CNN + Optical spectra [136]	86.9%		-		-		-		-	
Deep RNN [20]	66.1%		-		-		-		-	
3D-HoTMBC [137]	84.4%		-		-		-		-	
Cov3DJ [138]	85.6%		-		-		-		-	
CNN + JDM [132]	88.1%		-		-		-		-	
HON4D [64](reported by [139])	-		39.90%		-		-		-	
SNV [140]	-		42.80%		-		-		-	
AOG [128]	-		53.60%		-		-		-	
HBRNN-L [141]	-		78.52%		-		-		-	
LARP [71](reported by [139])	-		74.72%		-		66.95%		61.37%	
Actionlet ensemble [142]	-		76.60%		-		-		-	
HOPC [133]	-		80.0%		-		-		-	
Res-TCN [124]	-		-		-		83.10%		74.30%	
SkeletonNet [134]	-		-		-		81.16%		75.94%	
SO-Feature [143]	-		-		71.31%		-		-	
Lie Group SE[144]	-		-		75.78%		-		-	
Seq2Img[145]	-		-		83.33%		-		-	
	w/o-cvr	w-cvr	w/o-cvr	w-cvr	w/o-cvr	w-cvr	CV-w/o-cvr	CV-w-cvr	CS-w/o-cvr	CS-w-cvr
PGF1	**89.6%**	89.8%	74.6%	76.7%	93.0%	93.3%	82.5%	83.5%	70.3%	73.2%
PGF2	87.0%	87.0%	85.0%	87.0%	**93.1%**	**93.5%**	84.2%	85.4%	71.9%	74.2%
PGF3	85.9%	89.1%	79.6 %	78.3%	90.0%	91.4%	82.0%	83.2%	71.4%	75.0%
PGF4	88.8%	89.7%	85.0%	86.9%	88.9%	91.1%	**84.3%**	**85.9%**	**76.6%**	**79.1%**
PGF4-M	80.7%	85.0%	86.4%	84.7%	85.9%	86.5%	77.6%	79.6%	67.7%	70.1%
PGF4-E	77.0%	77.8%	85.2 %	87.6%	81.6%	83.5%	63.6%	68.8%	52.8%	59.0%
PGF4+PGF4-M	83.5 %	**89.9%**	**87.3%**	**87.9%**	88.9%	89.5%	83.8%	84.7%	75.1%	77.4%
PGF4+PGF4-E	80.0 %	79.5%	84.9 %	86.1%	86.7%	88.5%	82.9%	83.6%	74.8%	76.9%
PGF4-M+PGF4-E	88.9%	88.1%	83.3 %	86.4%	85.7%	86.6%	78.4%	79.5%	67.6%	69.2%
PGF4+PGF4-M+PGF4-E	85.8 %	83.2%	**87.3%**	86.6%	87.0%	91.0%	82.6%	83.2%	73.7%	75.8%

The proposed approach can achieve more accurate results on the NTU RGB+D dataset by following the two standard evaluation protocols, and the result indicates that the proposed approach can effectively distinguish the actions from large-scale dataset. From the results presented in Table 3.2, we can argue that the skeleton, our proposed motion and energy features can effectively characterize the human actions.

3.4.3.1 Result of UTD-MHAD dataset

For the UTD-MHAD dataset, the cross-subject evaluation protocol is adopted [27], a total of 431 video samples that were performed by the first five actors are used for training and the rest 430 examples of the last five actors are kept for testing. We compared our results with [132], [138], [20] and [135], and our proposed model achieved the highest recognition accuracy at 89.9%, taking the "PGF4+PGF4-M" as the input feature. All of the other feature combinations also achieved excellent results on this dataset. The confusion matrix in Fig.3.12 demonstrates that different features

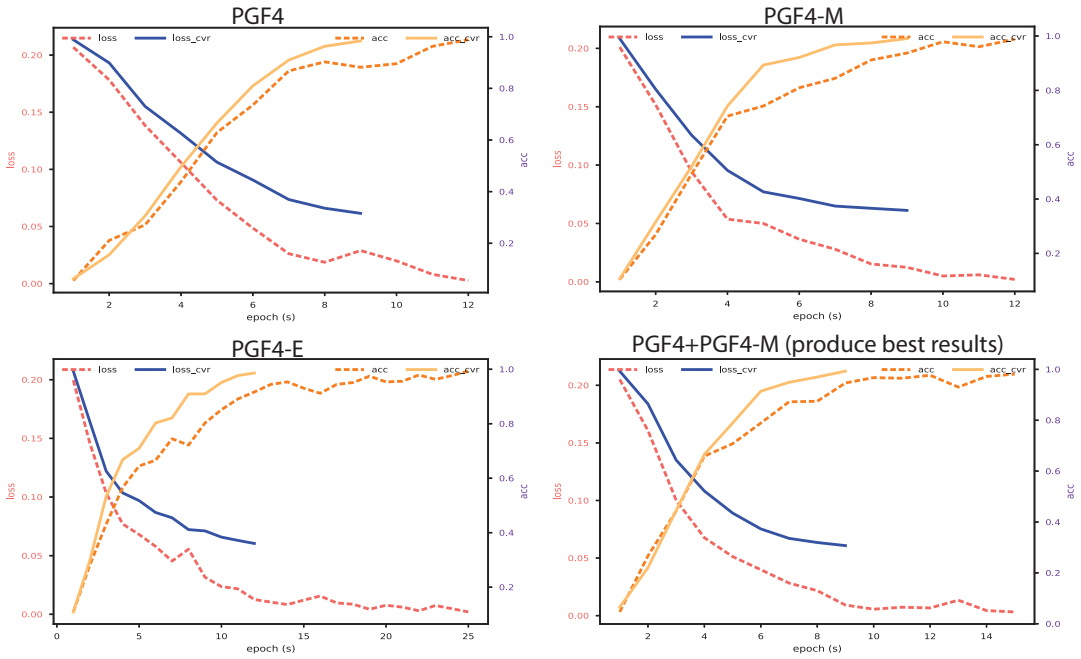


Figure 3.9: Convergence rate curves of training for different input features on UTD-MHAD

have different discriminative abilities for each action that are included in this dataset. From Fig.3.12 , we can observe that the confusion matrix with our proposed regularizer can achieve a better classification performance than its opposite. In order to investigate the effects of the proposed regularizer on the training process of the CNN model, the convergence rate curves and loss curves for several different settings are shown in Fig.3.9.

It is easy to find that the model with our proposed regularizer converges much faster and stops training earlier than the opposite configuration, training without proposed regularizer. It should be noted that the loss of the model with the correctness-vigilant regularizer is larger

than the loss of the model without the correctness-vigilant regularizer, while achieving a better recognition accuracy. This phenomenon proves that the regularizer can help to train a robust model, which can achieve better performance on the testing dataset. The confusion matrix of the best combination features "PGF4+PGF-M" is shown in Fig.3.10, from which we can see that the motion feature has effective discriminative ability. This result again verified the effectiveness of our proposed regularizer.

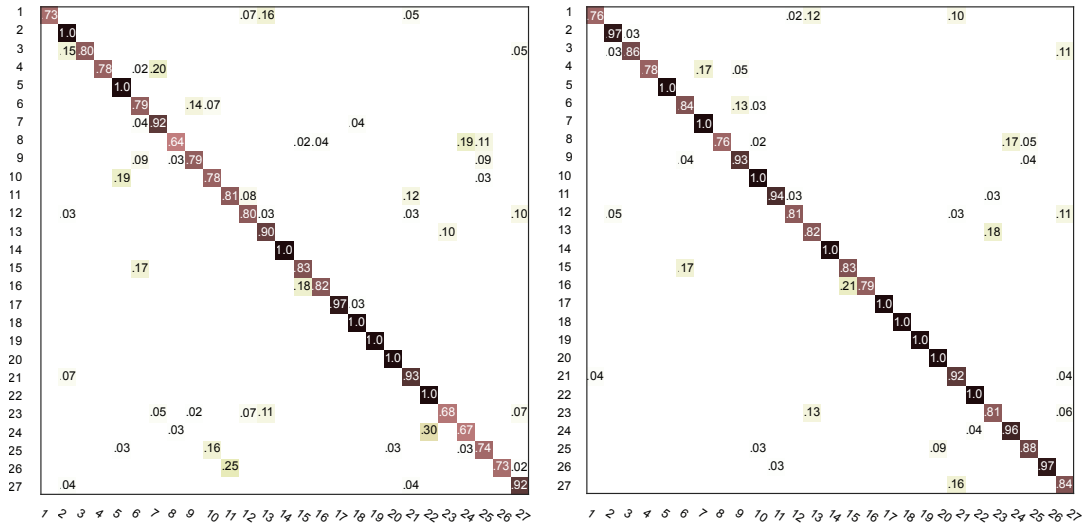


Figure 3.10: Confusion matrix for feature "PGF4+PGF-M" on UTD-MHAD

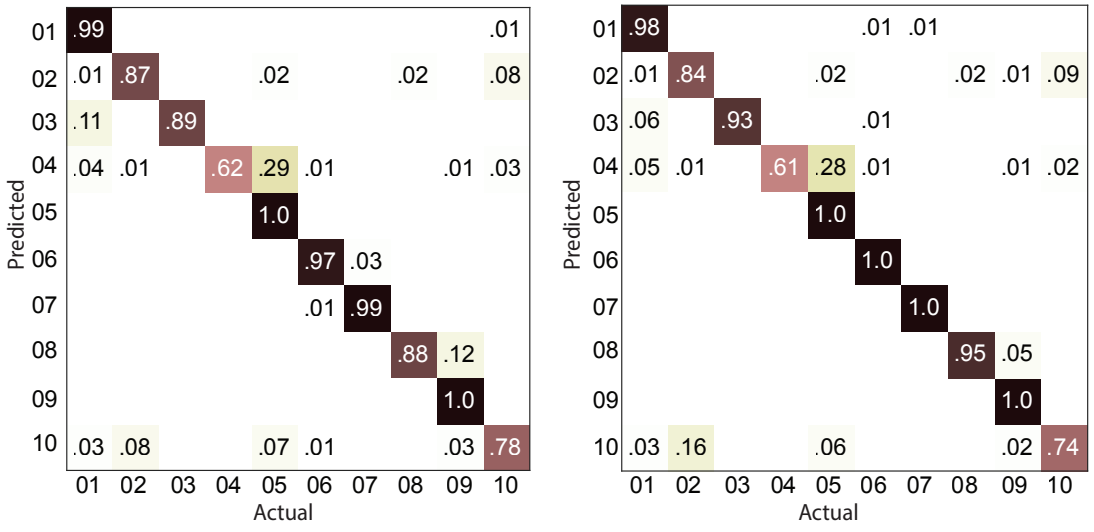


Figure 3.11: Confusion matrix for feature "PGF4+PGF-M" on Northwestern UCLA [128]

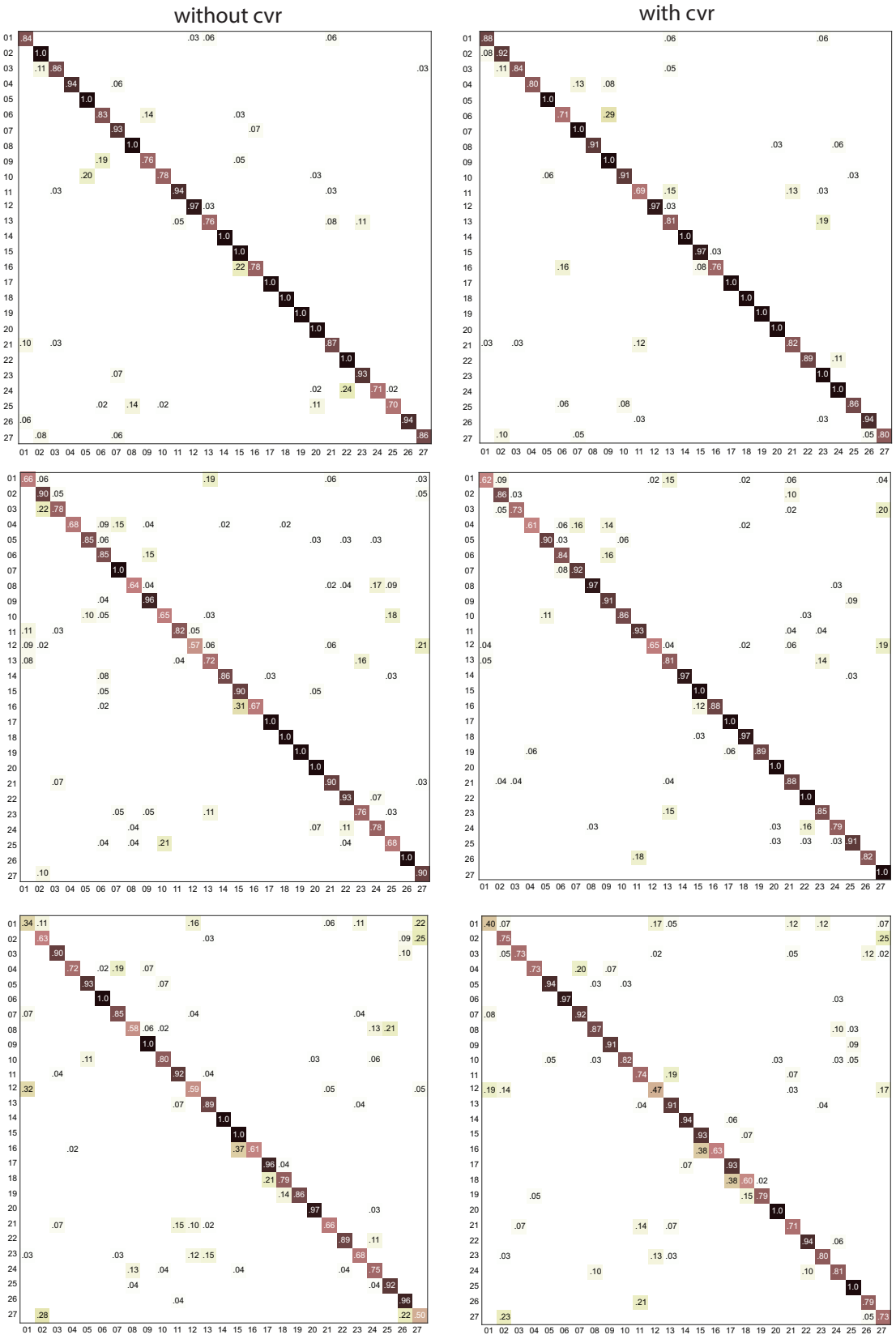


Figure 3.12: Confusion matrix for three different features on UTD-MHAD (Left: without correctness-vigilant regularizer, Right: with correctness-vigilant regularizer)

3.4.3.2 Result of Northwestern UCLA dataset

For the Northwestern UCLA dataset, we adopted the evaluation setup that was proposed by [76], the samples were captured by the third cameras are used as the testing dataset, and the samples of the first two cameras are used for the training dataset. As shown in Table 3.2, the results of our proposed features and regularizer have increased to 87.9%, an increase of 7.9% compared with [133]. The different discriminative ability of our proposed features is more evident on this dataset and the effectiveness of our proposed regularizer is more significant on this dataset. Our proposed approach greatly improves the performance on this dataset.

The confusion matrix of the most effective feature combination "PGF4+PGF-M" is shown in Fig.3.11. As can be seen, most of actions are well distinguished by both test approaches, but the model trained with the proposed regularizer can achieve a better performance than the opposite one. The training and testing accuracy and loss curves for the Northwestern-UCLA dataset are shown in Fig.3.13. As expected, the proposed method can efficiently prevent overfitting problem, achieving better recognition accuracy with higher loss and this fact can also prove that the trained model with our proposed regularizer is robust on the testing dataset.

3.4.3.3 Result of HDM05 dataset

The joint coordinates contained in this dataset are much more accurate than the other datasets that were obtained with the pose estimation algorithms, because this dataset is captured by utilizing the optical motion capture system. Some actions in this database are very similar, which should be categorized into the same category. For example, the "jogging starting from air" and "jogging starting from floor" should be the same class "jogging". We followed the strategy of [146] to recategorize the actions into 65 classes. Following the experimental setup of [145], we split randomly half of the sequences for the training and those that were left for the testing. As shown in Table 3.2, we achieved better results compared with [145] on this dataset.

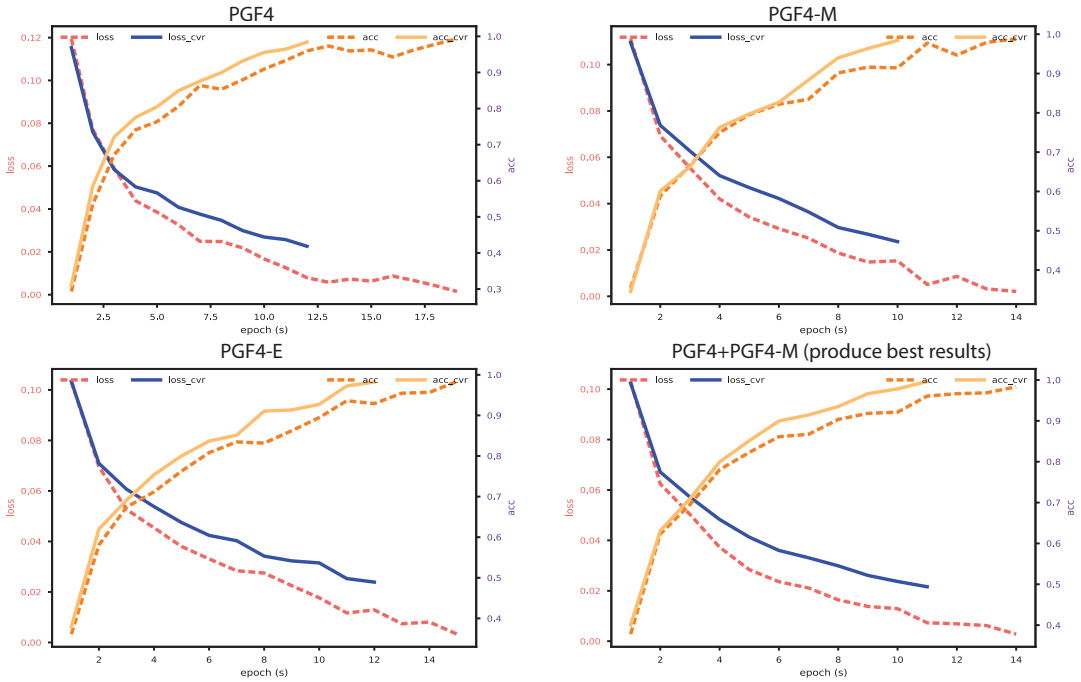


Figure 3.13: Convergence rate curve of training for different features on Northwestern UCLA

3.4.3.4 Result of NTU RGB+D dataset

The NTU RGB+D dataset is currently the most challenging dataset for action recognition. It includes 56880 sequences, among of which 448 sequences are mixed with one or two people appeared in the video. The standard evaluation protocols proposed in [20], the cross-subject (CS) evaluation and the cross-view (CV) evaluation, are adopted in our experiments. As shown in Table 3.2, our model can achieve competitive results on the cross-view experimental protocol, which indicates that our proposed primitive geometric relational features are suitable for modelling the temporal dynamics of actions captured in different viewpoints for this challenging dataset. The convergence curve for both cross-view evaluation strategy and cross-subject evaluation strategy is shown in Fig.3.14. Again, the regularizer performs consistently well to speed up the training process and improve the generalizability of the trained model. The confusion matrix for the cross-subject evaluation with our proposed model is shown in Fig.3.15.

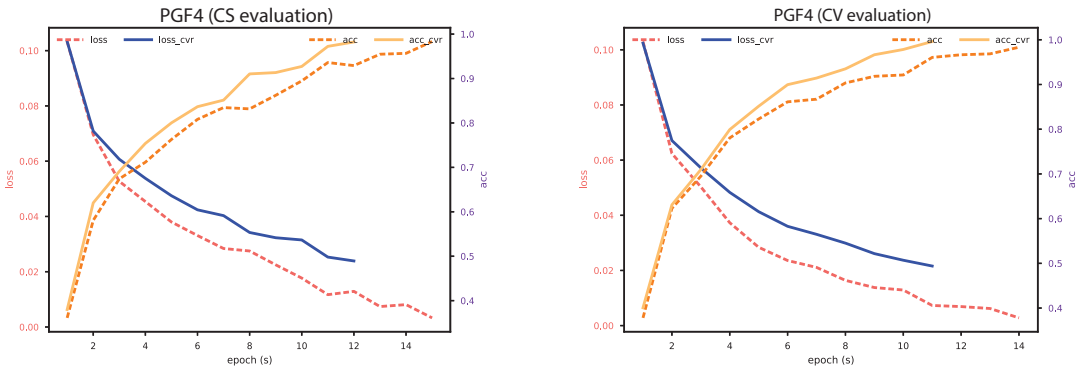


Figure 3.14: Convergence rate curves of training for best results on NTU RGB+D

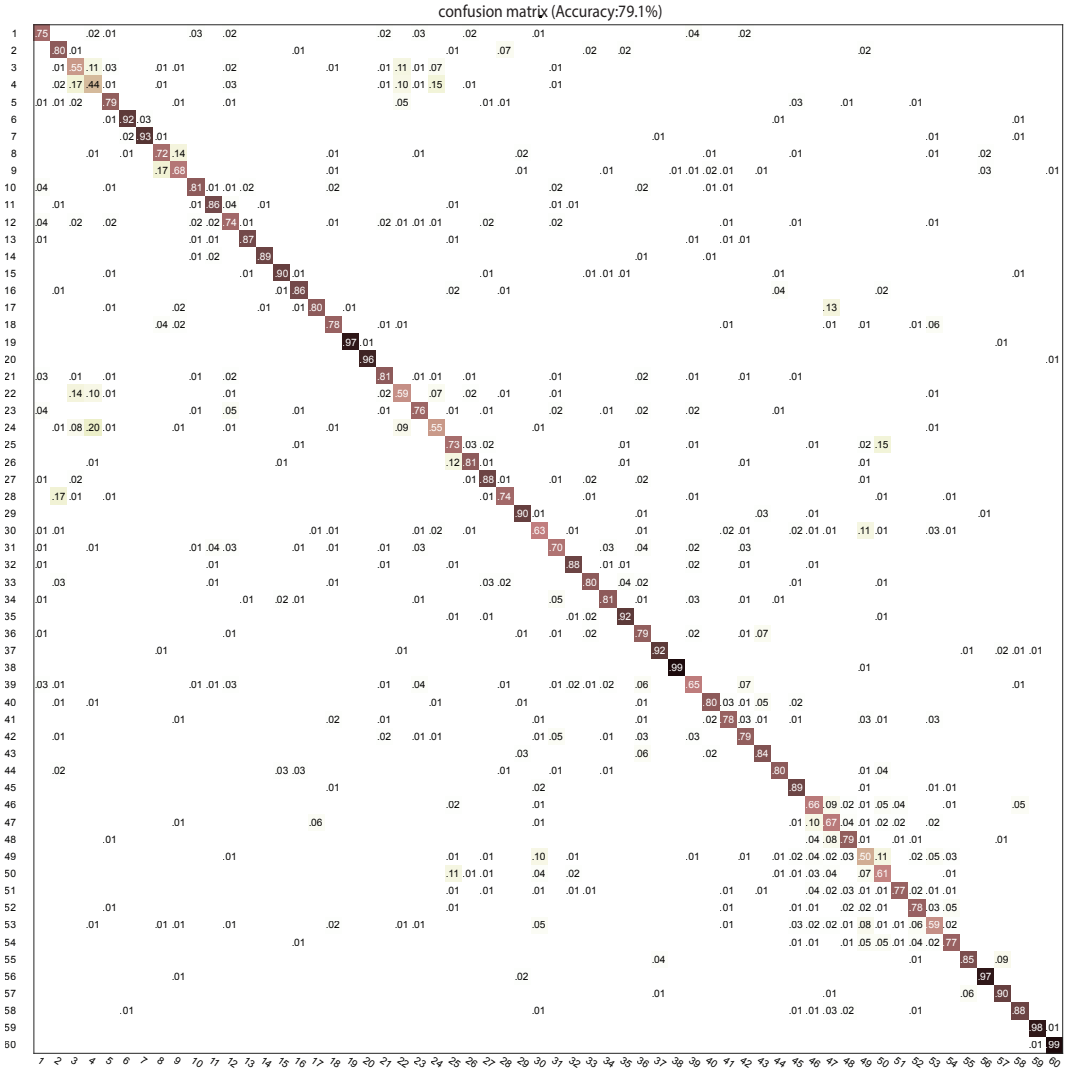


Figure 3.15: Confusion matrix for best result on NTU RGB+D

3.5 Summary and contributions

In this Chapter, we verified the effectiveness of our proposed framework and features for human action recognition. In summary, we can identify three major contributions as follows:
1) We introduced a novel temporal kinematic image representation for the skeleton-based motion and energy features. This approach can effectively embed a rich set of information consisting of the joints' coordinates and our proposed primitive geometric relational features into one static image to characterize the human actions.
2) The resulting classifier trained with our proposed correctness-vigilant regularizer performs consistently well on several benchmarking datasets. The proposed regularizer helps to speed up the training process and improve the generalizability of the trained model on the testing dataset.
3) We demonstrated that the proposed motion and energy features can also characterize the human actions, which works very well on small datasets. The performance of the motion and energy features will degrade on large datasets.

Chapter 4

Spatio-temporal Kernel based Temporal Convolutional Network for Human Action Recognition

In the previous Chapter, we converted the skeleton sequence classification problem into an image classification problem, which is not the most natural way for sequential signal processing since it cannot model the temporal dynamics very well in the temporal domain. Also, except for the previous introduced geometric features in Chapter 3, we extended the research to design several novel interpretable geometric features based on the skeleton sequence data. The proposed features are instinctively a sequence of signals, and the RNN model was created for modelling the sequential dynamics. Therefore, this Chapter explores the performance of our proposed spatio-temporal kernel-based temporal convolutional networks with the extended geometric relational features as input.

4.1 Motivation

Even though action recognition is an extensively researched topic, it is still a challenging problem in realistic applications. In order to adapt to the challenges posed by real-world scenarios, an ideal human action recognition system should be able to generalize well despite the presence of large variations within a class of action and it should be able to uniquely distinguish the actions that belong to different classes. For a multitude of action classes to consider, the classifier will find it more difficult to uniquely identify classes from one to another, since it is more likely to find highly indistinguishable classes. Aiming at addressing

this problem, we extend the idea of Chapter 3 and propose some geometric features for action analysis from the skeleton sequence, and then an advanced recurrent framework is utilized to extract the discriminative features from the raw input.

Inspired by the excellent ability of the CNN model in extracting the spatial features from the input image with multiple convolutional kernels, and the ability of the RNN model in dealing with the variation in the length of the sequential data, the proposed framework in this Chapter incorporates a spatio-temporal kernel into the convolution operation, to learn the dynamics of the motion in the whole sequence and to learn the dependency relations in a relatively local range of features, as well as the interdependency relations between the joints within one frame.

Most of the traditional approaches for skeleton-based action recognition utilize descriptors that are carefully handcrafted [22], [147] to represent temporal dependencies. However, the drawback of using these existing engineered features is that they are inadequate for use in complex scenes due to their limited discriminability [148]. Moreover, it is inevitable that measurements coming from skeleton data acquisition systems are usually plagued with noise and inaccuracies. As a result, relying on the acquired noise-infested skeleton data without de-noising first, and without advanced learning mechanisms to extract temporal geometric and kinematic features, it is hard to produce accurate action recognition classifiers. The pose variations that were presented during a person perform a specific action can be described by the variations of some geometric features, such as angles between adjacent limbs. An advanced neural network should be designed to extract these features from the raw joint coordinates. However, there is no such kind of neural network model available for the skeleton-based action recognition. In order to utilize the representation ability of these geometrical features for skeleton-based action recognition and take advantage of the learning ability of neural networks, we proposed to employ several geometric features as input for the proposed SKB-TCN model. The explicit geometric features for the SKB-TCN model is promising for improving the recognition accuracy and reducing the pressure for representative features extraction in the training process. This Chapter investigates the performance of a novel recurrent neural network with explicit geometric features as inputs for skeleton-based action recognition.

4.2 Related work

It is evident in the literature that activity recognition with the aid of multiple sensing devices strategically fixed in the environment to capture multiple features simultaneously is one promising research direction, yet is still lacking one ideal model that can mine the relationship between a consecutive frames of signals in the spatial and temporal domain simultaneously. In this section, we provide an extensive review on the different geometric features and signal encoding methods that are suitable for the skeleton-based action recognition, used by recently developed algorithms. Next, we introduce the LSTM-RNN based approaches, which are closely related to the proposed approach. Lastly, the attention mechanism techniques used by DNN based technologies are reviewed, because our model can also be treated as an attention-based model.

4.2.1 Geometric relational features based on 3D skeleton

While undergoing a physical activity, the skeletal and muscular systems should work together, producing movements. The movements of human body can be simply represented as the movement of a sequence of skeletal poses, formed by a hierarchy of interlinked joints and bones, positioned at different angles, possessing distinct kinematic energies. Previous skeleton-based approaches for action recognition proposed various methods for extracting discriminative information from the raw skeleton data. These approaches are specifically detailed and reviewed in Chapter 2. In this chapter, we will review the skeleton based features from a more high-level perspective. The skeleton-based feature extraction methods can generally be categorized into three groups, which includes the joint-based features, the mined joint-based features and the dynamics-based features.

1) Joint-based features aim to capture the relative body joint locations. For example, [111] concatenated together all the joints coordinates, and then transformed them into one static image, holistically representing one complete action sequence. In [149], after examining the features using a boosting algorithm, it was inferred that the pairing of the features between the current frame and historical frames is useful for subsequent action recognition. In Muller's work [150], various statistical features based on geometric features were introduced for efficient motion captured data retrieval. All of these features were all carefully

predefined, considering the relative position, angle and speed of the pre-selected body parts (e.g., torso, hand, elbow, legs, foot, and so forth.). However, they did not consider adding features that detail the geometric relationship of joints within one frame or between different frames explicitly.

2) The mined joint-based features rank the different joint-related features, e.g., the mean or variance of the joint angles, and so forth. to capture the most discriminative body parts for different actions. For instance, [113] examines a temporal window to generate an arrangement of important joints. They claim that the features are highly intuitive and interpretable. Seidenari et al. [151] breaks down the original 3D skeleton sequence into kinematic chains, and then map the joint positions into a locally defined reference system. In turn, they utilized an approach called the multi-part bag-of-poses, which paves the way for the alignment of the different limbs using a variant of the nearest-neighbor algorithm.

3) Lastly, the dynamics-based features aim at modeling the evolution process of the dynamic features, which are observable from either a subset of joints, or from the entire set of joints. For example, [152] utilized the Grassmanian manifold learning approach to map the 3D skeleton-based features into its subspaces so as to improve the performance of the action recognition system based on the extracted features in the subspaces.

In summary, even though we can mine rich information from the three kinds of descriptors mentioned above, most of them cannot describe the local variation of the skeleton sequence very well, so they cannot derive very good representation globally for the whole sequence. They either rely heavily on the learning model to learn the representative features or neglect the native feature of the skeleton sequence, and rarely consider the local range features of actions. In this present Chapter, extending the idea of Chapter 3, we utilize a set of highly intuitive geometric relational features to work together with a novel spatio-temporal kernel to learn geometric and kinematic representations for recognizing actions. This is motivated by the feature extraction process of speech processing technique, which extracts a feature vector from a window-sized frames. The proposed spatio-temporal kernel can undertake this work efficiently.

4.2.2 Local signal encoding strategies for action recognition

Contextual information modelling for different activities is the most critical component for activity analysis. Before feeding the extracted features into a learning model for further analysis, the raw signals should be processed properly into the type of the models' input. For skeleton-based action recognition, all the instances in the segmented action recognition dataset usually provides a skeleton sequence and a corresponding label. The target for action classification is to classify the provided video or skeleton sequence into its correct label. In general, there are two popular approaches to deal with this problem in the research community: 1) encode the variation of actions into a static image; 2) treat it as a time-dependency signal. The first approach convert the skeleton pose variation in one video sequence into a static image, which is introduced in Chapter 3, using the projection of coordinates on three orthogonal planes for action classification. The second approach instinctively treats the skeleton sequence or its derived sequence as a time-series signal, which contains the variation of joint coordinates or other relational attributes, such as our proposed various geometric features, and their derived motion and energy features.

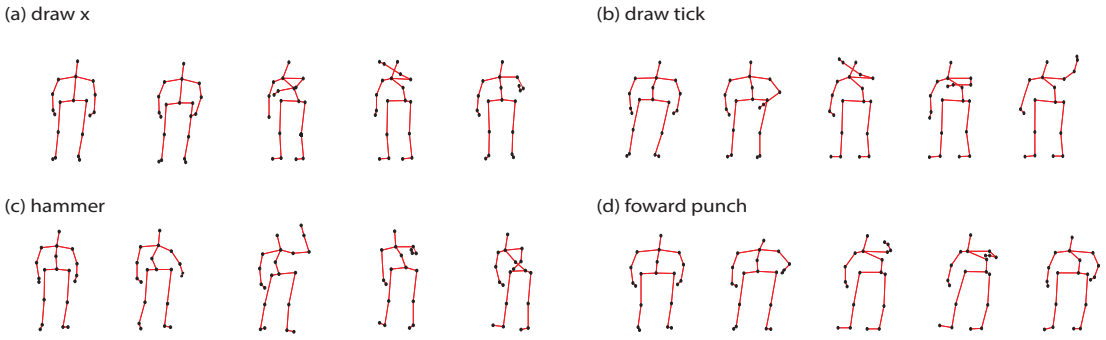


Figure 4.1: Actions instances with similar variation of skeleton sequences

For both cases, in the skeleton data preprocessing, the normalization process typically involves converting all of the data sequences so that they will all be of the same length for batch processing. To this end, some sequences will be shrunk while others will be expanded to fit the normalized length. In other words, the length of the sequence should perfectly align with the exact number of steps that the RNN model will be executing for. In the case of a sequence expansion, the original sequential data is typically padded with repeating frames so as to increase the sequence up to a target length. However, this introduces temporal

information loss, as the gap between consecutive frames in the original sequence widens. With regards to the case of sequence compression, it also brings about a similar problem, that is, some of the patterns are lost due to frame deletions. The loss of patterns inevitably causes a loss of temporal information contained in the original sequence. For example, sequences of four similar actions are shown in Fig.4.1, where it shows that the first two frames are almost the same, and therefore we can treat these three frames as one frame; however the traditional recurrent neural network cannot do this explicitly. So, we proposed our spatio-temporal kernel based temporal convolutional network to address this problem. The arrangement of features corresponding to the joints is also critical to the performance of the classifier. Previous works have proposed the arrangement of joints according to groups of body parts [111] as well as arrangement of joints according to some predefined body part traversal [153] that were proven to be effective for some datasets. However, so far, there is no universally known and proven arrangement of joints that works well with any of the datasets. Moreover, in principle, different actions will benefit from the involvement of the different arrangements of joints and correlations between them.

In light of these problems, our proposed model contains a spatio-temporal kernel which effectively serves as a pooling layer. According to the window size that is assigned to the model, the proposed kernel takes multiple frames as input, and then outputs the extracted features which are considerably smaller in size, filtering out all the unnecessary inputs, while exhibiting the important spatial relationships between joints in each frame. On the other hand, in the case of sequence contraction it is inevitable that to some extent, the original skeleton data are altered in a way that noise is introduced, usually polluted to some extent, therefore it is essential for the network to disambiguate them. With our proposed geometric and kinematic features, we can preserve the core information detailing the actions, such as the naturalness and smoothness of the movements, which can ensure that the learning process will be consistent for all samples.

RNN-based approaches have shown to be dominant when applied to skeleton-based action recognition. However, the existing methods typically feed on the coordinates of the joints as inputs, without any geometric or kinematic clue. An efficient approach for improving the accuracy of these systems is to allow the system to operate on features that are extracted from both the spatial and temporal domains. Even though the RNN models are good at finding the

temporal relationships between the features among the multiple frames, they neglect the spatial relations in each individual frame. In this Chapter, we specifically selected a set of simple geometric relational features for action recognition. While previous skeleton-based models considered a priori arrangements of limbs to extract meaningful features, in this study, we used of a spatio-temporal kernel to discover the rich set of relationships between the joints and limbs; thus addressing the weaknesses of the existing approach.

4.2.3 Skeleton-based action recognition

In recent years, the 3D skeleton-based action recognition has become one of the most popular approaches for HAR, because of the advancement of the skeleton estimation algorithms. Some attempts have applied the RNN model into action recognition successfully and the RNN-based models have shown great advantages in capturing the temporal dynamics in sequential skeletons. Recently, Du et al. [111] and Zhu et al. [154] successfully adapted the RNN model for the task of action recognition. Du et al. [111] are the first researchers who devised a hierarchical RNN model for skeleton-based action recognition, in which each frame of the skeleton data is divided into five parts according to the human body structure and then each part is fed into five independent recurrent neural networks. The outputs of the recurrent units in the five RNN models are then concatenated together in the next succeeding higher recurrent layer. The resulting concatenated features are then used as the input for the next recurrent layer, until all of the features are fused together taking into account all of the body parts. Lastly, the computed features are then fed into a classifier layer (softmax). In contrast, Zhu [154] proposed a fully connected deep LSTM network to discover the co-occurrence movements for human body parts. This work used the hidden states of the recurrent network as the extracted features, which relies heavily relies on the long-term dependency learning ability of the recurrent neural networks. Li et al. [155] proposed a spatial and temporal attention model to assign different weights to the joints within each frame and selectively extract discriminative features. A view adaptive model was proposed by Zhang et al. [156] to process the input skeleton to adapt to the a proper observation viewpoints. The TS-LSTM model was proposed by [157] to capture the short-term, medium-term and the long-term temporal dependencies as well as the spatial dependency.

Although RNNs are widely used for predicting and classifying sequential data, which are

rarely used for extracting features from a window's worth of time-series data and utilizing the hidden states or the outputs of the LSTM-RNN as discriminating features to be used as input vectors to another layer of RNN. In this Chapter, we propose a novel strategy for mitigating the reliance of action recognition systems on the RNN's long-term dependency learning ability.

4.3 The algorithms

4.3.1 Extension of geometric relational features

With the general conception in mind that was proposed in Chapter 3, as shown in graph (b) to graph (h) of Fig.4.2, we attempt to extract more interpretable geometric relational features in the following sections, such as the distance between different joints, joint-joint orientation, the angles between different joint-joint lines, the angles between joint-joint lines with planes, the angles between planes, the distance between joint and joint-joint lines, the distance between joint and planes. Following the steps introduced in Chapter 3, all of these geometric features can derive their corresponding motion and energy representation. For example, $\vec{\Delta}M\_Angle$ is the variation of our predefined angles between consecutive frames. $\vec{\Delta}M\_Distance$ indicates the variation of our predefined distance between the joints in a single frame. While $\vec{\Delta}E\_Angle$ represents the representative energy carried when the performer finished the variation of a predefined angle from the previous pose to the current pose. $\vec{\Delta}E\_Distance$ indicates the representative energy carried once the actor has finished the variation of our pre-defined distance between the joints from the previous frame and the current frame.

Joint-joint displacement: The joint coordinates change dynamically when human perform different actions, shown as graph (b) of Fig.4.2, the orientation of joint pairs changes also. As stated before, the j_{th} joint coordinate of the f_{th} frame is represented as $(J^f_{j,x}, J^f_{j,y}, J^f_{j,z})$, and the k_{th} joint coordinate of the f_{th} frame is represented as $(J^f_{k,x}, J^f_{k,y}, J^f_{k,z})$. The joint-joint orientation can be calculated as the following equation:

$$\phi^f_{j,k} = (J^f_{j,x} - J^{f-1}_{k,x}, J^f_{j,y} - J^{f-1}_{k,y}, J^f_{j,z} - J^{f-1}_{k,z}) \tag{4.1}$$

Angles between different joint-joint lines: Commonly, the angle between the human body

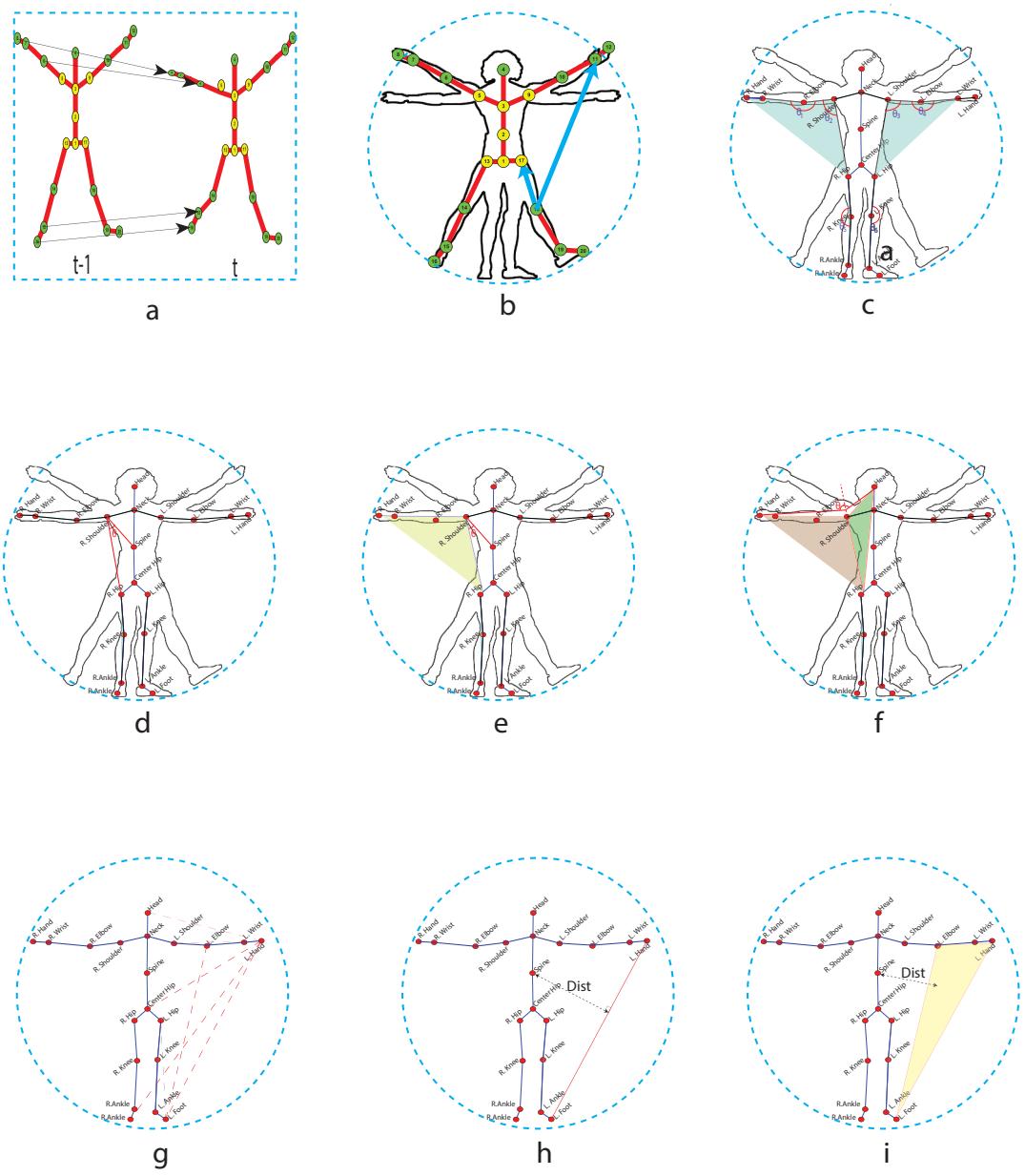


Figure 4.2: Extended geometric relational features. a) motion features derived from joint coordinates; b) Distance between two joints; c) Angle between adjacent limbs; d) Angle between joint-joint-lines; e) Angle between joint-joint-line and plane; f) Angles between plane and plane; g) Distance between two joints; h) Distance between joint and joint-joint-line; i) Distance between joint and plane

segments, shown as graph (c) and graph (d) of Fig.4.2, is used to describe the human pose. The variation of the angles displays different patterns for different actions. For example, when a human brushes his teeth, the angle of the upper part of the human body changes more frequently relative to the lower part. On the other hand, when a human runs or walks,

the angles of the lower part change more frequently. The angles between the segments can be calculated as the following formula:

$$\theta_{n,f} = \cos^{-1}\left\{\frac{\alpha \cdot \beta}{\left|\alpha\right|\left|\beta\right|}\right\} \tag{4.2}$$

where $\theta_{n,f}$ represents the n_{th} angle that we pre-defined for the f_{th}frame. Operator $(\cdot)$ is the inner product of the direction vector. Operator $|\ |$ denotes the magnitude of a vector. α and β are the joint-to-joint vector representations.

Angles between joint-joint lines and planes: In order to represent the relative position between the human body parts with our predefined planes, we propose using the angle between the joint-joint lines and the predefined planes to represent the relative position of the two joints and another three joints, which is shown in the graph (d) of Fig.3.3. For example, the angle between the joint-joint-line $J_j^f \rightarrow J_k^f$ and the plane $J_a^f \rightarrow J_b^f \rightarrow J_c^f$ can be calculated by the Equations 4.3 and 4.4. Given three joints J_a^f, J_b^f and J_c^f, the normal vector of the plane defined by these three joints can be calculated with following Equation:

$$\vec{n}_{a,b,c} = (J_a^f \rightarrow J_b^f) \odot (J_b^f \rightarrow J_c^f) \tag{4.3}$$

The angle between the line $J_j^f \rightarrow J_k^f$ with the plane $J_a^f \rightarrow J_b^f \rightarrow J_c^f$ can be represented by the angle between the line and the normal vector of the plane as follows:

$$\theta_{(j,k)\rightarrow(a,b,c)} = \cos^{-1}\left\{\frac{\alpha \cdot \beta}{\left|\alpha\right|\left|\beta\right|}\right\} \tag{4.4}$$

where α is the orientation vector of $J_j^f \rightarrow J_k^f$, β is the normal vector calculated by formula 4.3. Some databases provide joint angles in the motion capture files, whereas, some database do not provide this feature, so we proposed to compute this angle based on the raw joint coordinates.

Angles between predefined planes: The angles between our predefined planes, shown in the graph (e) of Fig.3.3, can also represent the human pose, the angles between the planes can describe the relationship between a group of six joint positions. The angle between two planes can be represented by the angle between the normal vectors of two planes. For example, the

angle between the plane $(J_a^f \rightarrow J_b^f \rightarrow J_c^f)$ and the plane $(J_i^f \rightarrow J_j^f \rightarrow J_k^f)$ can be calculated by formula 4.3 and formula 4.2 also.

Distance between different joints: The variation of the distance between the joints can also reflect the characteristics of different actions. As shown in the graph (f) of Fig.3.3, we selected a set of important joints, and applied the distance between them as the input features. The distance between each pair of key joints can be calculated by the following formula:

$$D_{i,j,k} = \left| P_{i,t} - P_{h,t} \right| \tag{4.5}$$

Distance between joint and joint-joint lines: The variation of the distance between the joints and body limbs can also reflect the characteristics of the different actions. As shown in the graph (g) of Fig.3.3, we selected a set of important pair of joints and joint-joint lines, and applied the distance between them as the input features. The distance between $(J_{i,x}^f, J_{i,y}^f, J_{i,z}^f)$ and line $(J_{j,x}^f, J_{j,y}^f, J_{j,z}^f) \rightarrow (J_{k,x}^f, J_{k,y}^f, J_{k,z}^f)$ can be represented as:

$$D_{i->j,k}^f = \frac{2S_{\triangle i,j,k}^f}{\left| J_i^f - J_k^f \right|} \tag{4.6}$$

where $S_{\triangle i,j,k}^f$ is calculated by using Herons Formula.

Distance between joint and planes: In order to represent the pose of a human while an actor performs specific actions, the relationship between one joint and a group joints is important in terms of characterising the pose of one human at a specific timestep. As shown in the graph (h) of Fig.3.3, we devise the concept of the distance between the joint and the predefined planes, which can represent the relative position between one joint and the other three joints, consisting of one plane in the space. The distance between the joint J_i^f and the plane $J_a^f \rightarrow J_b^f \rightarrow J_c^f$ can be calculated by using the normal vector of the plane, which can be obtained with the Equation 4.7, and the joint coordinate J_i^f.

$$\vec{n}_{a,b,c}^f = (J_a^f \rightarrow J_b^f) \odot (J_b^f \rightarrow J_c^f) \tag{4.7}$$

The angle between the line and the plane can be represented by the angle between the line and

the normal vector as follows:

$$Distance_{i\to(a,b,c)} = J_i^f \odot \vec{n}_{a,b,c}^f \tag{4.8}$$

In order to simplify the representation of our predefined features, we use the symbols that are listed in Table 4.1 to represent the features that are illustrated in Fig.4.2. The details of these symbols are also described in Table 4.1. For all of the derived features based on the above basic features, we use SOF_#N_M, $N = 1, ..., 8$ to indicate the motion features of the corresponding basic features, and represent the energy features of the corresponding basic features as SOF_#N_E, $N = 1, ..., 8$.

Table 4.1: Representation of proposed geometric relational features

#	Description
SOF_#01	Joint coordinate $(J_{k,x}^f, J_{k,y}^f, J_{k,z}^f)$
SOF_#02	Distance between two joints, Equation-4.5
SOF_#03	Orientation from joint one to joint two, Equation-4.1
SOF_#04	Distance between joint and joint-joint-line, Equation-4.6
SOF_#05	Distance between joint and plane, Equation-4.8
SOF_#06	Angle between joint-joint-lines, Equation-4.2
SOF_#07	Angle between joint-joint-line and plane, Equation-4.4
SOF_#08	Angles between plane and plane, Equation-4.2

4.3.2 General architecture

An ideal human action recognition system favors feature representation that is invariant to different types of human physiques and anthropometric differences between individuals. The workflow of the proposed framework in this chapter is shown in Fig.4.3. With the preprocess techniques that were proposed in the previous chapter, we transform all skeleton data into one universal system, and the origin is located at the hip of the human body, and this reduces the effect of different human body size for the extraction of geometric features.

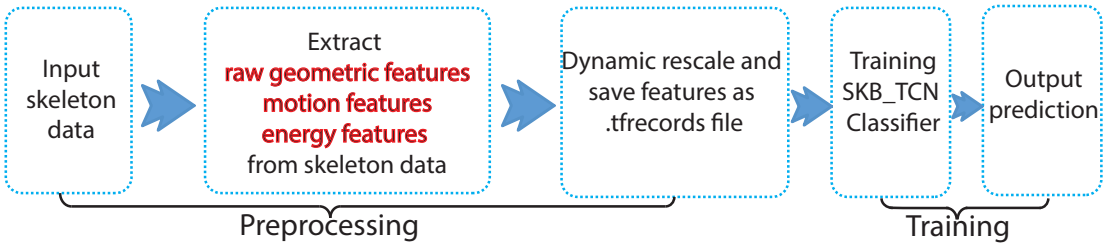


Figure 4.3: The workflow of the proposed SKB-TCN framework

The pseudocode of the preprocess phase in the above workflow graph is shown in the following code block.

```
# num_instance is the total number of instance that are included in the
    database
for (i = 0; i < num_instance; i++) {
   # ske and size are two input parameters
   # ske is skeleton data of instance i, size is the target size.
   ske, size:

   # generate the index of selected frames randomly
   index = rescale(len(ske), size)

   # extract the selected frames from the ske sequence
   ske_scaled = ske[index]
   # return the dynamically generated normalized sequence
   return ske_scaled
}
```

Code block 3: *Normalize the skeleton sequence into same length*

The pseudocode of the proposed SKB-TCN model is shown in following code block:

```
# Variables
# given a window of input sequences, window-size is 4
input_feature

X_t-2, X_t-1, X_t, X_t+1 = input_features

# modify the output of the LSTM model with proposed  algorithm
for i in range(4):
        o_t_i, c_t_i = LSTM(input_feature[i])

# feature fusion, mean can be replaced with other functions
o_t = mean(o_t_1, o_t_2, o_t_3, o_t_4)
```

Code block 4: *The pseudocode for one SKB-TCN unit*

After we have obtained the normalized sequence with the above dynamic rescale algorithm, we can then extract the predefined geometric relational features and train the SKB-TCN model.

```
Train_path, Test_path # Preprocessing, extract features for each sequence
joint_pairs, lines, planes # predefined variables
ske # given one skeleton data from training or testing dataset
# SOF1, ..., SOF8: reffer to Table 4.2, M and E indicate Motion and Energy
for i in range(len(ske):
   SOF1 = pos(i).reshape(T,-1);  # joint coordinates
   SOF1_M = pos[i+1]-pos[i]
   SOF1_E = sum(square(SOF1_M), axix=2)
   SOF2 = [pos[i][j]-pos[i][k] for j, k in joint_pairs] # Equation 4.1
   SOF2_M = SOF2[i+1]- SOF2[i]
   SOF2_E = sum(square(SOF2_M), axix=2)
   SOF3 = Lineline_angle(pos[i], lines)  # Equation 4.2, SOF3
   SOF3_M = SOF3 [i+1] - SOF3[i]
   SOF3_E = square(SOF3_M)
   SOF4 = line_pl_angle(SOF2[i], planes)  # Equation 4.3, SOF4
   SOF4_M = SOF4[i+1] - SOF4[i]
   SOF4_E = square(SOF4_M)
   SOF5 = pdist(pos[i])  # joint-joint distance, Equation 4.5, SOF5
   SOF5_M = SOF5(i+1)-SOF5(i)
   SOF5_E = square(SOF5_M)
   SOF6 = joint_line_dist(pos[i], lines)  # Equation 4.6, SOF6
   SOF6_M = SOF6[i+1] - SOF6[i]
   SOF6_E = square(SOF6_M)
   SOF7 = joint_planes_dist(SOF6[i], planes) # Equation 4.7, SOF7
   SOF7_M = joint_planes_dist(i+1)- joint_planes_dist(i+1)
   SOF7_E = square(SOF7_M)
   SOF8 = planes_angle(SOF2[i], planes) #Equation 4.2, 4.3, SOF8
   SOF8_M = SOF8[i+1] - SOF8[i]
   SOF8_E = square(SOF8_M)
   save(Train_path, Test_path) # save extracted feature from skeleton
}
# Training SKB-TCN model
model = SKB_TCN(input, w, s, depth, num_layers, rnn_type)
for i range(epoch):
   label, input = loading_train_batch(Train_path)
   outputs = model(input)
   outputs_prob = Dense(num_classes)(outputs[-1])
   predict = argmax(outputs_prob)
   # Calculate metrics for the model
   loss = softmax_cross_entrypy_with_logits(label, predict)
   accuracy = reduce_mean(correct_pred equal lable)
   # Update parameters
   loss.backward()
   if i % 5 = 0:
      label, input = Load_testing_batch(batch_size, Test_path)
      accuracy = test(input)
```

Code block 5: *Extract predefined features for N samples and train the SKB-TCN model*

The pseudocode of the feature extraction and training process is shown in code block 5. The hyperparameters that can be tuned in the proposed framework are listed in Table 4.2.

Table 4.2: Hyperparameters tuned in our proposed model

parameter	Description	Default value
w	window size of spatio-temporal kernel	3
s	stride of spatio-temporal kernel	1
type	the type of recurrent cell, such as LSTM, GRU, RNN	LSTM
nb_layers	number of layers of spatio-temporal kernel	1
nb_hidden	number of hidden nodes of spatio-temporal kernel	128
lr	learning rate	0.001
length	length of the normalized sequence	30
epoch	epoches of training	50
batch_size	batch size of training	64
input	input features, which are listed in Table 4.1	

4.3.3 Pipeline for action recognition

As a data-driven method for learning long-term sequential information, RNN has recently gained attraction in the skeleton-based action recognition research community. The Vanilla RNN model was proposed for learning the time-dependent temporal relationships between the structured inputs. They take in the sequential data $(x_1, ..., x_t)$, then recursively generates a sequence of hidden states $(h_1, ..., h_t)$ and a sequence of outputs $(y_1, ..., y_t)$ in the following way:

$$h_t = \sigma(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{4.9}$$

$$y_t = W_{hy}h_t + b_y \tag{4.10}$$

where σ is an activation function, e.g., the logistic sigmoid function, which squash the value of a vector into 0 and 1, W_{xh} is the weight matrices between the input and the hidden layer, W_{hh} is the weight matrices between the previous step and the current step of the hidden layer, W_{hy} is the weight matrices between the hidden layer and the output layer, b_h, b_y are the biases for the hidden and the output layer, respectively.

The LSTM network [18] is an upgraded version of the RNN model, and the novel concept of the gating mechanisms is integrated into the LSTM to control the information flow in the hidden layer of the recurrent network. The architecture of one LSTM units is demonstrated in Fig.4.4. One LSTM units includes an input activation function, one memory cell and three

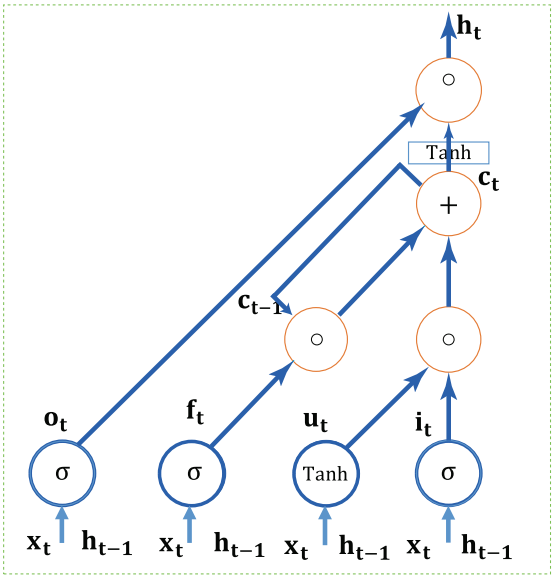


Figure 4.4: Information flow of one LSTM node

gates. The three gates are named as input i_t, forget f_t, and output o_t respectively. The input gate i_t controls how to update the memory cell. Aiming at avoiding the gradient vanishing and error bellowing up problems during the training process, the forget gate f_t was introduced, which decides what is to be forgotten and remembered by the memory cell. The output gate o_t enables the memory

cell's state so it can control the final output state of the LSTM unit. These upgrades all together enable the LSTM model to learn long-range temporal dependencies in the time-series problem and capture the extremely complex patterns that exist in the input sequences. The information flow in one LSTM unit is controlled by the following formula:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\ c_t &= f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\ h_t &= o_t \tanh(c_t) \end{aligned} \tag{4.11}$$

As illustrated in Fig.4.4 and Equation 4.11, the output vector of three gates are indicated as

i, f and o respectively. The state of the memory cell and the hidden nodes are represented as c and h. σ represent the logistic sigmoid function. Furthermore, in order to enhance the feature inference capability and achieve performance improvement for the LSTM model, [158] proposed the Bidirectional LSTM (BiLSTM), which processes the input sequences from past to future and in a reverse direction simultaneously. This enhanced model consists of two LSTM layers, which generate the output vectors $\overrightarrow{h}, \overrightarrow{c}$ and $\overleftarrow{h}$, $\overleftarrow{c}$ from two opposing directions respectively. Subsequently, the outputs $\overrightarrow{h}$, $\overleftarrow{h}$ are combined together to calculate the final output sequence for the BiLSTM layer.

$$y_t = W_{hy}\overrightarrow{h}_t + W_{hy}\overleftarrow{h}_t + b_y$$

4.3.4 Local temporal contextual feature extraction

The 3D skeleton sequences are composed of structured joints and data segments, where action-specific identification patterns only appear in a short period of time. Taking as an example, an action sequence with 10 frames depicting an action, such as "waving hands". We can suppose that the sequences can be decomposed into frames where from first to the 5th corresponds to a "waving left arm", while from the 6th to the 10th frame corresponds to a "waving right arm", we can segment the two actions and treat them separately as two different "local action features". In this work, our system extract representative motion and energy features based on the raw 3D joints coordinates, and then feeds them into a Spatio-temporal Kernel Based Temporal Convolutional Network (SKB-TCN) model, which is shown as Fig.4.5. As can be viewed in Fig.4.5, this architecture can be viewed from the bottom-up, as comprising of a hierarchy of layers, e.g., the input representation layer, the temporal convolutional layer, the extracted features from the output layer (pooling layer). Evidently, it is possible to construct multiple layers, repeating the same hierarchy, where the extracted features from the previous layer serve as the input layer to another temporal convolutional layer, followed by another output layer, generating extracted features. This proposed model is different from the traditional 1D convolutional recurrent neural network, which learns n feature maps by optimizing the n designed filters from the input representations. Our proposed spatio-temporal kernel serves as an internal operation in the framework, which enable this framework can extract much more complex features.

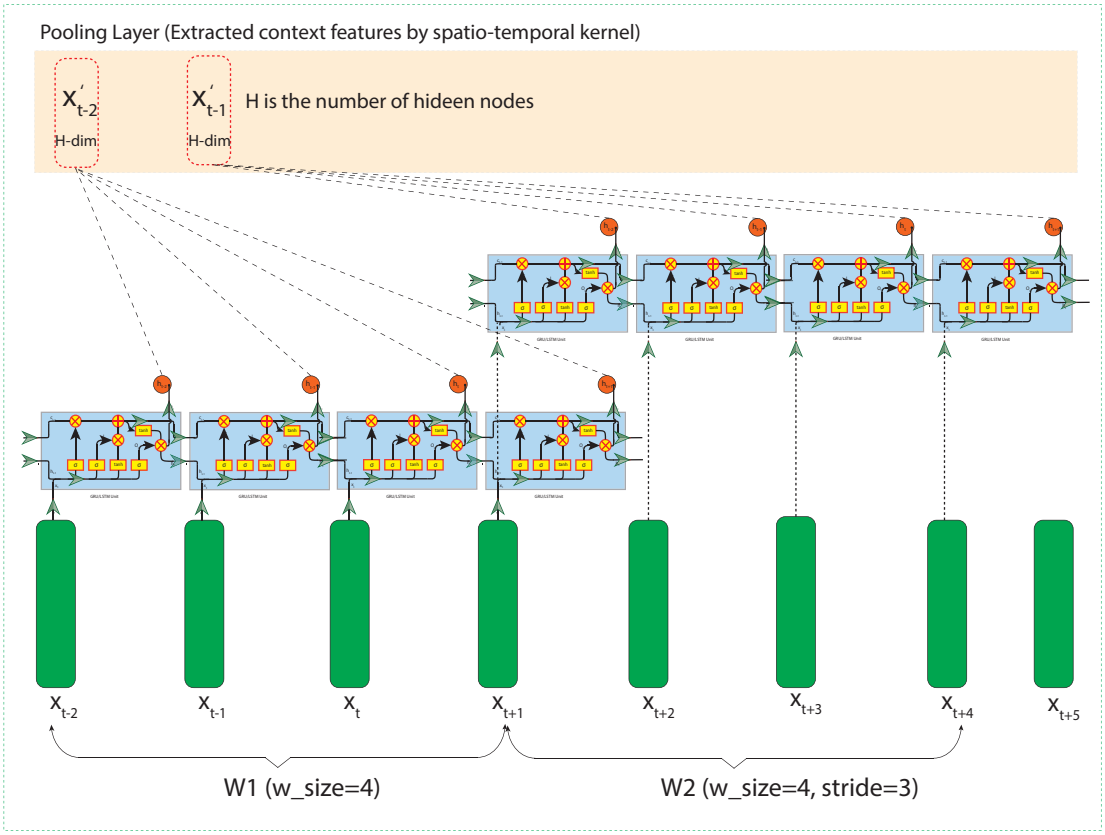


Figure 4.5: The architecture of spatio-temporal kernel.

In order to extract a window of meaningful local features from the input sequences with the proposed model, we employ a preprocessing technique to keep the number of input frames, input features and the number of dimensions, per feature constant. Specifically, let X represent one complete action skeleton sequence from a dataset, let N be the number of frames of an action sequence X, and use $X_1, X_2, ..., X_N$ to denote N frames of the whole sequence. $X'_1, X'_2, ..., X'_{N'}$ are the features corresponding to $N'(N' \leq N)$ windows with window size as w_{size}. Let l denote the dimension of the extracted features per frame, the input representation for the spatio-temporal kernel therefore is of size $l \times w_{size}$. In this case, each frame is represented as one-dimensional feature vector, with length to be l. For a complete action sequence, we slide an inspection window w, comprised of w_{size} consecutive frames, according to a stride length of s, until the entire action sequence is examined completely. This window of features, as a short sequence signal with shape $l \times w_{size}$, serve as an input to our spatio-temporal kernel. Then, the spatio-temporal kernel, as a features extraction function, produces the corresponding hidden states $(h_1, ..., h_{w_{size}})$ for each

window input sequence. These output features are then used to describe the input window w frames by utilizing a pooling operation (max, average or other customized functions), generating the final feature vector of size $H \times 1$, where H is the number of hidden nodes of the spatio-temporal kernel. In turn, these extracted features are concatenated together through time, producing another input sequence called X', which contains N' frames of size H each.

4.3.5 Variation of the proposed model

The backbone of our proposed framework is built on the multi-layers LSTM model and the embedded spatio-temporal kernel. Taken the input sequence of skeleton-based features, the spatio-temporal kernel refines the representations hierarchically. In order to enhance the performance of the spatio-temporal kernel and extract better representation for each window worth of input frames, we can substitute the LSTM network as the BiLSTM network, the Gated Recurrent Unit (GRU) and the vanilla RNN model can also be the alternatives.

4.4 Empirical testing and analysis

4.4.1 Datasets

We tested our proposed model on three benchmark datasets, including the UT-Kinect action dataset, the SBU-Kinect Interaction dataset and the UTD-MHAD dataset.

UT-Kinect-action dataset: The UT-Kinect action dataset [66] consists of the following 10 actions: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands. These actions were acquired by using a single Kinect camera, and all actions are performed by 10 different actors, and each actor performed each of the actions two times. There are 199 action sequences, each sequence consists of a different number of frames and each frame contains 20 3D joint coordinates.

SBU-Kinect Interaction dataset: The SBU Kinect interaction dataset [159] is captured by using one Kinect V1 camera within the controlled settings. This dataset is designed for depicting the interaction of two-persons, aiming at addressing complex action recognition problem. This database is still a challenging action recognition database for the existing

action recognition models, even though all of the interactions in this dataset are simple, mainly because the length of each of the sequences is relatively short and because of the low accuracy of the joint coordinates. It contains 6614 frames for the eight types of interactions, e.g., "approaching", "departing", "kicking", "punching", "pushing", "hugging", "shaking hands" and "exchanging". This dataset is carefully designed for interaction action recognition, since all frames of this database include two actors performing an interaction actions.

UTD-MHAD dataset: As introduced in the previous Chapter, the UTD-MHAD dataset is one challenging dataset, which provides a limited action instance for training and it includes 27 actions, e.g., "swipe to the left", "swipe to the right", "wave", "front clap", "throw", "cross arms", "basketball shoot", "draw x", "draw circle (CW)", "draw circle (CCW)", "draw triangle", "bowling", "boxing", "baseball swing", "tennis swing", "arm curl", "tennis serve", "two hand push", "knock door", "catch", "pick and throw", "jogging", "walking", "sit to stand", "stand to sit", "forward lunge", "squat". The large number of actions pose an extra challenge for action recognition on this dataset.

4.4.2 Implementation Details

The proposed model is implemented with the Tensorflow toolkit. The start learning rate is assigned as 0.001, and we report all the best results after 50 epochs of training on the testing dataset. After some preliminary experimental testing, we found that the performance of the proposed framework is not sensitive to the number of the hidden nodes, therefore we set the number of hidden nodes to be the same as the dimension of the input features. The spatio-temporal kernel and the backbone recurrent neural network contains the same number of hidden nodes. For the spatio-temporal kernel, we use max-pooling at the top layer and the LSTM unit is utilized in the following experiments. In order to investigate the performance of the variant model, in section 4.4.4, we report the effects of the different number of hidden units of recurrent network to our model. The values of the initial weights for the model are initialized randomly from the uniform distributions. The models are trained with the Adam optimizer with momentum.

4.4.3 Results and comparisons

The proposed model is evaluated on three datasets, e.g., the UT-Kinect dataset, the SBU-Kinect interaction dataset and the UTD-MHAD dataset.

Table 4.3: Performance comparison of SKB-TCN model and related models on UT-Kinect, SBU-Kinect and UTD-MHAD dataset.

#	UT-Kinect	SBU-Kinect	UTD-MHAD
Skeleton Joint Features [160]	87.9%	-	-
Elastic functional coding[161]	94.9%	-	-
Lie Group [71]	93.6%	-	-
Ensemble Learning [157]	96.97%	-	-
Body-pose features [159]	-	71.31%	-
ST-LSTM [153]	95.0%	93.3%	-
Multiple instance learning [159]	-	80.3%	-
Contrast mining [162]	-	86.9%	-
CHARM [163]	-	83.9	-
Active Joint Interaction Graph [164]	-	94.12%	-
Hierarchical RNN [141]	-	80.35%	-
Co-occurrence LSTM [154]	-	90.41%	-
Body part-based features[134] -	93.47%	-	
Bag of Points [65]	-	-	72.9%
SOF_#01	76.5%	59.4%	4.4%
SOF_#02	92.2%	**89%**	69.8%
SOF_#03	90.6%	77.5%	50.0%
SOF_#04	**95.3%**	86.9%	51.3%
SOF_#05	64.1%	69.4%	48.4%
SOF_#06	92.2%	71.8%	**79.6%**
SOF_#07	70.3%	31.3%	54.9%
SOF_#08	12.5%	26.9%	3.6%

Table 4.3 presents the summarized results of our proposed method and a comparison with the results of other related models. It can be seen that the proposed model can achieve competitive

results on the first two datasets compared with the existing models. The results of our model are listed in eight parts, the results of the different features are shown as an individual input.

4.4.4 Results of UT-Kinect dataset

We follow the experimental setup proposed by [160], in which the first 5 subjects are used for training while the remaining 5 subjects are used for testing to test our proposed model. As shown in Table 4.3, some features works very well on this dataset, while some cannot perform well on this dataset. For example, the proposed feature SOF_#04, which is known as distance between the joint and joint, can achieve the highest accuracy on this dataset. And also, we can note that the proposed feature SOF_#08, which is known as the angle between the pre-defined planes, cannot achieve good accuracy. The convergence rate curves for all the proposed eight features are shown in Fig.4.6. From the graphs shown in Fig.4.6, we can see that there is an overfitting problem and the training process is not stable.

In order to exploit the proposed features further, we investigated the derived motion and the energy features on this dataset extensively. The results with the different configurations of the derived motion and the energy features of the aforementioned features on this dataset are shown in Table 4.4. We can notice that the combination of the two features that performs individually well on this dataset, but cannot always produce the best result. It is worth pointing out that the feature combination of SOF_#06 + M outperformed the other features and SOF_#02 + E, SOF_#02 + M + E , SOF_#06 + E also have significant discriminability and can achieve competitive performance. The best result of our model hit the state-of-the-art results (98.4%) compared with the previous best model [157] (96.97%) on the UT-Kinect-Action Dataset.

The above reported results are achieved with a window size of 3 and stride of 1. Since the proposed model is flexible to be configured with a different window size and stride, we evaluated our model with different settings for these two parameters on the most significant features and combinations, namely, the SOF_#04, the SOF_#02 + E, the SOF_#02 + M + E, the SOF_#06 + M, and the SOF_#06 + E. As shown in the Table 4.5, the SOF_#04 and the SOF_#06 + M can both achieve the highest recognition accuracy on this dataset with different settings. We can generally conclude that different features present a different performance with a different window size and stride settings. This verified the assumption

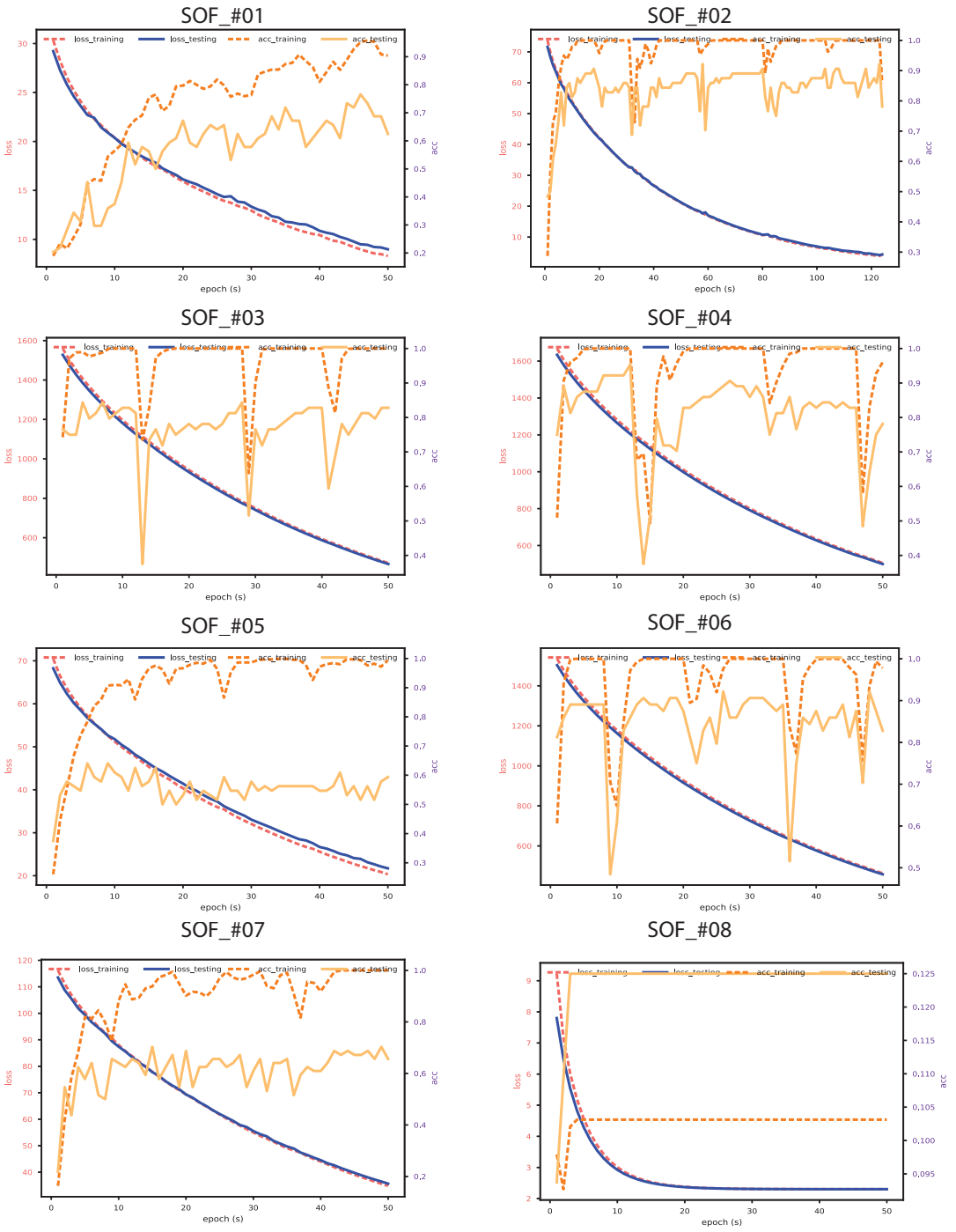


Figure 4.6: Convergence rate curve for different geometric relational features for UT-Kinect

that different geometrical features present different temporal patterns in the time domain. It is worth noting that, the same window-size with big stride will lose some temporal

Table 4.4: Performance of different combination with the derived motion and energy features on UT-Kinect dataset

#	M	E	SOF_#0N + M	SOF_#0N + E	SOF_#0N + M + E
SOF_#01	10.9%	12.5%	73.4%	76.5%	70.3%
SOF_#02	70.3%	54.6%	93.8%	**95.3%**	**95.3%**
SOF_#03	85.9%	76.6%	85.9%	90.6%	89.0%
SOF_#04	70.3%	64.0%	92.2%	92.2%	93.8%
SOF_#05	17.2%	12.5%	64.0%	64.0%	70.3%
SOF_#06	87.5%	70.3%	**98.4%**	**95.3%**	93.8%
SOF_#07	15.6%	56.3%	67.2%	70.3%	70.3%
SOF_#08	7.8%	5.0%	14.1%	14.1%	10.9%

information, and this results in degrading the performance of the recognition accuracy. In the future we will investigate methods to extract the best temporal patterns from the different features.

Table 4.5: Recognition accuracy based on the derived features with different configurations

#	SOF_#04	SOF_#02 + E	SOF_#02 + M + E	SOF_#06 + M	SOF_#06 + E
$window\_size = 3, \text{stride} = 1$	95.3%	95.3%	95.3%	**98.4%**	95.3%
$window\_size = 5, \text{stride} = 3$	96.8%	95.3%	90.6%	79.7%	71.9%
$window\_size = 7, \text{stride} = 3$	**98.4%**	96.8%	93.8%	85.9%	96.8%
$window\_size = 7, \text{stride} = 5$	96.9%	92.2%	93.8%	93.8%	95.3%

4.4.5 Results of SBU-Kinect Interaction dataset

The experimental results for the SKB-TCN model on the SBU-Kinect interaction dataset are listed in Table 4.3. As the results in Table 4.3 shown, the features based on skeleton, motion and energy concept outperform the other feature choices. We compared our proposed models with [141], [154], [164], and the large margin between our model and the previously existing models can be attributed to both the discriminative ability of our proposed features and the spatio-temporal kernel. Because the interaction actions involve more geometric relationships than those actions that are performed by only one person, this can be explicitly embedded in our proposed features.

In order to verify the effectiveness of the proposed approach based on the derived motion and energy features on this dataset, we divide the whole dataset into two subsets, 25% samples

are used for testing and the remaining samples are used for training. The recognition accuracy for the different combinations are presented in Table 4.6.

Table 4.6: Performance of different combination with the derived motion and energy features on SBU-Kinect.

#	M	E	SOF_#0N + M	SOF_#0N + E	SOF_#0N + M + E
SOF_#01	64.4%	57.8%	17.2%	15.9%	55.3%
SOF_#02	75.9%	15.3%	85.9%	71.6%	**89.1%**
SOF_#03	68.8%	68.8%	71.6%	79.7%	75.0%
SOF_#04	75.3%	15.9%	**90.6%**	**93.8%**	87.8%
SOF_#05	35.0%	15.3%	62.8%	60.9%	65.9%
SOF_#06	65.6%	61.6%	76.6%	73.4%	81.3%
SOF_#07	52.2%	44.7%	34.4%	29.1%	32.2%
SOF_#08	17.5%	15.3%	29.7%	32.8%	35.9%

From the experimental results, we can identify that the proposed feature can work together with their derived motion and energy features. However, the performance of the derived motion and energy features based on the geometric feature cannot perform consistently well. In Chapter 6, more approaches will be investigated to explore the potential of our proposed geometric features .

4.4.6 Results of UTD-MHAD dataset

The UTD-MHAD is one of the most challenging action recognition datasets, because of the large scale of actions that are included with limited training examples. The low resource of training data in this database poses a great challenge to the deep learning based approach. We followed the cross-subject evaluation protocol proposed by [27], which proposed to use the first 5 subjects as training dataset and the other 5 subjects are used for testing. It can be observed from Table 4.3, the proposed feature SOF_#04, which is known as the distance between joint and joint, can achieve the highest accuracy on this dataset. And we can also note that the other proposed features cannot achieve good accuracy, and they cannot even converge.

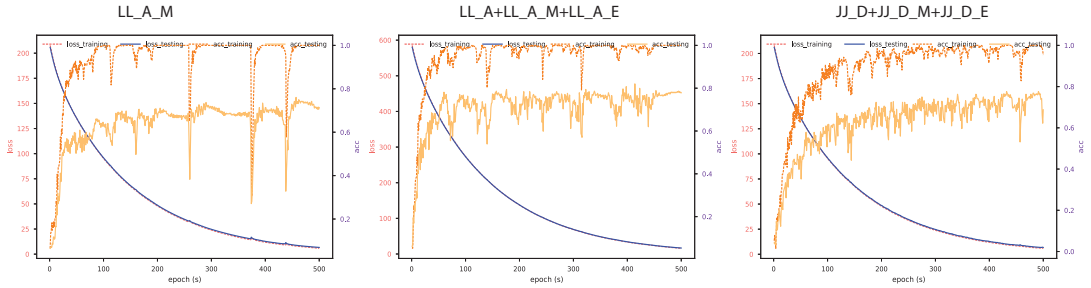


Figure 4.7: Convergence curve for best three combinations on UTD-MHAD

The recognition accuracy for the different combinations for the original features and their derived motion and energy features are presented in Table 4.7. The convergence curves for our best three results are shown in Fig.4.7.

Table 4.7: Performance of different combination with the derived motion and energy features on UTD-MHAD.

#	M	E	SOF_#0N + M	SOF_#0N + E	SOF_#0N + M + E
SOF_#01	3.9%	4.2%	44.8%	7.3%	3.9%
SOF_#02	7.0%	12.2%	74.0%	66.9%	77.6%
SOF_#03	43.2%	27.6%	57.6%	53.1%	48.2%
SOF_#04	8.9%	17.2%	46.9%	12.1%	49.2%
SOF_#05	3.6%	3.6%	43.5%	3.0%	46.6%
SOF_#06	7.6%	56.0%	81.3%	36.0%	82.0%
SOF_#07	3.9%	4.2%	12.2%	8.1%	53.9%
SOF_#08	3.6%	3.6%	4.4%	5.2%	59.3%

From the obtained results on this database, we can see that the performance of the proposed geometric feature and their derived features cannot perform very well compared with the previous two datasets. This reason for the performance degrade is due to the large scale of the actions included in this database. Based on the experimental result we can argue that the recurrent neural network can work efficiently with the proposed geometric features for action recognition on small dataset, but the performance will degrade on large dataset. The performance degrade on large dataset can generally be attributed to the following two reasons: 1) due to the low discriminability of the features for the large-scale of actions, the

different people present a different action performance and produce different geometric features; 2) the recurrent neural network is not easy to train, especially if the temporal dependency relationship is complex.

4.4.7 Ablation Study

In this ablation study [1], we aim to highlight the advantage of our model compared with the normal LSTM model, we tested the performance of SOF_#02 feature on the UT-Kinect with normalization length as 50. Utilizing the SOF_#02 feature, we can achieve 95.31% accuracy while we use the SKB-TCN model, and 87.5% accuracy if we use the traditional LSTM model. The convergence rate curve is shown in Fig.4.8. We envisage this phenomenon will be more obvious while the sequence is involved with more noise.

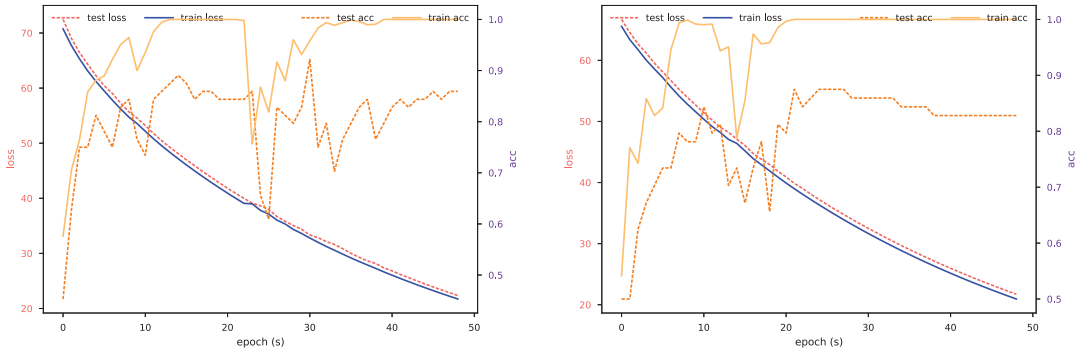
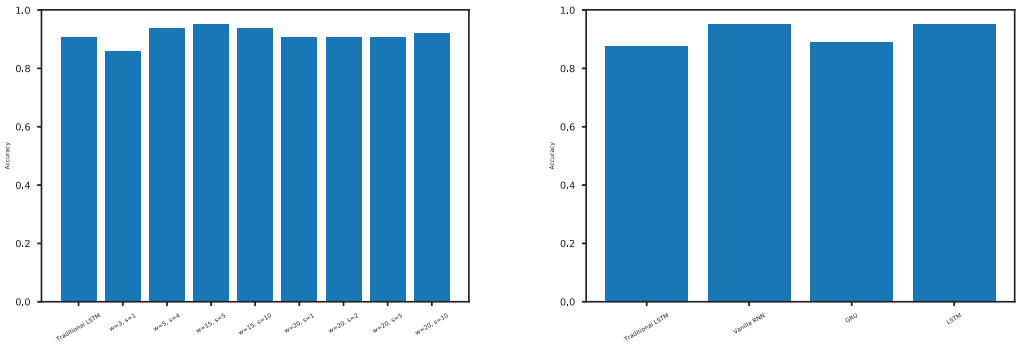


Figure 4.8: Convergence rate curve for the proposed model with(left) and without(right) spatio-temporal kernel

Aiming at suggesting an optimal setting for the spatio-temporal kernel, we examined the effect of the different settings of the kernel for the normalized length with 100 frames on the UT-Kinect dataset further. The performance of the spatial-temporal kernel with different settings taking the SOF_#02 as the input features are shown in Fig.4.9-(a). It can be observed that when we set the stride size between 1/3 and 1/2 of the window size, the performance of the spatio-temporal kernel can outperform the traditional model. Moreover, we don't encourage setting the window-size too large, because it will slow down the convergence rate.

[1] An ablation study typically refers to removing some "feature" of the model or algorithm, and seeing how that affects performance.



(a) Performance of various settings for spatio-temporal kernel

(b) Performance of variant spatio-temporal kernels

Figure 4.9: Performance of variant model with spatio-temporal kernel

As stated in Section 4.3.5, our proposed spatio-temporal kernel can be configured with different recurrent units, which can be the LSTM, the GRU and the Vanilla RNN model. In order to verify the performance of different variants model, we test the variant model on the UT-Kinect dataset with the normalized length of 100 frames and as suggested by the above experiments, the window size is 15, and the stride size is 5. The performance of different models is shown in Fig.4.9-b. As can be observed from the graph, even though the vanilla RNN unit can achieve similar recognition accuracy with LSTM model, we suggest to use LSTM model because the output model is much more robust than the vanilla RNN model and can avoid gradient vanishing and error exploding problems efficiently. In the future, we intend to utilize network-based kernels to enhance this framework, so that it can be extended to other applications as well.

4.5 Summary and contributions

In this Chapter, we demonstrated the performance of the SKB-TCN model with our proposed features as input. The following contributions can be identified:

1) We proposed several skeleton based optical flow-guided features to characterize the human actions. As shown in the experiment sections, different features present different performance on each database. This can be attributed to the variation of actions performed by different actors in different environmental settings.

2) We incorporated the Spatial-Temporal Kernel into the LSTM-RNN model. This model can learn the spatial relationship among multiple attributes that are used to represent the human activities. By extending the RNN model to the spatial domain, we demonstrated how the SKB-TCN model can explore the constraints and correlations between the geometric relational features in a single frame. It can also learn the temporal dependencies between the consecutive features from a sub-sequence of frames and aggregate them to get compact features for the subsequent layers of the recurrent neural network model.

Chapter 5

Action Recognition with CNN Features using LSTM-C RNN Model

Recognizing activities from RGB-videos is the foundation of various computer vision applications, for example, automating action recognition can advance the way we monitor activities so that it is not necessary to ask a human to classify each video by watching them day and night (e.g. for airport security purposes, etc.), not unlike a computer that never gets tired. However, the classification of activities from a video is still a challenging topic since we need to consider spatial and temporal features simultaneously that together represent each activity. Previous chapters explored the spatio-temporal feature extraction from skeleton data, while the extraction of spatio-temporal features from RGB video is much more difficult than the previous one. Therefore, an efficient approach for obtaining discriminative spatial and temporal features from RGB videos is proposed in this Chapter. This approach first extracts CNN features from segmented video frames, and then uses our proposed LSTM-C model to extract global features from a video level perspective for classification.

5.1 Motivation

With the impressive advancements achieved by deep learning models in recent years, the CNN and RNN models have attracted attention, and have been extensively exploited in vision-related tasks because of their powerful feature extraction ability [165]. Combining CNNs and RNNs expands the diversity of architectures available for researchers solving action understanding tasks [166], [167]. As reported in the literature, variants of both the RNN and CNN models have demonstrated their powerful representation learning ability in

vision-based problems (e.g., image/video classification) and time-series problems (e.g. language modeling, speech recognition and image/video captioning).

Computer vision is a research line that uses computer science techniques and artificial intelligence algorithms to extract rich representation from the input images or videos, aimed at understanding 2-dimensional or 3-dimensional images for promising vision-based applications. Action recognition focus on interpreting and acquiring meaningful representation of human movements from the provided images or videos. Applications of human action understanding can be found in search engines for image identification, surveillance systems for security and healthcare monitoring for patients, and so on [147]. Therefore, more advanced algorithms should be developed for discriminative representation learning for these applications, which can help to provide more accurate solutions for realistic applications and save the cost and time for a particular application.

Even though optical flow-based approaches[168] were the most popular solution for HAR, but the limitation of this approach is that it cannot capture the sudden changes robustly. For example, in some constantly changing image structures in videos they usually includes some key points, which present non-constant motion, which may provide important cues for characterizing the change of the image structure. The Local feature descriptor for static images, e.g., 2D SIFT [169], were adopted for action recognition in traditional approaches. However, the depth information was lost in 2D images. In order to overcome this challenge, 3D SIFT [170] descriptors are developed to extract features that contain 3D information. In summary, the performance of all of these methods is limited mainly because of their insufficient representational ability.

Most recently, action recognition from videos with CNN-based features has attracted increasing attention in video classification. Several successful models were developed based on RNNs, taking frame-level CNN-based features for video-based action recognition. However, the aforementioned framework takes extracted features based on pre-trained CNN models as inputs for the LSTM classifier. In our opinion, the modelling of temporal relationship between these features extracted by the CNN model may be improved to capture richer motion information. It is a known problem that successful regularizers for the CNN models does not work well with RNNs and LSTMs, therefore, a proper regularizer is needed to regularize the learning process. Therefore, we introduce a time-series correctness-vigilant

regularizer via adding our novel constituent nodes to the LSTM cells. Altogether, we propose a novel pipeline that involves an attention-aware CNN-RNN model, using the LSTM with constituent nodes (LSTM-C), for video-based action recognition.

5.2 Related work

Automatically classifying videos is one significant task for scene understanding, which is a major goal of computer vision. While detecting and classifying human actions is a more challenging problem in video classification, it not only requires the representation learning model to be powerful enough to localize the human beings and capture their motion in a short time period. For this reason, it is still is an unsolved problem for most of the current advanced algorithms, since the algorithms need to mimic the special human ability to interpret the visual and motion variation of the videos. The literature that is most related to our proposed approach includes following two categories. The first one combines the local space-temporal engineered descriptor [171] with deep learning techniques, which is similar to the approach introduced in Chapter 3. The second one is based on deep learning techniques that use neural network to inference representation from the original images. We employed the second approach and proposed a framework to efficiently and effectively incorporate the spatial-temporal information for action recognition in this Chapter. Motivated by the latest approaches for image classification, we introduced the CNN-based image features into our framework because it can extract semantic features robustly that are related to subjects from the video sequence.

As stated in Chapter 2, most carefully designed descriptors used in image processing [172] have been transferred into the action recognition area. In order to discover more useful video-based features for video-based action recognition, some researchers attempt to use deep learning techniques to extract video representation directly [15], [88]. Different from image classification, a video sequence that includes human actions usually presents the evolution of visual appearance along the time axis, basically these are 3-D signals. Following this research idea, Ji et al. [173] customized a 3-D CNN model based on the 2-D CNN model, that is able to extract both the spatial and temporal features efficiently. Simonyan et al. proposed another successful model, the two-stream ConvNet [88]. This model can capture

both the appearance information from the static video frames and the motion features from the consecutive video frames. More recently, some of the literature shows that the LSTM model is able to learn the long-term dependency relationship by introducing memory units into the vanilla recurrent neural network. And it combines the CNN and RNN model for the spatio-temporal feature extraction from the video and since this is the most natural choice, some breakthroughs were achieved on several benchmark datasets with this idea. However, due to the fact that the existing models that take the frame-level CNN-image features as input for the LSTM models, it is not easy to remove redundant information of the input sequence and it may fail to capture the most important temporal patterns from consecutive frames. Another limitation of the RNN neural networks for action recognition is that the training is not as efficient as the CNN models, because there are a lot of training tricks that can be used on CNN training, therefore in this Chapter, we introduced a novel regularize mechanism for RNN model, which can be extended to other time-series problems.

5.3 The algorithms

5.3.1 General architecture

As a well-known and important solution for the modern video-based action understanding system is the choice of the combination of both the visual and the temporal representation. In order to address the above two issues and extract the spatial and temporal features simultaneously from videos, combining the CNN and the LSTM-C model for video based action recognition is a promising approach. We integrate the CNN features as part of the spatio-temporal features fed into our customized LSTM nodes. The whole pipeline shown in Fig.5.1 includes the following two stages, the pre-processing phase and train the LSTM-C models (classification). In the preprocess phase, we augment the training data by randomly selecting the fixed length of frames from the RGB frames of RGB video.

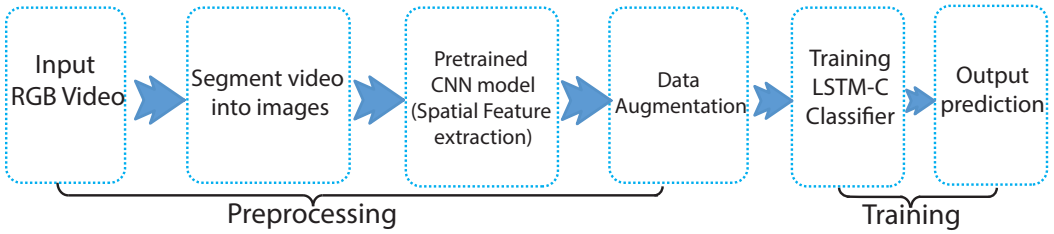


Figure 5.1: The workflow of the proposed CNN-LSTM-C framework

With this architecture, we enable the network can detect what we call the significant action motions. With the CNN feature extraction as a preprocessing step, the proposed framework can extract more semantic features from the raw images for video classification. Then extracted feature vectors were fed into the subsequent recurrent learning classifier, which then extract the final representation of the current input video. The pseudocodes of the workflow for the proposed LSTM-C and the whole framework is shown as code block 6 and code block 7:

```
input_feature  # extracted features with VGG, shape: 20488*N, N indicates
    the length of the video
memory_size # memory_size
o_t # hidden nodes
c # memory state
O_t # constituent node output

# modify the output of the LSTM model with proposed  algorithm
for i in range(len(input_feature)):
   o_t, c = LSTM(input_feature[i])

   # calculate the constituent nodes with Equation 5.2
   O_t = constituent_nodes(o_t(t-2), o_t(t-1), o_t(t)) # memory_size=3
```

Code block 6: *The pseudocode for one layer LSTM-C model*

The hyperparameters that can be tuned in the proposed framework are listed in Table 5.1.

Table 5.1: Hyperparameters settings in our proposed model

parameter	Description	Default value
memory_size	the memory size of LSTM_C model	3
nb_layers	number of layers of LSTM model	1
nb_hidden	number of hidden node of LSTM units	128
lr	learning rate	0.001
length	length of the normalized sequence	30
epoch	epoches of training	500
batch_size	batch size of training	64

```
# Preprocessing
videos  # given path of videos

num_videos = len(videos)
for i in range(num_videos):
   images = ffmpeg_segment(video[i])
   cnn_features = VGG(images)  # use VGG model

   # Make tfrecord example
   for feature in features:
      ex = make_example(feature)
      writer = TfRecordWriter(save_path)
      writer.write(ex.SerializeToString())  # save features
      writer.close()
# given training and testing path
Train_path, Test_path
# Define the proposed model
model = LSTM_C(input_feature, memory_size, depth, num_layers)  # code
    block 4

for i in range(epoch):
   label, input_feature = Load_training(batch_size, Train_path)
   outputs = model(input_feature)
   outputs_prob = Dense(num_classes)(outputs[-1])

   if i % 2 = 0:
      label, input_feature = Load_testing(batch_size, Test_path)
      accuracy = test(label, input_feature)
```

Code block 7: *Extract CNN-features for N videos and train the LSTM-C model*

5.3.2 Workflow of the CNN-LSTM-C framework

As demonstrated in the code block 4, the whole architecture of this framework can be illustrated as Fig.5.2, which can be divided into four phases: 1). Pre-processing phase which involves segmenting the input videos into static images, which are used as input for the subsequent CNN models for feature extraction; 2). The spatial feature extraction phase with the VGG model, which processes the input images and outputs one compact feature vecto and finally 3). The spatial-temporal feature extraction phase, training our proposed LSTM-C model with extracted features in step 2 as input to reach the fourth phase; 4). Classification phase with Softmax layer.

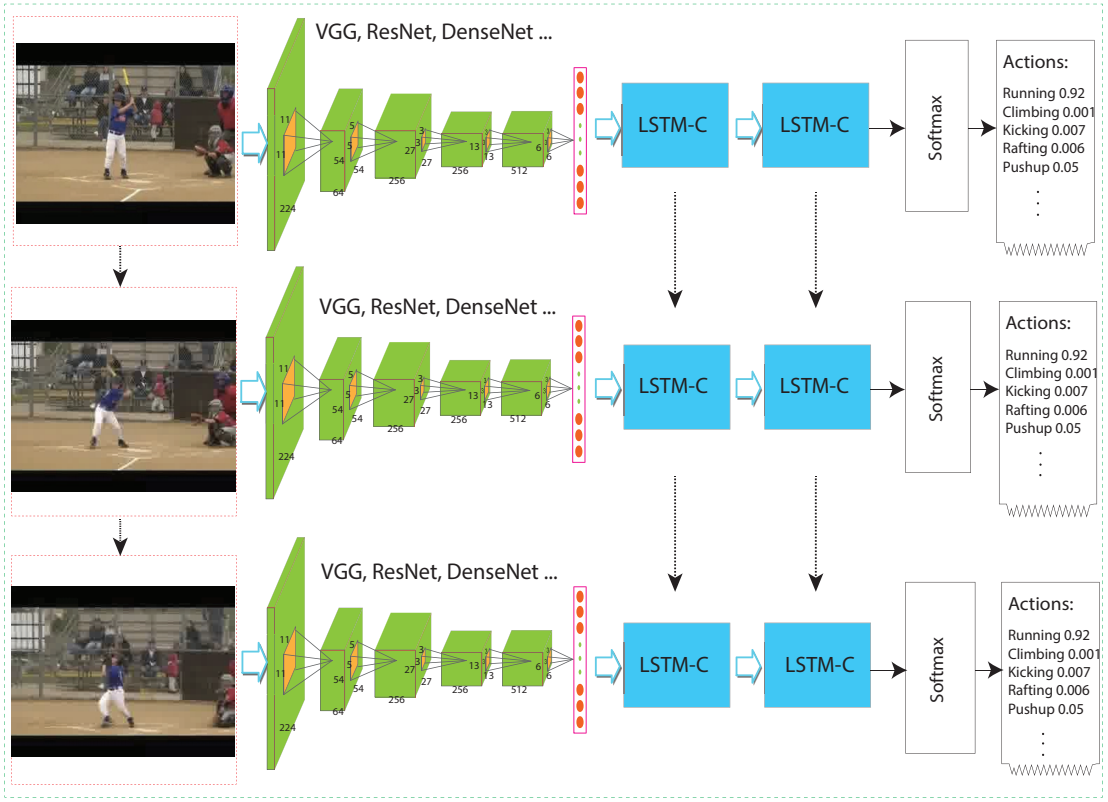


Figure 5.2: Proposed framework for video-based action recognition

5.3.2.1 Pre-processing phase

Due to the fact that the pretrained CNN models can only work on static images, therefore we segment the videos into a sequence of image frames. And the pretrained CNN model takes a fixed size of images as input for feature extraction. We then resize the segmented images into fixed size, e.g., 224×224 according to the requirement of the adopted pre-trained CNN model.

5.3.2.2 CNN feature extraction

With these segmented images, the VGG model is used to extract the compact feature vectors from the preprocessed images.The video frames derived from the RGB video are first resized to 224x224, and then fed into the VGG model, which outputs a fixed dimension (2048) vector for each image, and a sequence of this vector is used to represent the input video. In the training and testing phases, the extracted flat vectors are fed into our proposed LSTM-C units to extract the spatio-temporal features for the whole video.

5.3.2.3 Classification phase: LSTM-C

The RNN model is good at solving the sequence-to-sequence modeling problem, which has been successfully used in many commercial applications. As regards the RNN Network training, gradient explosion or vanishing is one common problem, and there are various ways to mitigate this problem, e.g. proper initialization of the weight, regularization, proper activation function (ReLU is preferred than sigmoid or tanh activation function). The ReLU derivative is a constant value, which is 0 or 1, this advantage of the ReLU activation function can efficiently prevent the gradient vanishing problem. Another advanced solution is to employ the LSTM or GRU model, which are especially proposed for addressing the vanishing gradients problem so as to discover the long-term relationships from the time series signals.

Although both the LSTM and GRU model can solve the convergence problem to some extent, what could be further improved is the discriminative ability of the extracted features. This thesis proposes an LSTM-C architecture based on the traditional RNN architecture, which can work with both the LSTM and GRU units. These two architectures are devised to prohibit the vanishing gradients via the integrated gating mechanism. Based on the existing framework, aiming at improving the representation learning ability of the LSTM/GRU based architecture, we propose a modified version, which is called the LSTM-C.

The output of LSTM or GRU units is the new hidden state s_t, but they are calculated in a different way compared with vanilla RNN model. In the vanilla RNN model, the hidden state can be calculated by

$$s_t = \tanh Ux_t + Ws_{t-1} \tag{5.1}$$

where x_t represent the inputs feature for current time step t, and the s_{t-1} indicates the state of previous hidden nodes. Previously, the network treats the LSTM or GRU as a black box, they output the next hidden state based on the current input and previous hidden state recursively. Yet in reality, each decision can be made dynamically, by considering and comparing several historical outputs and then deciding the final output of the model.

In our LSTM-C architecture, even though most of the calculations are the same with the LSTM, we devised a novel way to compute the final hidden or output state by involving a new algorithm. In order to make this Chapter self-explanatory, we will recap some of the basics

of the popular LSTM model, and then introduce our proposed LSTM-C unit in the following sections. With this architecture, the final action recognition result can be obtained by using a softmax layer which assigns a probability distribution to each of the possible actions in the problem domain.

5.3.2.4 LSTM with constituent node

As stated in Chapter 4, the RNN-based classification approaches utilize the output of one LSTM Unit to encode the original single input frame, and use the output of the last LSTM unit to encode the whole input sequence. Nevertheless, for video-based action recognition, this approach has significant limitations, since the last frame of the video will have the most influence on the action class recognition result. On the other hand, the previous frames will have minimal impact on the classification due to the forgetting effect. Regardless, a human action is often defined by the whole movement process, and while some frames are the key frames,

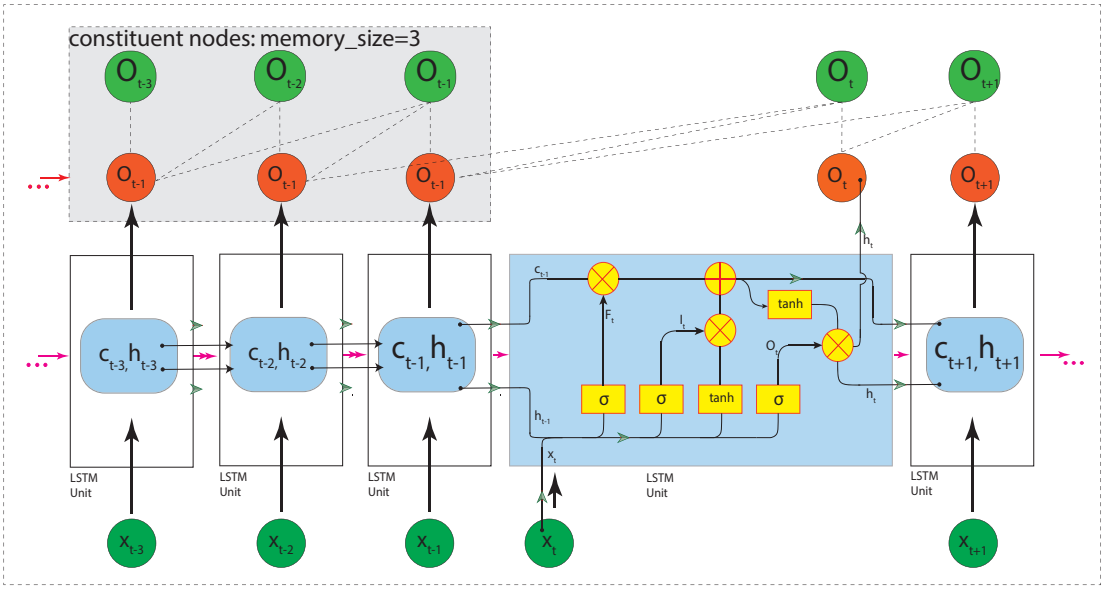


Figure 5.3: The architecture of the LSTM-C model

some frames are irrelevant, which should be ignored. Motivated by this instinctive phenomenon, we propose a novel LSTM architecture which is called LSTM with constituent node (LSTM-C), that integrates an external constituent node for each LSTM cell. Each constituent node looks at a history of output vectors corresponding to human movements,

then assigns weights proportional to the correctness of the probability distribution of each LSTM node. The architecture of the LSTM-C model is demonstrated in Fig. 5.3.

As illustrated in Fig.5.3, a historical output vector O_t is introduced at time t. It is generated by a score weighting scheme using the output response vector o_t at time t and the historical output at time t-1, t-2 and so on. The update equation of the historical output vector O_t can be expressed as the following formula:

$$O_t = \begin{cases} \alpha_t o_t + (1-\alpha_t) O_{t-1}, & \text{if } \epsilon_{h_t} \geq \epsilon_{o_{t-1}} \\ \sum_{k=1}^{t} w_k^t o_k, & \text{if } \epsilon_{o_t} \leq \epsilon_{o_{t-1}} \end{cases} \tag{5.2}$$

where α_t is the weight controlling the balance between the current response o_t and the historical state O_{t-1}. The weight α_t is calculated by the following formula:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{\epsilon_{o_{t-1}}}{\epsilon_{o_t}}\right) \tag{5.3}$$

where ϵ_{o_t} denotes the loss between the training label y and the estimated label $\hat{y_t}$ at time t using the softmax function on $c + VO_t$. w_k^t denotes the weight of response o_k. It is calculated by:

$$w_k^t = \begin{cases} 0, & \text{if } k \leq \tau \\ \frac{1}{t-\tau}, & \text{if } k \geq \tau \end{cases} \tag{5.4}$$

where τ is the parameter controlling the forgetting effect. Finally, a softmax layer is employed to provide the estimated label $\hat{y}$ of action video, which is determined by the last output O_T:

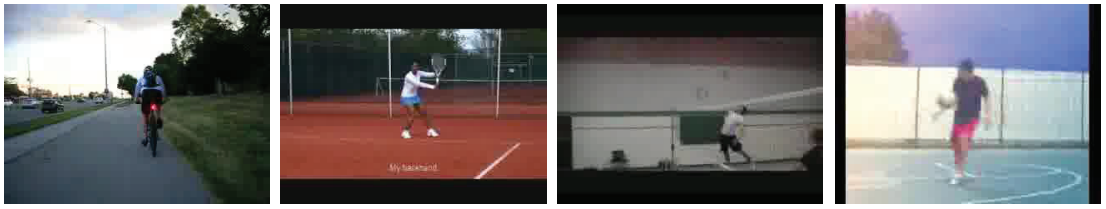
$$\hat{y} = softmax(UO_T + b) \tag{5.5}$$

where U and b are the bias vector and the weight matrix of the softmax layer respectively.

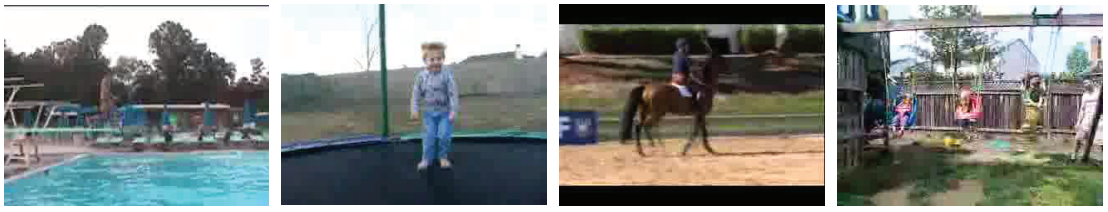
5.4 Empirical testing and analysis

5.4.1 Datasets

Several popular video datasets, e.g., the UCF11 dataset, the UCF Sports Action dataset and the UTD-MHAD dataset, are used as our testbed for our proposed framework. Fig.5.4 demonstrated four action samples of the first two datasets.



(a) UCF Sport dataset



(b) UCF 11 dataset

Figure 5.4: Samples of experimental action recognition datasets

UCF-11 Human Action Dataset includes 11 human action categories, 1600 sequences in total. The 11 categories are horse back riding, volleyball spiking, basketball shooting, walking with a dot, trampoline jumping, tennis swinging, swinging, juggling, soccer, golf-swinging, diving, and biking/cycling.

UCF Sports Dataset is a low resource database [174], which contains 10 sports actions in 150 video samples, including walking, golf swinging, swinging (on the bench), skateboarding, running, horse riding, weight lifting, kicking ball, diving and swinging. All of these videos were collected from various websites, e.g., BBC Motion gallery and GettyImages. The length of all of these video clips ranges from 2.2 seconds and 14.4 seconds.

UTD-MHAD Dataset provides multiple modality data, such as video, depth, skeleton and inertial data. More details can be referred to the introduction about this dataset in Chapter 3. One example action for the different signals of this dataset is presented in Fig.5.8.

5.4.2 Implementation Details

We conduct experiments with two goals. The first goal is to study the influence of the LSTM-C in our algorithm. Secondly, we compared our results with other state-of-the-art algorithms. We used the pre-trained VGG model to extract the CNN features, which is then fed into the LSTM-C model. In the LSTM-C model, there are 128 hidden neurons in the hidden layer. The probability of dropout is set as 0.5. The start learning rate is set as 0.001 for the LSTM-C model. The learning rate decay with a base of 0.1 every 10 epochs in the training process. The regularization value of the LSTM model is set as 0.003. The batch-size is set as 32. The Adam optimizer is employed to train the network for the LSTM-C model.

5.4.3 Results and comparisons

Three datasets, such as, the UCF-Sport Action, the UCF 11 and the UTD-MHAD, have been investigated with our proposed approach. We summarized the results of action recognition accuracy across these datasets in Table 5.2.

5.4.3.1 Results on UCF-11 Dataset

The UCF-11 dataset is a challenging human action recognition dataset, because of the complex environmental factors, e.g., the varied illumination, variation in object scale, varied viewpoint, complex background, and the low quality of the videos. Similar to the original setup, we select examples of one subject as the testing dataset and the left examples are used for training. The performance metric is calculated by the average accuracy over all of the experiments. The results indicate the robustness of our proposed model across the different configurations. As shown in Table 5.2, our proposed approach can increase the recognition accuracy over the best state-of-the-art results.

In order to compare our model with the traditional LSTM model, we configure our model with different configurations, the memory size ranges from 0 to 90. The model is the same with LSTM model if the memory size is set as 0. As shown in the above graph, even the traditional LSTM model can achieve similar recognition accuracy as with the LSTM-C model with a memory size of 30. Aiming at discovering more insights about the effect of the constituent nodes to the model training process, we show the training loss and accuracy and

Table 5.2: Comparison of LSTM-C and related models on UCF11 and UCF-Sports.

#	UCF 11	UCF Sport
Orientation tensor [175]	75.4%	-
Dense trajectory [33]	84.2%	89.1%
Tensor motion descriptor[168]	68.9	-
Local motion[176]	88.0%	-
Visual attention [177]	84.96	-
Interest points motion [178]	91.3%	-
Visual-DTAM [179]	91%	-
CRNN [180]	91.2%	-
Two stream LSTM	94.6%	-
Le et al. [89]	-	86.5%
Kovashka & Grauman[52]	-	87.2%
Souly & Shah [181]	-	85.1%
Wang et al. [182]	-	85.6%
Weinzaepfel et al.[183]	-	90.5%
LSTM-C($memory\_size = 3$)	97.9%	89.18%
LSTM-C($memory\_size = 15$)	97.3%	81.08%

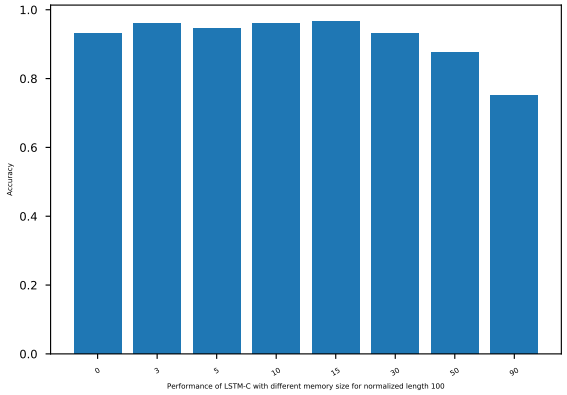


Figure 5.5: Performance of LSTM-C on UCF 11 dataset

testing loss and accuracy in Fig.5.6. It is evident to observe that the model with constituent nodes makes the training process much more stable than the original LSTM model, and it can

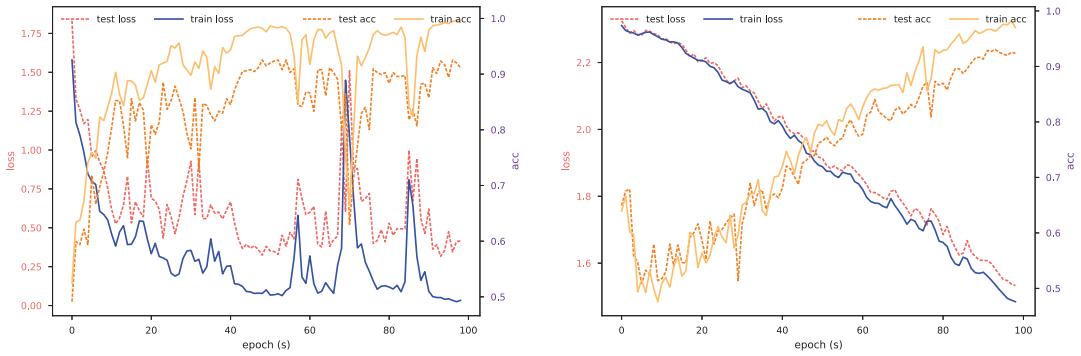


Figure 5.6: Convergence curves for LSTM and LSTM-C with $\mathtt{memory\_size} = 30$ on UCF 11

effectively prevent the overfitting problem.

5.4.3.2 Results on UCF Sports action dataset

The results of our model on the UCF Sports dataset is shown in Table 5.2. We follow the experimental setup that was proposed by [184], in this work they proposed to use 2/3 of the action examples for the training dataset, and the left as the testing dataset. In this way, the training dataset and the testing dataset includes 103 videos and 47 videos respectively. As described before, the lengths of these videos are different, so, in order to train in a batch manual, we rescale the length of each video clip into 24 frames in the preprocess phase. We compare our results with [52], [181]–[183], and our model can achieve competitive results compared with other solutions except for [183]. The results of [183] are better than ours, because they used video and optical flow as the input, while ours only utilizes the RGB video frames.

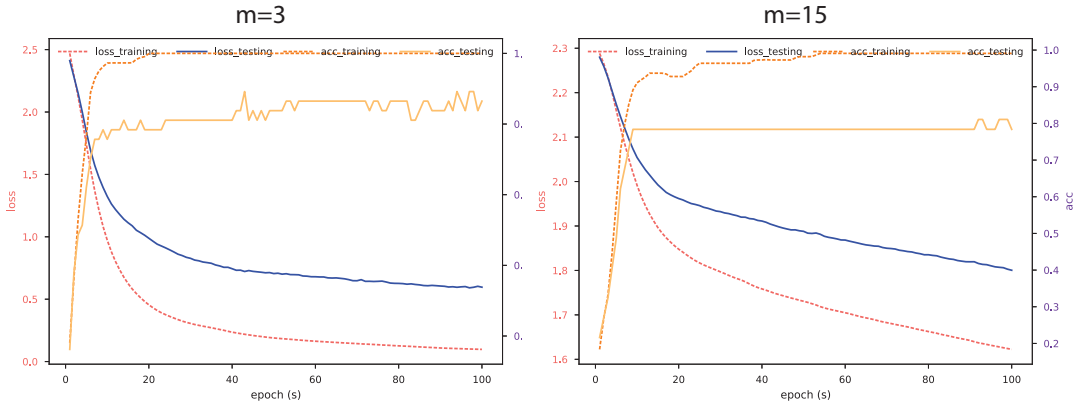


Figure 5.7: Convergence curves for RGB video-based action recognition for UCF-Sports

The convergence curve of the LSTM-C model with the memory-size configured as 3 and 15 is shown in Fig.5.7.

5.4.3.3 Results of UTD-MHAD database

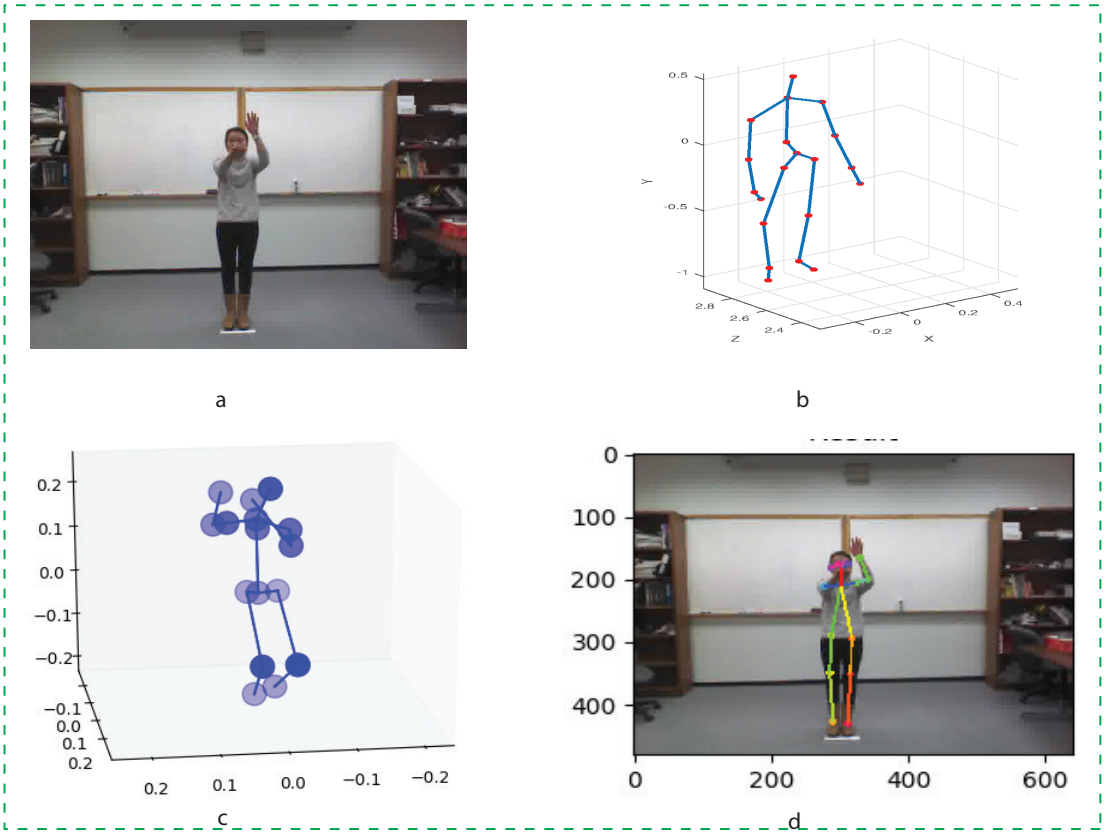


Figure 5.8: An action instance of UTD MHAD dataset: a) RGB image, b) original skeleton data (provided by the database, (x, y, z) for 20 joints), c) estimated noisy 3d skeleton data, d) estimated 2D skeleton data

Different from the previous two datasets, which contains only 11 and 10 actions respectively, the UTD-MHAD contains 27 actions, which is much more challenging compared with the previous two evaluated datasets. The result of our experiments on the UTD-MHAD dataset with video as input is shown in Table 5.3. Our approach with video as input on this dataset can only achieve 38% accuracy. In order to improve this, we attempt to estimate the skeleton data from the video for the video-based action recognition.

One example action instance is shown in Fig. 5.8, Fig. (a) shows one RGB frame for this action, while Fig. (b) shows the original skeleton data provided by the database. This skeleton

data was captured by a motion capture system, which is much more accurate than the skeleton data that was estimated by the pose estimation algorithm. Motivated by the advancement in pose estimation technology, we used OpenPose [185] to extract the skeleton data directly from the videos, and then evaluated our models on the extracted features. Therefore, we firstly use OpenPose to estimate the 2D and 3D joint coordinates from the captured video provided by the UTD-MHAD dataset. The estimated 3D and 2D joint coordinates for one action are shown in graph (c) and graph (d) of Fig.5.8. As shown in Fig.5.8-(c), the estimated 3D joint coordinate is noisy and the classification accuracy is also not good, while the estimated 2D joint coordinate is much more accurate. Therefore, we use the estimated 2D joint coordinates as input for our model that were proposed in Chapter 4. Table 5.3 presents the result of the proposed approach and the related results reported in the literature which adopt the video as input. As can be seen, the extracted 2D skeleton with the model proposed in Chapter 4 improves the recognition accuracy

Table 5.3: Performance comparison of video-based approach on UTD-MHAD dataset

#	RGB Video
DMM-CRC [27]	66.1%
STIP-BOW-SVM [186]	67.37%
WHDMMs-ConvNets [187]	73.95%
RGB-baseline	38%
RGB-based-2d-skeleton	72.1%

significantly. As shown in Table 5.3, the RGB video-based approach cannot get a good result. We attribute the reason for this is because the video frames contains much redundant information, which degrades the performance of the classifier, and the limited training data. The state-of-the-art approach based on the RGB video usually needs a huge scale of training data and a big deep neural network to

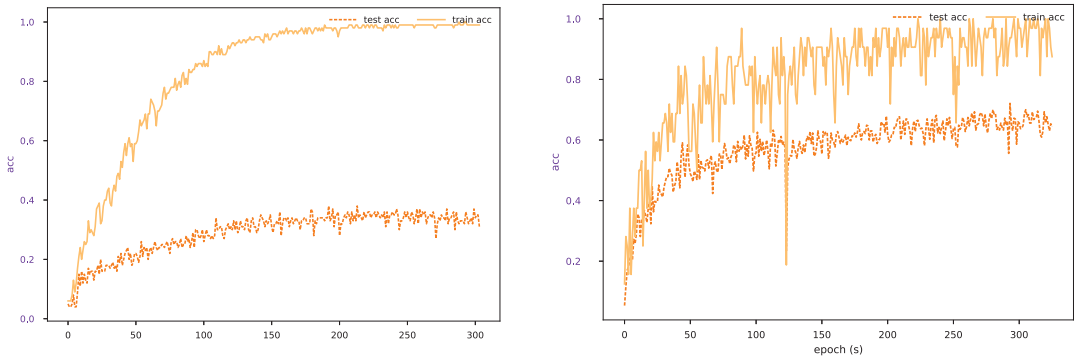


Figure 5.9: Accuracy of RGB video-based action recognition for UTD-MHAD dataset

model the evolution of the temporal features that existed in RGB videos. Because of this, there are few results on this dataset based on the RGB video data that are reported in the literature.

As the convergence rate curve of RGB video based training and estimated 2d skeleton training indicates in Fig.5.9 that the performance of the estimated 2d skeleton data significantly improved the accuracy from the 38% to 72% taken video as input features. The left graph of Fig.5.9 is the result of the video features as input, the right graph is produced by taking the estimated 2D skeleton from the video as the input features.

5.5 Summary and contributions

For this Chapter, there are three major contributions that can be identified as follows:

1) A novel framework is proposed by integrating the LSTM-C with the CNN-based spatial features for the video-based action recognition. The proposed model can achieve competitive results on the test datasets, although our model only utilizes the images based spatial features of the human actions at specific timestep, without optical flow and other hand-crafted features.

2) On the other hand, the LSTM-C cells can help to explore the most significant temporal features, thereby improving the holistic features representation for the whole video. The experimental results demonstrated that the incorporated attention mechanism can produce a superior performance compared to our baseline system. The findings of this Chapter have provided a glimpse for future research based on visual attention.

3) We find that the performance of action recognition based on video data is extremely

difficult to improve further, which will also be degraded significantly on large datasets. Extracting the skeleton data from the videos directly with the pose estimation algorithm for action recognition is one efficient approach for video based action recognition. This approach can reduce the computational cost significantly and improve the performance for video-based action recognition.

Chapter 6

Multi-stream CNN model for Human Action Recognition

From the previous experimental results, we can observe that it becomes more challenging to improve the performance of accuracy further with video as the input features, even with a more powerful model and the computation resources that is available. Even though the proposed hand-crafted features can improve the recognition accuracy on small human action recognition datasets, the performance will degrade significantly on large datasets. As we have observed in previous Chapters, the skeleton-based motion and energy features can also efficiently characterize human actions. Motivated by the most popular two-stream CNN model for video classification and the powerful feature extraction ability of the CNN model demonstrated in previous Chapters, we devised a multi-stream CNN-based framework to accommodate the skeleton-based motion and energy features for 3D skeleton-based action recognition in this Chapter.

6.1 Motivation

The ultimate objective for computer vision systems is to mimic the way of humans process and understand meaningful data from multiple features (some features have constraints between each other, e.g., images, video, audio) regarding a subject in action within a scene. For example, humans are capable of interpreting visual information using natural language systems. It is also possible to infer from a visual stimulus what is the implied intent is behind the human actions that are present in the video. A huge amount of work has been conducted to achieve this goal including object detection/recognition (equivalent to identifying the

noun descriptor), action recognition (equivalent to identifying a verb descriptor) and attribute learning (equivalent to identifying an adjective or adverb descriptor). All of these research initiatives aim at providing more accurate and more sensitive semantic information for human action understanding. Among all of these tasks, action recognition is essential to develop more complicated systems since it serves as the foundation for capturing the context of what is happening in a scene, as well as to provide some clues on other more difficult classification tasks, moving from coarse-grained to fine-grained classification, such as recognizing emotions and interactive actions. The successful two-stream model takes advantage of the spatial features and temporal features contained in the videos, which can merge both the spatial and temporal features efficiently. As we have proposed, the motion and energy features based on the skeleton sequence in the previous Chapters, we attempt to fuse the skeleton and their derived motion and energy features in this Chapter by proposing a multi-stream CNN model.

6.2 Related work

The most popular Recurrent Neural Network (RNN) was proposed for the time-series problem, which can model the temporal dependency of the sequential signals. Thus, it is a natural choice for 3D skeleton-based action recognition to extract global features for one action sequence. However, based on a review of the literature more and more of the research has adopted the CNN model to learn the skeleton features and it has achieved an impressive performance in recent years. Previous research has made great progress in action recognition and many excellent models have been proposed, and different test strategies have been explored. Some works extract different features from the video frames and then fuse them for classification. The RNN model is also widely used to fuse the extracted features together [141], and various pooling methods have been developed [188]. CNN gained great success in image processing. Among many of the CNN models used in the field of video processing, the 3D-CNN [173] is efficient in extracting the spatio-temporal features from the RGB video. Other methods capitalise on the temporal information including the optical flow, trajectories, and the human pose estimation. While all of these methods do a good job in action recognition based on video, the universal limitation of these methods is that they are

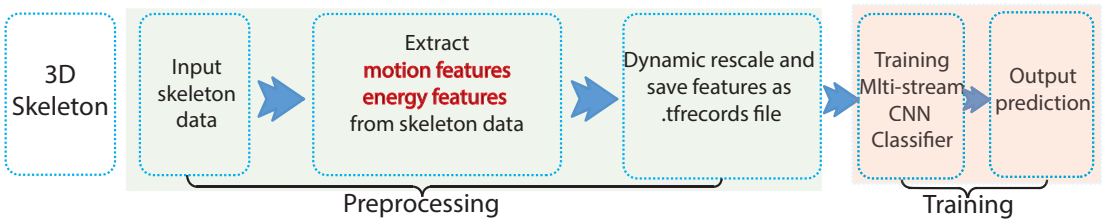


Figure 6.1: Workflow of the proposed Multi-stream CNN model

expensive in terms of computations. However, using the multiple skeleton-based features for action recognition has not gained too much attention, especially with the low resource of the training data, in the research community, therefore, we explore the performance of the fusion with different skeleton-based geometric features as input in this Chapter.

Although various models have been extensively researched in the past years, action recognition in realistic application is still challenging, because of the viewpoint changes, occlusions, the complex environment background and similar actions. In this chapter, aiming at improving the performance of deep learning systems for action recognition tasks, in this Chapter, we utilize a novel hybrid model that is able to employ a set of multiple skeletal features and test it on three challenging benchmarking datasets. We believe that learning more compact and meaningful representations from a set of multiple inputs will be one of the most effective tools to further advance the topic of action recognition.

6.3 The algorithm

6.3.1 General architecture

As stated in previous Chapters, the input action-related features are the key for accurate action recognition and subsequent applications. Although our previous approach can work well based on the different features independently or the merged features, it is not easy to improve the performance further and the performance in complex scenes will be degraded to some extent. In order to derive more complex semantics surrounding the action recognition (e.g., to understand the intent behind the actions), more sensitive semantics should be learned from a set of multiple features. In this Chapter, we explore combining the skeleton-based feature for action recognition, and formulating a multi-stream model, which

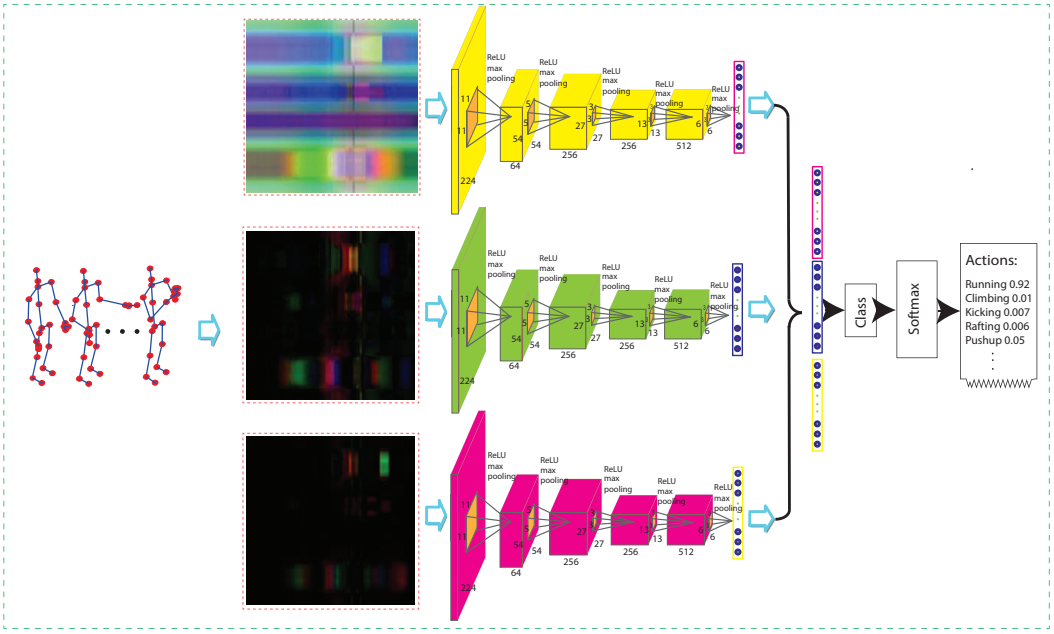


Figure 6.2: Overview of the proposed multi-stream CNN model for fusing geometric and kinematic features. The input streams are combinations of the aforementioned features, such as "joint coordinates", "motion features" and "energy features".

can extract richer information for action recognition from the different combinations of the input features. The workflow is shown in Fig.6.1.

```
Train_path, Test_path
# Loading geometric relational features
train_data, test_data, label = loading_data(Train_path, Test_path)

for i in range(epoch):
   label, Feat, Feat_M, Feat_E = Load_training(batch_size,train_data)

   # use multi cnn model to extract features
   Feat1, Feat2, Feat3  = model(Feat), model(Feat_M), model(Feat_E)

   # Maximum, Average, Sum, Multiply
   out_feat = Maximum(Feat1, Feat2, Feat3 )
   prediction = Dense(num_class)(out_feat)
   loss = cross_entropy(label, prediction)

   # update parameters
   los_final.backward()
   accuracy = reduce_mean(prediction == label)

   if i % 5 == 0:
      label, Feat, Feat_M, Feat_E = Load_testing(batch_size,test_data)
      loss, accuracy = test(label, Feat, Feat_M, Feat_E)
```

Code block 8: *Training and testing pseudocode of Multi-CNN model*

The proposed model can be illustrated as Fig. 6.2. This multi-stream model can take multiple features as input, and the extracted features of each single stream CNN model are flattened and merged to represent the input of a subsequent fully-connected neural network for classification. The pseudocode for feature extraction is presented in Chapter 4. The pseudocode for training and testing of the proposed framework is shown in code block 5.

6.3.2 Multi-stream CNN model

In the previous Chapters, we designed several geometric features to characterize the human actions based on the acquired joint coordinates, each feature can characterize the relative position of each pose independently. While the Recurrent Neural Networks are also designed to extract this kind of high-level features from the input joint coordinates or the input geometric features as described in Chapter 4, nevertheless it is not easy to train and they cannot extract effective features from multiple features at the same time. In order to combine these features efficiently and extract the most effective features from the input streams, in this Chapter we attempt to use a multi-stream CNN model to fuse multiple features with different fusing strategies. Different from the most popular two-stream model, which is based on videos, our multi-stream model is based on the skeletal features. The skeleton sequence provides more compact and accurate information to describe human actions. And the skeleton data does not contain the noisy background information and it is extremely small in size compared with the RGB video data.

The interaction between the different geometric features and the combination of different geometric features is important to characterize a human action. In Chapter 3 we used one stream CNN model to extract the high-level spatio-temporal information from the skeleton-based static images. This approach converts the action recognition problem into an image classification problem, which unavoidably will lose some information once we resize all of the concatenated geometric features into a fixed size of static image. In terms of the RNN-based model introduced in Chapter 4, with our proposed spatio-temporal kernel, even though it can improve the spatial features extraction ability of the model, it is insufficient for complex actions. Therefore, in this Chapter we attempt to propose a multi-stream CNN framework to extract the discriminative representation taking the multiple features as input. Specifically, the proposed architecture for a single person is shown in Fig.6.3, which can be

extended into a more general architecture to accommodate the actions performed by multiple subjects.

Specifically, the skeleton sequence of one person is represented as X, and the various geometric features can be derived from the sequence data X by following the steps introduced in Chapter 4. Then, the extracted skeleton-based geometric features can be fed into the multi-stream CNN model independently. In order to fuse the information from the multiple sources, we fuse their features maps across the output channels in the subsequent layers of the network (shown in Fig.6.3), and the fuse operation is based on the flattened features that

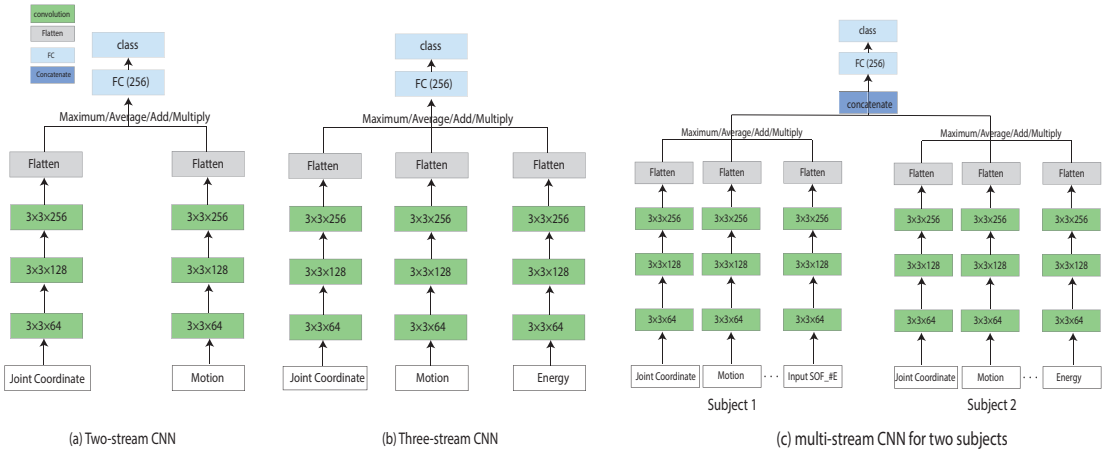


Figure 6.3: Variants of proposed multi-stream CNN model for fusing geometric features

were extracted via the CNN models. The parameters of each stream are learned independently, and the extracted features of each stream are fused by the concatenation along the output channels after the third convolution layer. Compared with the model that was used in Chapter 3 the proposed model contains much less trainable parameters. The lightweight model allows us to train the model from scratch so it can achieve super performance on the low-resource UTD-MHAD dataset.

6.3.3 Interaction actions

In most activities, e.g., shaking hands, hugging, kissing, multiple persons are involved, so we make our model scalable to multi-person scenarios, which is promising for group activity recognition. To make our proposed model scalable to group activities, we perform a comprehensive evaluation on our proposed features and feature fusion strategies, including

the average, maximum, sum and multiply fuse strategy. In order to discover the co-occurrence features in the interaction actions, which are performed by multiple subjects, we tested our model on the SBU-Kinect dataset and the NTU RGB+D dataset. Different from experiments on actions that involve only one subject, we fed the coordinates and motion features of each subject into the one-stream CNN model separately. As illustrated in Fig.6.3-b, different features go through a separate sub-network and their Conv3 feature maps are merged with an element-wise maximum, mean, sum or multiply fusion strategy. It is worthwhile mentioning that the element-wise fusion method can generalize well to actions that are performed by a variable number of persons. This feature fusion strategy is promising for group activity recognition.

6.4 Empirical testing and analysis

6.4.1 Dataset

We evaluated our proposed framework on three benchmark datasets, namely, the SBU-Kinect, the UTD-MHAD and the NTU RGB+D dataset.

6.4.1.1 SBU-Kinect Interaction Dataset

The SBU-Kinect Interaction dataset [159] depicts the interaction between two actors. There are 15 joints for each actor in every skeleton frame that captured by the Kinect camera. For the evaluation protocol, we followed the evaluation strategy proposed in [159], performing subject-independent 5-fold cross-validation.

6.4.1.2 UTD-MHAD Dataset

As described before, this is a multimodal dataset for action recognition, providing skeleton joint positions, inertial sensor signals, RGB video and depth video. This dataset includes 861 video instances for 27 actions performed by eight subjects. This dataset was chosen because of its difficulty and accuracy gap compared with the other action recognition dataset. In all of our experiments, 861 sequences of UTD-MHAD datasets were investigated, half of sequences were used for training and the rest sequences of the dataset were used for testing. This is a

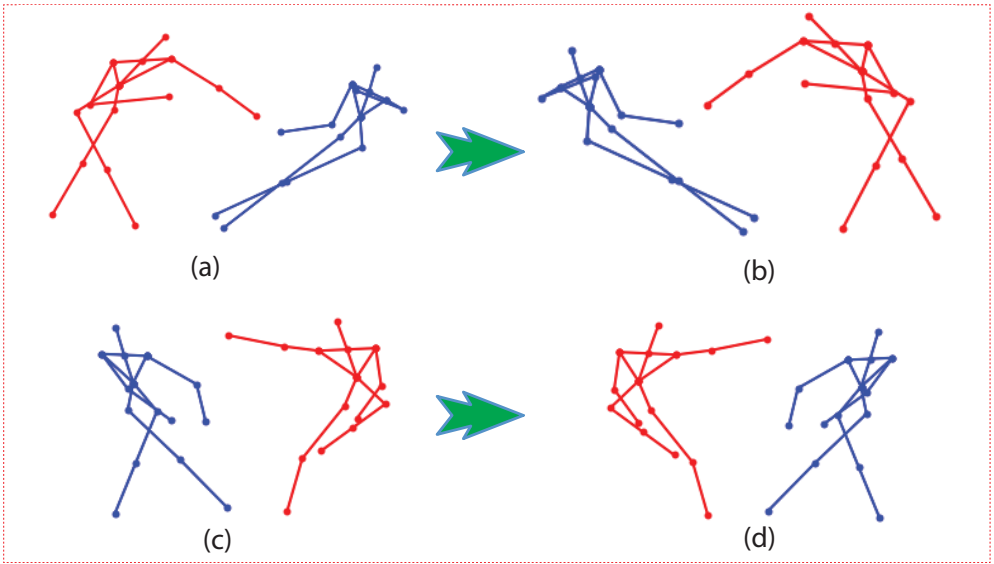


Figure 6.4: Visualization for two actions with swapping the positions of two subjects

challenging dataset, because the evaluation protocol on this dataset is using a half-to-half training and testing strategy, training with low resource data is one most challenging topic for deep learning, which is known as a data-hungry approach.

6.4.1.3 NTU RGB+D Dataset

As introduced in Chapter 3, the NTU RGB+D dataset is so far the latest and the most challenging dataset for skeleton-based human action recognition. This database contains more than 56000 action instances and about 4 million frames in total for 60 actions. These action instances are performed by 40 different subjects, and the actions are recorded from 3 views. Due to the changing viewpoints, intra-class and length variations, this dataset is a very challenging action recognition dataset. More details about this dataset can be referred to previous chapters.

6.4.2 Results of SBU-Kinect Interaction database

The previously proposed features work well on UT-Kinect dataset by using our proposed SKB-TCN model, because the data is captured in a controlled environment and the actions are relatively simple and are performed by one person. The performance of the proposed feature degrades once we scale the same model to complex actions, such as interaction

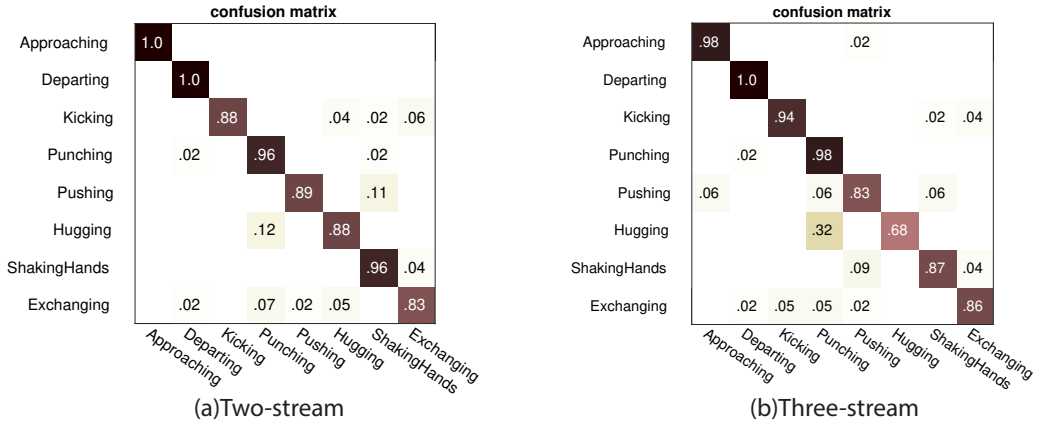
actions, and actions with various backgrounds. Therefore, for the SBU-Kinect dataset, we only use the motion features and the original coordinates as input, and introduced some extra data augmentation techniques in this experiment. **We augment the training data via mirroring the positions of the two actors randomly, and we dynamically normalize the sequence length.** Two of the action examples with mirrored position of the two subjects are shown in Fig.6.4, this data augmentation technique mimics the view-point changing situation. Motivated by this, we will utilize the neural network to learn the transformation to augment the training dataset in the next Chapter.

Table 6.1: Performance comparison of Multi-CNN and related models on SBU-Kinect

#	RGB Video
Yun et al. [159]	80.3%
CHARM [163]	83.9%
HBRNN-L [141]	80.35%
Co-occurrence LSTM [154]	90.41%
ST-LSTM [153]	93.3%
Two-Stream CNN	Ave: 93.3%, Std: 0.013
Three-Stream CNN	Ave: 91.52%, Std:0.016

We followed the setup proposed in [154], and the result details are listed in Table 6.1. For data augmentation, we dynamically rescale the sequence length to 16. We compared our proposed model with [154], significant improvement on the recognition accuracy for this database can be identified from Table 6.1. With the maximum fusion strategy, our model achieves accuracy of 93.3%, which is a competitive result compared to the state-of-the-art model [153]. The standard deviation of our result for 5-fold validation is 0.013, which proves that our model is stable and robust. Other fusion strategies, e.g., average, sum and multiply can also achieve very close results compared with the maximum fusion strategy.

The superior advantage of our proposed framework is that it is able to address multiple person's action recognition in a flexible manner, which is an efficient approach to explore the relationship between our proposed geometric features of a single person or multiple subjects simultaneously. Because this database is performed by two people, in order to analyse the



(a)Two-stream (b)Three-stream

Figure 6.5: Confusion matrix of five-fold testing with multi-stream CNN model

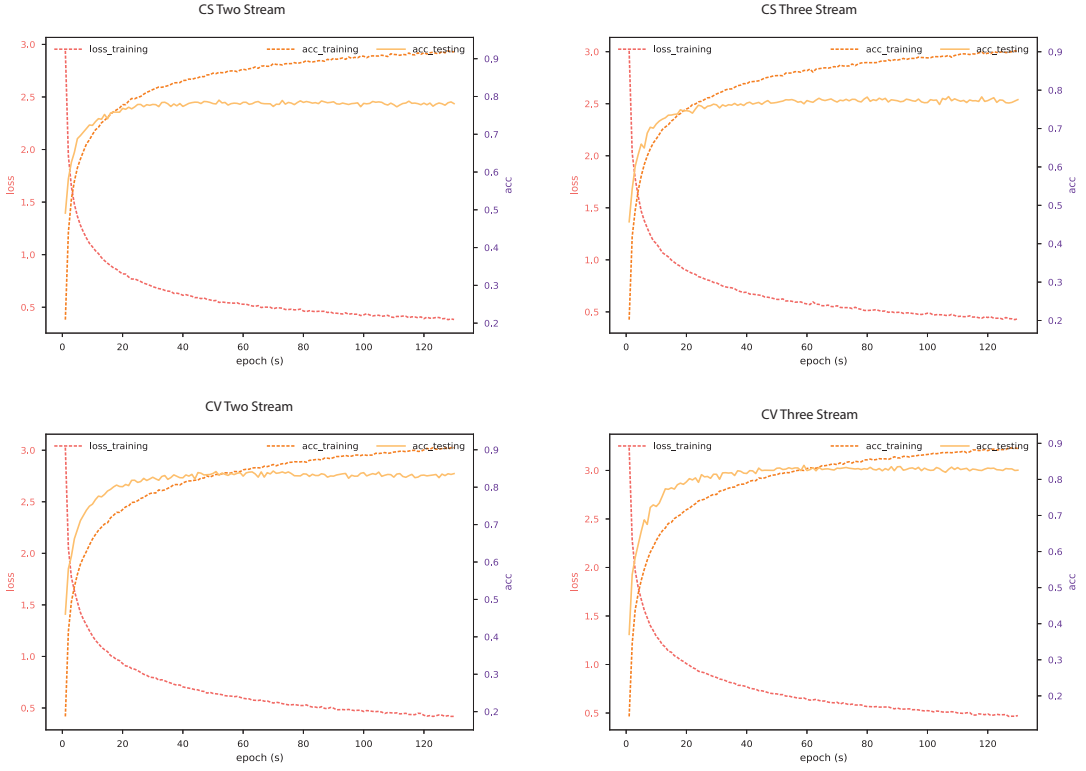


Figure 6.6: Convergence curve of Multi-stream CNN model on NTU RGB+D

pose relationship between these two people while they perform different actions, we attempt to feed the features of different people into a separate stream and fuse the extracted features in the top layer. The confusion matrix of the five-fold testing result is shown in Fig.6.5 above.

6.4.3 Results of NTU RGB+D database

In order to extend this idea to a large scale action recognition application, we evaluated our model on the latest challenging action recognition dataset, NTU RGB+D. Table 6.2 presents the result of our model and the results of similar models. As shown in this table, better results can be achieved than Chapter 3. From the experimental results, we can see the two-stream model usually achieves better performance than the three-stream model on this database. It seems that the derived energy features cannot augment the discriminative ability of the learned representation, while introduce some redundant information to the representation that learned from the joint coordinates and motion features. The proposed model works well on this dataset, and can achieve competitive results compared with the state-of-the-arts results. The convergence curves for both the Cross-View and the Cross-Subject evaluation strategies are demonstrated in Fig.6.6 below.

Table 6.2: Performance comparison of Multi-CNN and related models on NTU RGB+D.

	CS		CV	
HBRNN [20]	59.1%		64.0%	
LSTM [189]	70.3%		82.4%	
CNN [139]	76.0%		82.6%	
TS-LSTM [157]	74.6%		81.3%	
	two-stream	three-stream	two-stream	three-stream
Sum	79.0%	78.3%	84.3%	83.9%
Multiply	77.0%	75.9%	82.0%	78.8%
Average	**79.6%**	**78.9%**	84.5%	83.8%
Maximum	78.7%	78.7%	**84.7%**	**84.1%**

6.4.4 Results of UTD MHAD database

In order to discover the potential of the geometric features, we extensively tested some possible combinations of the proposed features on the UTD-MHAD dataset. The result of each feature (mono-feature) is shown in Table 6.3.

6.4.4.1 Mono-SOFs feature based multi-stream model

For multi-stream CNN model, we can fuse features extracted by a different CNN model via different strategies, such as Maximum, Sum, Average and Multiply. The results of our proposed multi-stream model with maximum fusion strategy on the UTD-MHAD dataset are shown in Table 6.3.

6.4.4.2 Dual-SOFs feature based multi-stream model

For the purpose and with the goal to discover the co-occurrence relationship between the different features, we fed multiple features into our proposed model. We have 8 skeletal

Table 6.3: Performance comparison of Multi-CNN and related models on UTD-MHAD.

Deep RNN [20]		66.10%		
Cov3DJ [138]		85.58%		
CNN + JTM [135]		85.81%		
CNN + SOS [136]		86.97%		
3D-HoTMBC [137]		84.40%		
CNN + JDM [132]		88.10%		
Fusion approach	maximum	sum	average	multiply
SOF1	80.23%	71.16%	73.49%	78.37%
SOF2	84.18%	83.95%	82.55%	83.72%
SOF3	87.90%	83.02%	84.88%	87.21%
SOF4	84.65%	79.77%	81.39%	65.12%
SOF5	76.97%	76.74%	77.91%	73.49%
SOF6	76.27%	71.63%	76.04%	74.65%
SOF7	68.60%	63.49%	74.65%	64.65%
SOF8	60.00%	54.88%	57.20%	58.83%

geometric relational features, so there are 28 possible combinations. The results of these 28 combinations are listed in Table 6.4. The combination of SOF2 and SOF3 can obtain the state-of-the-art recognition accuracy with a Maximum fusion strategy on this dataset. Fig.6.7 shows the convergence curve of four different fusion strategies with the SOF2 and SOF3 as input features. The output results demonstrate that all of the proposed features can describe the human actions efficiently on this database with the multi-stream CNN model as the classifier, and each of the features can work together with the other features.

6.5 Summary and contributions

In this Chapter, a multi-stream framework is adopted to fuse the proposed motion and energy features. Furthermore, we extend our framework to deal with multi-person involved activities.

Table 6.4: Results of different feature combinations on UTD-MHAD

	SOF1	SOF2	SOF3	SOF4	SOF5	SOF6	SOF7	SOF8	SOF1	SOF2	SOF3	SOF4	SOF5	SOF6	SOF7	SOF8
	Average Fusion								Maximum Fusion							
SOF1	-	85.12%	84.18%	81.16%	79.77%	82.56%	75.35%	73.26%		86.51%	85.34%	81.39%	81.62%	80.93%	75.11%	76.05%
SOF2	-	-	87.44%	84.42%	85.58%	78.13%	80.47%	80.23%	-	-	90.47%	84.88%	84.65%	86.05%	81.86%	79.07%
SOF3	-	-	-	83.72%	88.14%	85.11%	82.09%	82.33%	-	-	-	86.51%	87.20%	83.95%	84.41%	84.88%
SOF4	-	-	-	-	80.0%	78.37%	77.67%	81.40%	-	-	-	-	84.42%	80.70%	79.77%	82.33%
SOF5	-	-	-	-	-	82.79%	76.51%	78.37%	-	-	-	-	-	83.02%	79.07%	76.04%
SOF6	-	-	-	-	-	-	78.37%	78.37%	-	-	-	-	-	-	81.16%	76.04%
SOF7	-	-	-	-	-	-	-	78.37%	-	-	-	-	-	-	-	76.04%
SOF8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Multiply Fusion								Sum Fusion							
SOF1	-	85.12%	79.53%	76.28%	67.67%	72.33%	64.65%	59.30%	-	86.27%	84.88%	83.95%	84.65%	85.11%	70.23%	74.41%
SOF2	-	-	86.05%	83.25%	84.65%	82.09%	83.95%	83.95%	-	-	89.30%	83.26%	85.12%	83.72%	82.79%	82.79%
SOF3	-	-	-	72.56%	82.79%	80.47%	81.86%	82.33%	-	-	-	88.84%	89.30%	83.26%	84.88%	88.37%
SOF4	-	-	-	-	74.19%	71.40%	64.19%	68.37%	-	-	-	-	84.19%	78.84%	78.13%	83.26%
SOF5	-	-	-	-	-	83.95%	70.70%	71.16%	-	-	-	-	-	82.09%	78.37%	78.60%
SOF6	-	-	-	-	-	-	72.33%	74.65%	-	-	-	-	-	-	78.83%	78.60%
SOF7	-	-	-	-	-	-	-	71.16%	-	-	-	-	-	-	-	78.60%
SOF8	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Following major contributions can be identified from this Chapter:

1) An efficient approach for fusing multiple features is proposed for skeleton-based action recognition. The proposed multi-stream CNN model is adopted to extract discriminative representation from the skeleton and its derived motion and energy features, which shows better performance compared to other CNN based models.

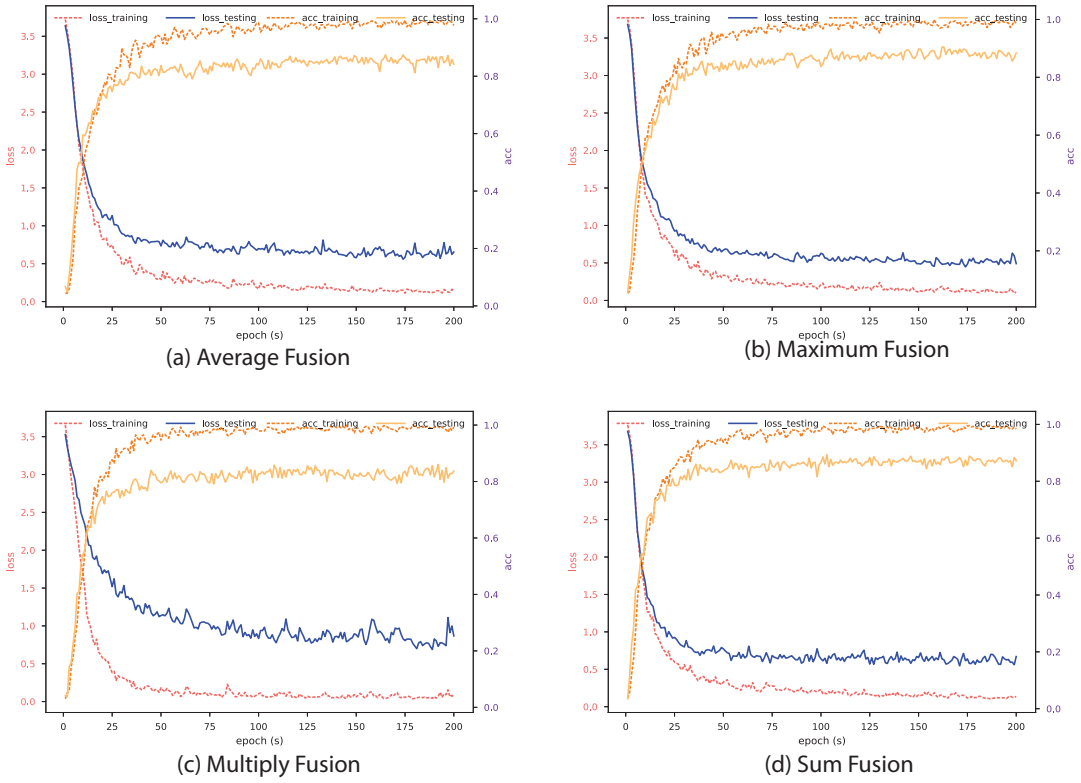


Figure 6.7: Convergence rate curves for different fusion strategies on UTD-MHAD (SOF2+SOF3)

2) We devised a novel element-wise features fusion strategy, which can extract and fuse the extracted features efficiently. This fusion strategy is better than the other approaches, such as concatenation fusion strategy, that needs to consider the number of people involved in the actions beforehand.

3) The proposed framework presents a better performance on the interaction action recognition dataset, SBU-Kinect interaction dataset. This means that the multi-CNN model and the element-wise fusion strategy are efficient for group action recognition, which involves multiple people in one action.

Chapter 7

Human Action Recognition based on Skeleton Transformer Network

As we found in Chapter 4, data augmentation is one of the most crucial factors for the training of the RNN-based model taking the skeleton-based geometric and kinematic features as input. Even though the approaches proposed in Chapter 4 perform very well on small datasets, but it is hard to scale these approaches to large-scale datasets, such as NTU RGB+D, because of the significant cost in terms of storage and computation. Also, the geometric features' representation ability is insufficient for large scale of actions in complex scenes. In this Chapter, we customized a skeleton transformer based on the LSTM network to transform the input sequences. This improved the robustness of the trained model on the testing dataset.

7.1 Motivation

Recognizing human actions based on the 3D skeleton data, commonly referred to as 3D action recognition, is fast gaining interest from the scientific community recently, because this approach presents a robust, compact and a perspective-invariant representation of motion data. In the recent literature, the common ideas for all skeleton-based action recognition approaches are attempting to generate a robust spatio-temporal representation for the original skeleton data and make the trained model generalize well on the testing dataset. In line with this research, there are various advanced approaches that have been proposed [22], [147], [190] to extract the complex spatial and temporal relationships from the raw skeleton coordinates. Despite the significant progress that has been achieved by

introducing the novel hand-crafted spatial and temporal features in the previous Chapters, the limitation of existing hand-crafted features in the spatial and temporal domain is that their representation ability is not sufficient for the complex actions and cannot meet the demand in the real-world application. With the normalization operation in the preprocess phase, the existing temporal features contain some redundant and irrelevant information in the temporal domain. So there is an open question that remains in terms of the best way to learn a more robust and effective representation in order to eliminate these irrelevant and redundant temporal information. And using the RNN model to learn these features automatically will be a future research focus. Motivated by these facts, we attempt to propose a framework that can consider both the local and global information contained in an input sequence to refine the input sequence. The proposed model can generally consider the local and global information for action recognition, but cannot identify the local and global features accurately. Identifying the local and global features from the input skeleton sequence will be our future research topic.

7.2 Related work

Relevant recent attempts that aim at solving the 3D human action recognition problem [20], [141], [191] have shown encouraging performance, and provided evidence of the efficacy of RNNs in modeling the complex dynamics of human actions in the temporal domain. The main focus of the previous existing models was to utilize RNNs over temporal domains to discover the dynamic motion patterns for 3D action recognition globally. However, all of these works used hand-crafted rules to extract the discriminative information in static postures in each individual frame and ignored the temporal features in a the short term, which we call local features, and temporal features in the long term, which we call global features.

The training of a robust deep learning model usually has a high demand for the training data, which requires the training data to cover more patterns that appeared in the testing dataset, to improve the robustness on the test dataset and avoid the underfitting or overfitting problem. Data augmentation is one of the most effective approaches to solve the low-resource training problem and improve the generalization of the model. Various approaches for augmenting the

training data have been reported in the literature, but most of them are proposed for image classification. [191] proposed to use angle rotation, scaling and shear transformation. [192] employed a randomly cropping approach and randomly horizontally flip the converted static image as a data augmentation approach to increase the robustness of the input sequence. [193] devised a novel data augmentation approach, which rotates the 3D coordinate randomly and adds some Gaussian noise to augment the amount of the training dataset.

However, most of the existing approaches for training data augmentation are based on some predefined transformations in the spatial domain, and we extended the data augmentation to the temporal domain in Chapter 3 and Chapter 4. All of these methods are restricted by either a low representation ability or an expensive computational cost. For example, while the method of horizontal flip approach neglects the temporal dependence between the sequential frames, a randomly cropping approach will lose some of the spatial information. All of these operations that have been adopted in the preprocess phase will introduce some interference information that is unhelpful to the classification to enlarge the training dataset. Motivated by the aforementioned works, in this Chapter, we propose a novel Skeleton Transformer Network (STN) for the 3D skeleton-based action recognition, which utilizes advanced deep learning techniques to learn useful representation for classification so as to augment the training dataset by itself. Also, due to the proposed loss function makes the transformed sequence shares similar characteristics with the original input sequential features. In another words, the transformed sequence is a blend of the input sequences. Each frame of the transformed sequence is a combination of the local and global information that is contained in the input sequence, which can efficiently handle the variation in the execution of an action. With this proposed framework, the output model can be robust to counter the noisy input data, and this proposed approach can easily be extended to the other applications to process the input sequential data.

7.3 The algorithm

7.3.1 General architecture

The proposed framework for STN is shown in Fig. 7.1, it consists of two streams, one of which accepts the original input sequence (raw joint coordinates) directly and another one includes the sequential transformer network, which accepts the original input and then outputs the transformed sequence of the input feature. The transformed sequence will be fed into the subsequent classifier.

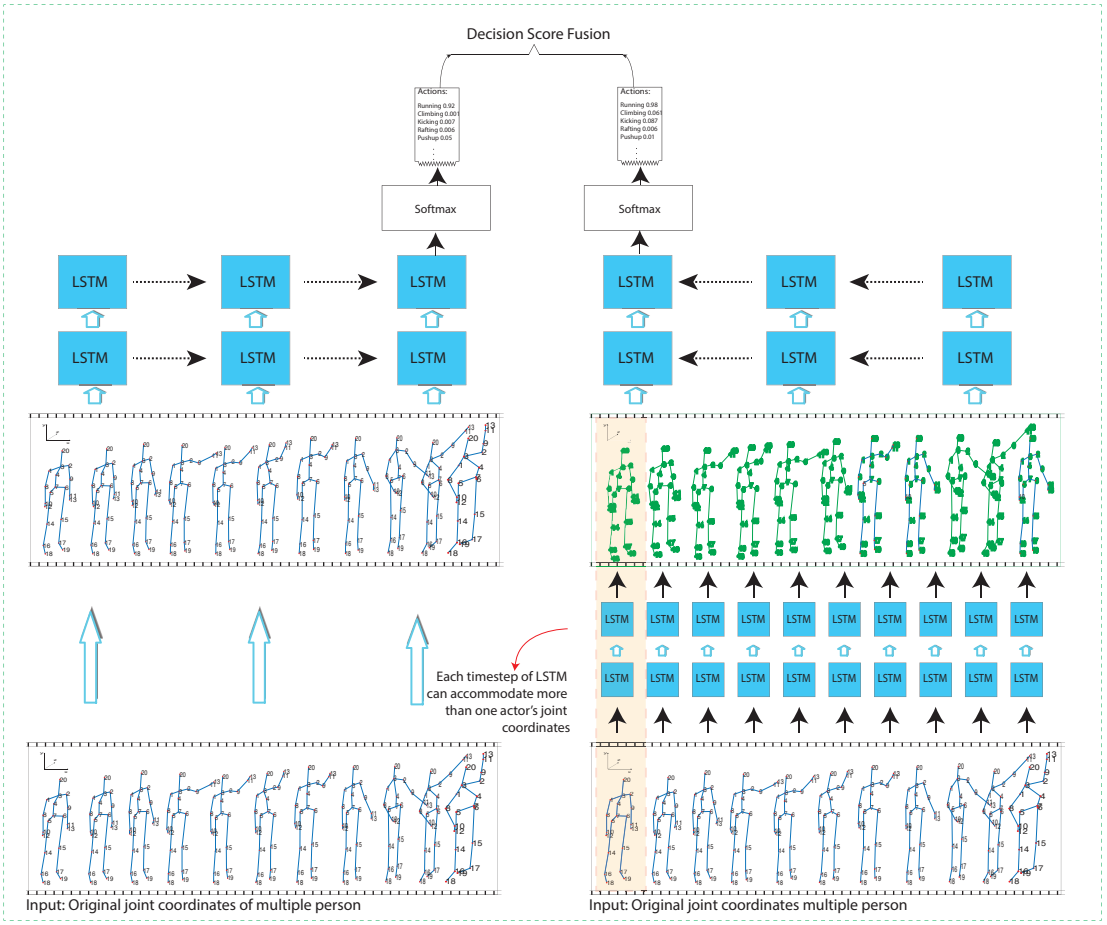


Figure 7.1: Framework of the STN model

It is worth mentioning that the proposed model is easy to be extended for group activity recognition. For example, as mentioned before, the interaction actions are usually performed by more than one person, the integration of all of these involved actors can be inputted into

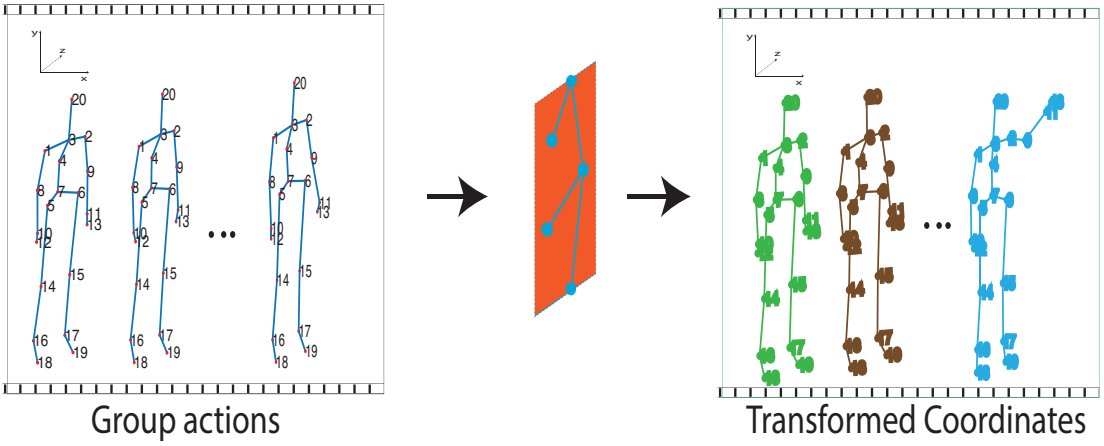


Figure 7.2: Variant of STN model to accommodate actions performed by multiple actors. The transformed coordinates are hypothetical examples, a visualisation of the actual results are provided in Fig.7.9

```
Train_path, Test_path # path of the training and testing data

skeleton = iterate(Train_path, Test_path)
N = number of training and testing examples
# preprocessing
for i in range(N):
   skeleton, label = read(skeleton[i])
   normalized_skeleton = norm(skeleton)
   save(normalized_skeleton)
# Loading skeleton data
train_data, test_data, label = loading_data(Train_path, Test_path)

# Training and testing
for i in range(epoch):
   # Define the proposed model:
   label, input = Load_training(batch_size,train_data)
   out_ori = multi_layer_LSTM(input)
   out_trans = multi_layer_LSTM(transform(input))
   final_out = merge(out_ori, out_trans)
   loss_ori,  loss_trans = cross_entropy(label, out_ori, out_trans)
   loss_mse = MSE(transform(input), input)
   loss_final = loss_ori + loss_trans + loss_mse
   # update parameters
   loss_final.backward()

   accuracy = reduce_mean(out_original equal label)
   if i % 5 = 0:
      label, features = Load_testing(batch_size,test_data)
      loss, accuracy = test(label, features)
```

Code block 9: *Training and testing pseudocode of STN model*

our transformer network for preprocessing, and then input the transformed coordinates into the subsequent classifier. This process can be illustrated by Fig. 7.2. The pseudocode for this framework is shown in code block 6.

7.3.2 Skeleton transformer network

As illustrated in Chapter 4, the RNN model excels in modelling the temporal relationship contained in the time series sequence and extracting the global representation for the whole sequence which can preserve the temporal information in the extracted global representation. The Vanilla RNN model cannot model long-term dependency very well because of the gradient vanishing and error blowing up problem. Therefore, we adopt the advanced RNN architecture LSTM [18] as the backbone model for our framework. The LSTM neural includes three gates and one cell memory unit. The gates can control the information to pass or forget at each timestep, the details of the information flow are stated in Chapter 4.

The LSTM unit is utilized to build the STN, as shown in the right part of Fig.7.1, which accepts the original skeleton data and outputs a transformed version of the input skeleton data. This transforming process can be treated as a transformation operation on the input joint coordinates, such as the angle rotation operation, but it is different from the traditional transformation operation, because it is learned by the deep learning framework. For example, one input skeleton sequence X as input to the sequential transformer network, and we can get the transformed version of the input sequence X'. Then the processed sequence X' and the original skeleton sequence X are then fed into the subsequent multiple layers of action classifier in parallel. The output of action classifier for the original skeleton sequence X and transformed X' can be indicated as $F(X)$ and $F(X')$, then we fuse these two outputs as the final output of the model with a maximum or an average strategy. Even though the structure of our proposed model is similar to the model that was proposed in [105], but their model is designed for video classification and they did not feed the original input into the classifier.

For the skeleton transform stream, it consists of two transform modules with symmetric architecture. One is used to compress the dimension of the raw input and the other one is used to output a transformed vector, that has same dimension with the input sequence. The STN model first compresses the original input into a subspace and then recovers the compressed representation into the original space. This process is learned by the training of

the whole framework in an end-to-end learning manner, and the transformation capability of the network determines the generalization ability of the whole model. So, in order to train this model well for the spatial-temporal-structural information extraction, we devised a new loss function, which will be described in the following sections. This loss function considers both the loss of the final classification and the transform loss into consideration, which helps to train the whole framework in an end-to-end manner.

7.3.3 Training optimization

In order to train the proposed sequential transformer network to extract more robust and proper representations from the input skeleton sequences, we customized a loss function based on the widely used Cross-Entropy loss function. Specifically, for one original input sequence X, the corresponding label is y; the transformed sequence of X can be dedicated as X'; the loss function can be formulated as the following formula:

$$\mathcal{L} = L_{original}(\theta) + L_{transformed}(\theta) + \lambda \|X - X'\|^2 \tag{7.1}$$

where $\mathcal{L}$ consists of three parts, the original classification loss $L_{original}(\theta)$, the transformed skeleton classification loss $L_{transformed}(\theta)$, the mean squared loss between the transformed sequence and the original skeleton sequence $\|X - X'\|_2$. The first two items of the loss function are the cross-entropy loss, which can be calculated by following two formulae:

$$L_{original}(\hat{y}^i_{original}, y_i) = -\sum_{i=1}^{C} y_i \log \hat{y}^i_{original} \tag{7.2}$$

$$L_{original}(\hat{y}^i_{transformed}, y_i) = -\sum_{i=1}^{C} y_i \log \hat{y}^i_{transformed} \tag{7.3}$$

where the $y_i = (y_1, y_2, ..., y_C)$ represents the ground truth label for the i_{th} sequence skeleton data; the $\hat{y}^i_{original}$ and $\hat{y}^i_{transformed}$ represents the prediction of the original input sequence and the prediction of the transformed sequence. The third part of the loss function is the transformed loss between the original input sequence and the transformed sequence. The λ is a scalar, which determines the significance of the transformed loss.

7.4 Empirical testing and analysis

7.4.1 Dataset

In this Chapter, we verified the proposed algorithm on three 3D benchmark databases, e.g., the UTD-MHAD dataset, the Northerwestern-UCLA dataset and the NTU RGB+D dataset, which are described in previous Chapters. The first two datasets have limited training data. In order not to repeat, we have only described some of the preprocessing phases on these datasets in this Chapter. For these three datasets, we adopted similar preprocessing steps, except that for the NTU-RGB+D dataset, because the skeleton sequence contains multiple persons' joint coordinates. The main issue that affects the performance of the proposed model is that it is expecting an input with 150 dimensions to accommodate the joint coordinates of two persons within a frame. However, in the dataset there are cases where there is only one actor within a frame. This means that only half of the 150 features are available. Therefore it could throw the algorithm off.

In order to process these sequences in a batch style during the training phase, we tested the following two different preprocessing approaches: 1) copy the joint coordinates of the first subject as the joint coordinates of the second subject; 2) keep the joint coordinates of the second subject as blank if there is only one subject involved in the target actions. With these two preprocessing approaches, the output skeleton features for the NTU-RGB+D dataset have 150 dimensions for each frame. This is different from our approach in Chapter 3, where we only kept the skeleton coordinates of one person, because if we keep the joint coordinates of the second subject as blank, the converted image will have large patches of black areas.

Table 7.1: Performance comparison of STN and related models on UTD-MHAD.

Model	Accuracy%		
Deep RNN [20]	66.10%		
Kinect+Inertial [27]	79.10%		
3D-HoTMBC [137]	84.40%		
CNN + JDM [132]	88.10%		
CNN + JTM [135]	85.81%		
CNN + SOS [136]	86.97%		
	transformed	original	fusion
STN	87.44%	85.11%	86.51%

7.4.2 Results of UTD MHAD database

Table 7.1 listed the results for our proposed model and the results of several other related models on the UTD-MHAD. Compared with the other existing methods, our proposed approach is able to achieve competitive results on this dataset. The final classification score produced by the three strategies, e.g. the transformed, the original, and the fusion, demonstrate different results. And it is worth mentioning that the fusion strategy cannot outperform the other two strategies, in a realistic application, we can select the maximum confidence score as the final result.

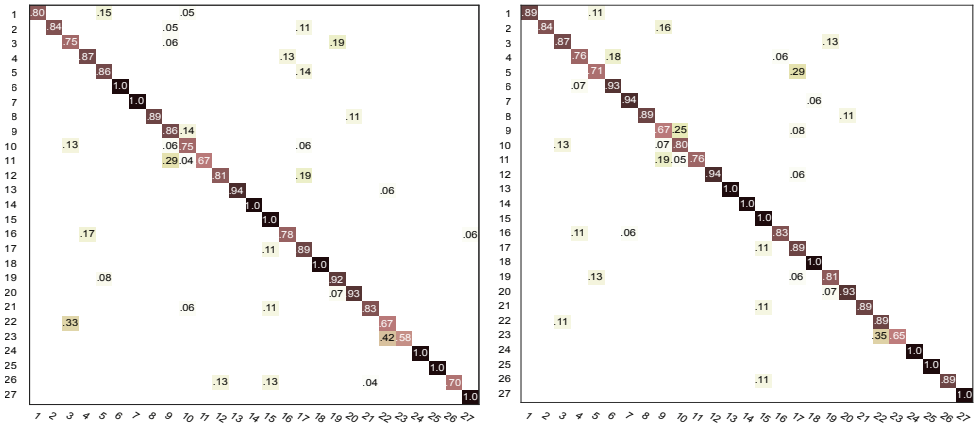


Figure 7.3: Confusion matrix of STN on UTD-MHAD

The convergence rate curve of the training and testing on this database is shown in Fig.7.4. It can be observed that both the training and the testing accuracy increase steadily, and the loss for both the training and testing decrease consistently.

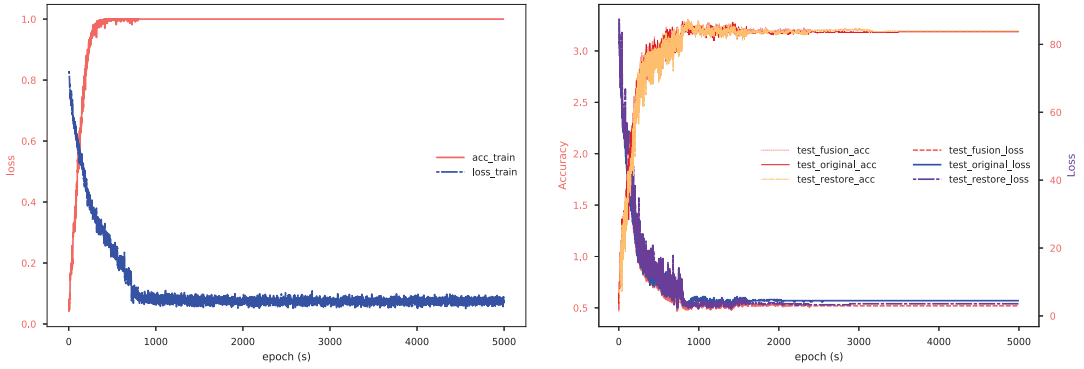


Figure 7.4: Convergence curve of STN model on UTD-MHAD

7.4.3 Results of Northwestern UCLA database

The best result of our proposed STN model on the Northwestern UCLA dataset is shown in Table 7.2. In terms of the better recognition accuracy of our model, it can be identified compared with other existing methods, specifically, the RNN based baseline model, e.g., the HBRNN-L and the EnTS-LSTM. The output of the transformed stream can achieve the best recognition accuracy compared with the other two fusion strategies. This means that our proposed transformer network is beneficial for processing the view-variant action recognition problem, because the output of the transformer network can cover some unseen patterns that appeared in the testing dataset.

Table 7.2: Performance comparison of STN and related models on Northwestern-UCLA

Method	Accuracy%		
AOG [128]	53.60%		
Lie group [71]	74.20%		
HBRNN-L [141]	78.52%		
HOPC [133]	80.00%		
EnTS-LSTM [157]	89.22%		
	transformed	original	fusion
STN	89.78%	85.00%	87.60%

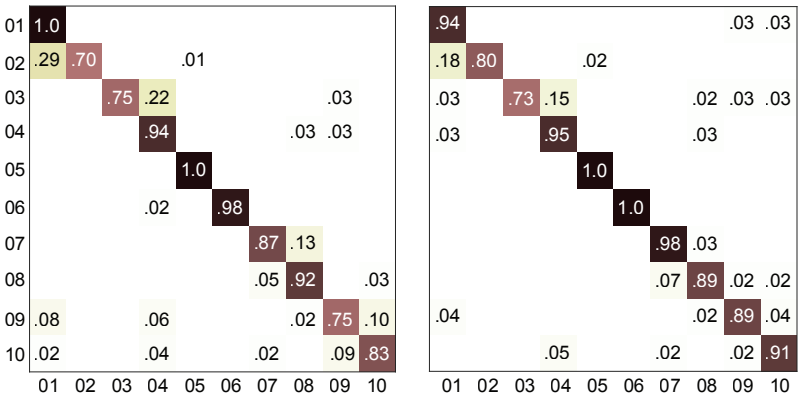


Figure 7.5: Confusion matrix of STN model on Northwestern UCLA

In order to discover more insights about the training process of the proposed framework on this dataset, we show the convergence rate curves of the training and testing on this database in Fig. 7.4. As can be seen, the training and testing performance for the fusion and original streams increase steadily in the whole training. The performance of the transformer stream decreases significantly sometimes, because of the uncertainty of the transformed sequence.

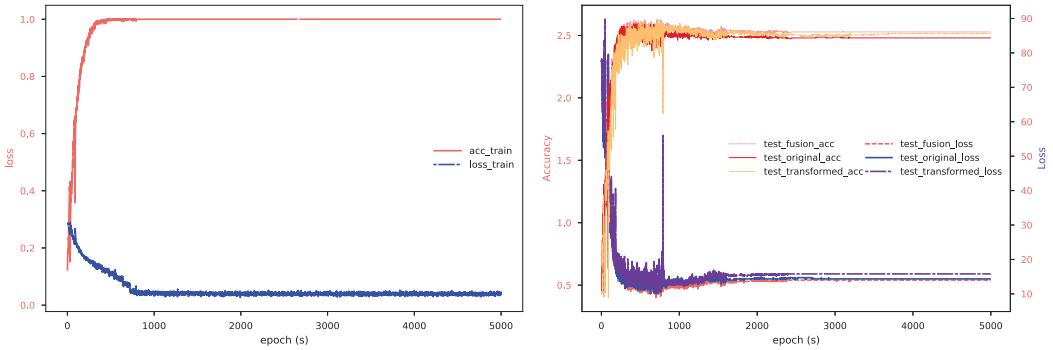


Figure 7.6: Training and testing accuracy and loss for STN on Northwestern UCLA

7.4.4 Results of NTU RGB+D database

The NTU RGB+D dataset is one most popular action recognition datasets, which includes 56,880 skeleton instances in total, and the noisy skeleton coordinates contained in this dataset pose great challenges to distinguish 60 actions. Following the two popular evaluation protocols: the Cross-Subject protocol and Cross-View protocol, the data is splitted into the training and testing subsets. For the cross-subject evaluation protocol, the training subset includes 40,320 samples and testing subset includes 16,560 samples. For the cross-view evaluation protocol, the training subset contains 37,920 samples, the left 18960 samples belong to the testing subset.

Following the standard evaluation protocol, the skeleton sequence contained in the training and testing dataset are different significantly, the difference can be alleviated by our proposed network. The proposed model reduced the difference existing in the training and testing dataset by introducing more noise during the training process to improve the generalization ability of the trained model on the test dataset. Table 7.3 presents the result of our proposed model and the various widely reported results on the NTU RGB+D dataset. The implementation of our sequential transformer network consists of 2 layers of the LSTM

Table 7.3: Performance comparison of STN and related models on NTU RGB+D

#	CS			CV		
HBRNN [141]	59.07%			63.97%		
P-LSTM [20]	62.93%			70.27%		
ST-LSTM [194]	69.20%			75.70%		
GCA-LSTM [189]	74.40%			82.80%		
Clips+CNN+MTLN [195]	79.57%			84.83%		
ST-GCN [77]	81.50%			88.30%		
IndRNN (6 layers LSTM) [196]	81.80%			87.97%		
	Transformed	Original	Fusion	Transformed	Original	Fusion
STN (copy-skeleton-of-first-subject)	81.30%	80.86%	83.22%	88.16%	88.03%	89.96%
STN (nocopy-skeleton-of-first-subject)	79.43%	78.80%	80.92%	84.94%	85.60	87.37%

network and the top action classifier is a 3 layers LSTM-RNN model, all of the LSTM layers contains the same number of hidden LSTM nodes, and the number of nodes of the last fully-connected layer equals the number of the action categories, which is 60 for this dataset. We used the Adam optimizer to optimize the whole framework, and the start learning rate is initialized as 0.001. The batch size of the training process is set as 32. The dropout regularization is utilized to alleviate the overfitting problem. We trained the model for 6000 epochs and report the best test result. We compared our results with [141], [20], [194], [189], [195], [196], our method can achieve superior performance compared with the other existing works.

The results listed in Table 7.3 show that our proposed approach greatly outperformed the other baseline systems. For Cross-View evaluation protocol, our STN outperforms the best existing model IndRNN by 2%. The result of our model on both the cross subject evaluation protocol and the cross view evaluation protocol outperforms the other state-of-the-art models, which means the proposed model is not only able to efficiently models the variation of the same action performed by the different people, it also can deal with the view changing problem very well. Because some action sequences in the NTU RGB+D database include more than one person, we investigated two strategies to process this problem, the "copy" and the "nocopy" strategy. The "copy" strategy means we copy the first person's joint coordinates if there is only one subject in the action sequence. In terms of the "nocopy" strategy, it means we keep the second subject as blank if there is only one person in the action sequence. The confusion matrix for the best recognition accuracy produced by the CV evaluation strategy is

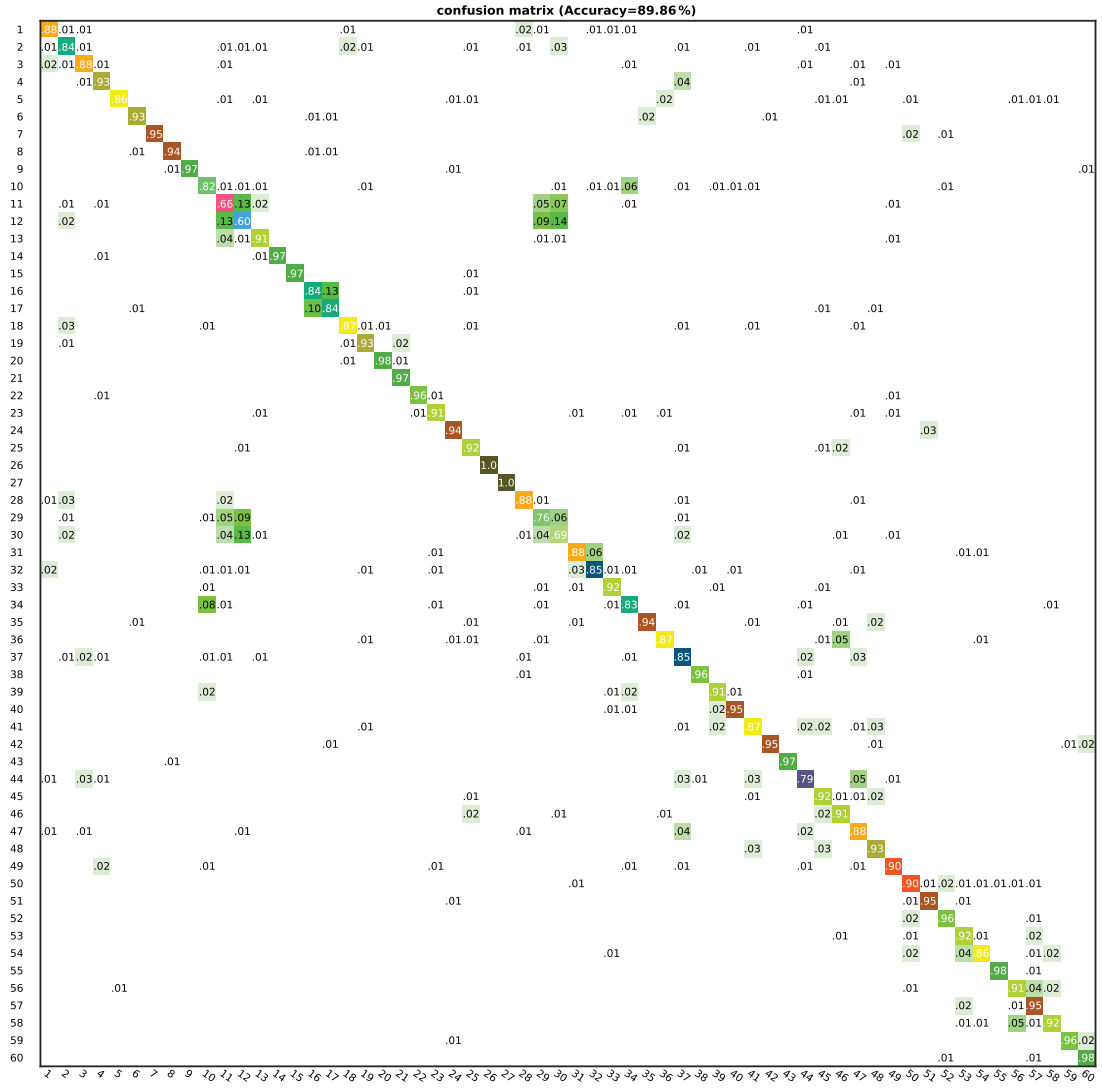


Figure 7.7: Confusion matrix of best result with Cross-View evaluation strategy

shown in Fig.7.7, it is easy to identify that the trained model has excellent performance, which can correctly classify most of the action sequences.

The convergence rate curve of our model during the training process is shown in Fig.7.8. The model converges much faster on this big database than that on the UTD-MHAD dataset, and the model can be well trained after around 2000 epochs. We predict that our proposed model can work well on an even larger dataset, we will verify our model on the NTU RGB+D 120 dataset, which includes 120 actions and more training examples. The great margin between our results with the compared results indicates that our proposed model has superior improvement on this dataset, so this validates that our approach can be employed to

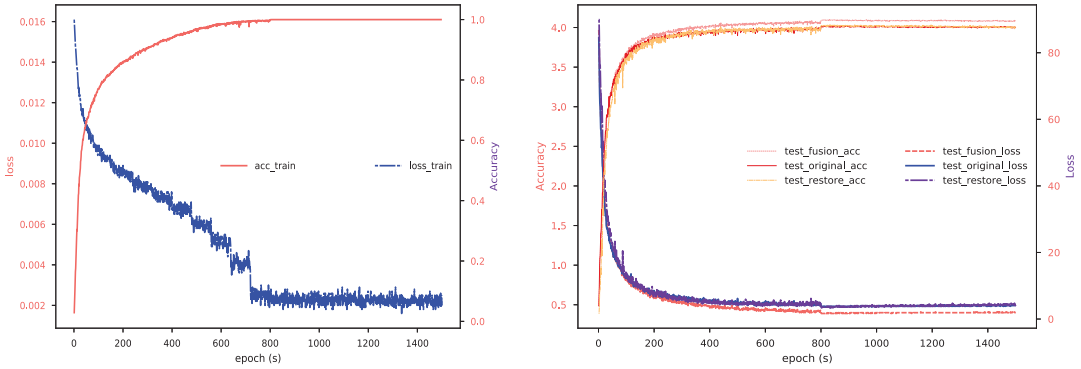


Figure 7.8: Convergence rate curve for training and testing on NTU RGB+D

other sequential problems.

7.4.5 Visualization

In order to provide more insights for the proposed skeleton transformer, we visualize the original input sequence and the transformed input sequence with the static images by following similar steps as those that were used in Chapter 3. Each of the joint coordinates, (x, y, z), is treated as a pixel value in the RGB image. The image representations of several raw input sequences and transformed sequences are shown in Fig. 7.9. There are two columns for 10 actions. The left column corresponds to the original raw input sequence, while the right column corresponds to the transformed sequence. From the visualisation of the original and the transformed features, it can be seen that some types of actions (as indicated by the numbers alongside each frame) have very similar feature patterns in the original features. For instance, action type 8 and action type 39 have features that resemble each other before the transformation. Inspecting their transformed features, we can see that the STN converted the original features to make them easier to discriminate. This verified that the skeleton transformer network removed the redundant noise and made the significant discriminative feature more outstanding. The transformed sequences were used to augment the training set, making the STN model more robust against noisy input sequences; thus, improving its generalisation ability.

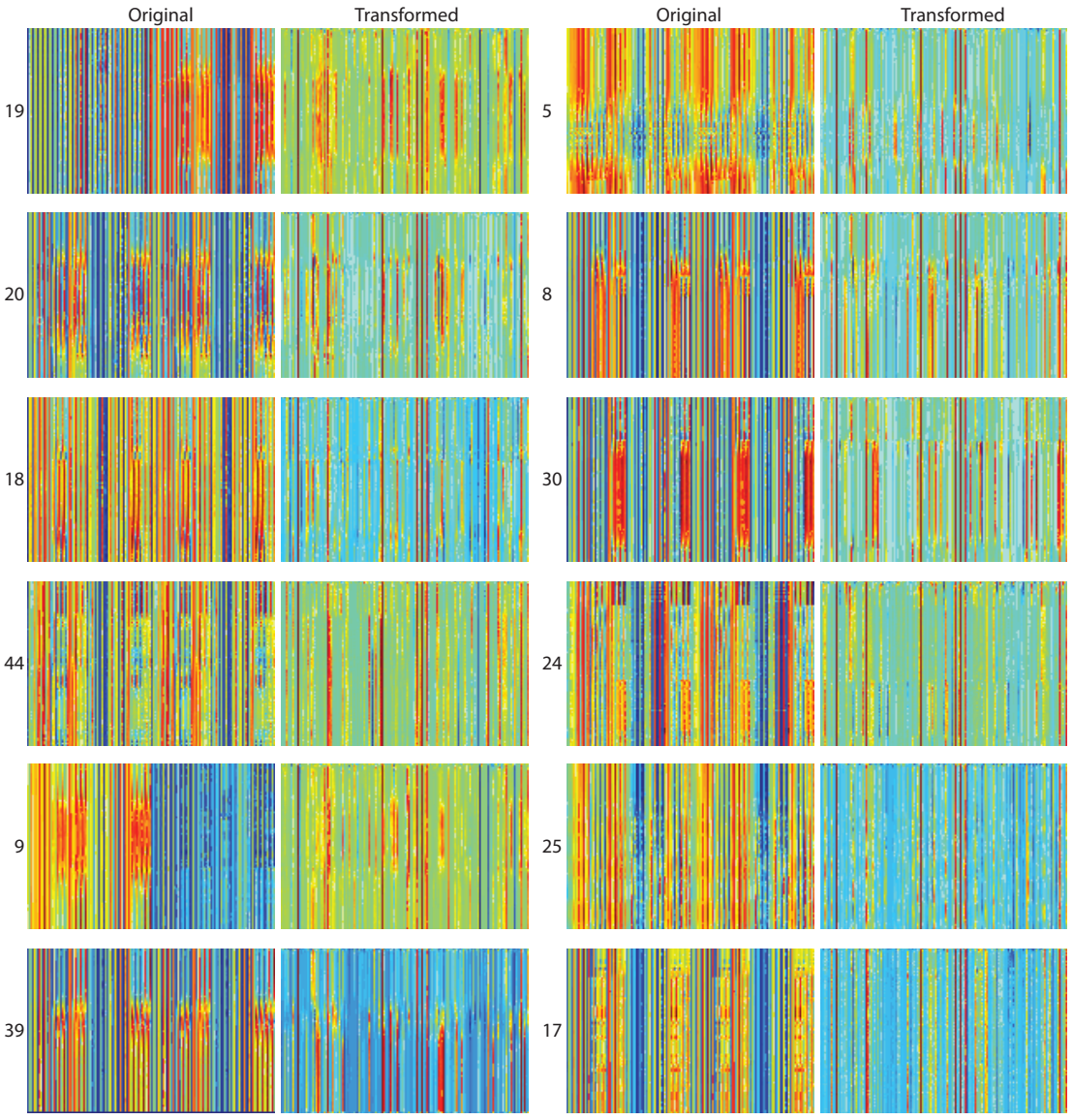


Figure 7.9: Image visualization of raw skeleton sequence and the transformed sequences. The number alongside each frame indicates the action type.

7.5 Summary and contributions

In summary, the following contributions can be highlighted for this Chapter.

1) We devised a novel and a highly efficient model for the skeleton-based human activity recognition and this framework significantly improves the recognition accuracy by introducing a novel skeleton transformation mechanism. The introduced STN model can help to produce a robust model that can perform better on the test dataset by augmenting the training dataset and training the recurrent model in an end-to-end manner. Another

function of the proposed skeleton transform operation is that it can help to remove the noise from the input sequence and keep the discriminative temporal dependency information that exists in the original sequence.

2) The customized loss function can control the loss between the classification loss and the transformation loss so as to extract a more discriminative representation from the input skeleton sequence. Experiments conducted on several challenging action recognition datasets, such as the NTU RGB+D, demonstrate that our proposed model can outperform those similar models reported in the latest literature.

3) The proposed STN model provides an efficient solution for representation learning from time-series data, especially if the time-series data includes complex patterns and if the data is corrupted by noise. This model can be easily adapted to other applications for time-series signal processing.

Chapter 8

Conclusion and perspectives

This thesis extensively explored and highlighted several contributions for HAR using a multi-features input. To conclude our work, we first summarize our experimental results and major contributions, and then point out some promising research directions for our future research.

8.1 Key contributions

This thesis investigated several automatic techniques for representation learning with multiple features as the input for HAR. The CNN based approaches were shown to achieve super good results as they can utilise the pretrained model to enhance the training process. The RNN based models are the most natural choices for human action recognition, which perform well in modelling the temporal dependency between the consecutive frames. Another reason for the popularity of the RNN models is that it requires much fewer parameters to be trained compared with the CNN model. However, when compared with the CNN model (demonstrated in Chapter 4 and 5), the RNN based approaches are not easy to train due to the overfitting problem. In summary, the four major contributions of this thesis can be identified as follows.

Our first contribution is that we introduced the concept of skeleton-based optical flow into the skeleton-based human action recognition, deriving the motion and energy features based on the raw skeleton coordinates. For this part, we investigated the traditional skeleton-based features and developed new geometric and kinematic features. Additionally, the performance of the proposed features were verified with two of the most popular deep learning models, the CNN model and the RNN model. Based on the baseline performance, we optimised these two models by proposing the correctness-vigilant regularizer and the

spatio-temporal kernel for the CNN and RNN model respectively. The proposed correctness-vigilant regularizer can help to speed up the training's convergence and improve the generalisation of the trained model (Chapter 3). The limitation of the CNN model is that the conversion of the skeleton to static images will lose temporal information. This is because any sequence longer than the predefined input size of 224x224 will have to be compressed, so key frames might be dropped. This can be addressed by using our proposed spatio-temporal kernel-based temporal convolutional model (Chapter 4). With the proposed spatio-temporal kernel, we could aggregate the sequential features locally and globally in a hierarchical way to effectively extract the instance-level features from the input skeleton sequence. Regarding the output the proposed two models, the experimental results showed a significant improvement on the tested benchmarking databases in terms of recognition accuracies.

The second contribution is that we devised a novel feature selection mechanism for the video-based action recognition system via the LSTM-C model (Chapter 5). It can integrate spatial features contained in static images and the temporal dynamics between the static frames together to formulate the final representation of the whole video. We proposed using CNN features to represent the spatial features, which will mitigate the stress of the RNN model to extract spatial features from image frames. With these CNN features as the input to our proposed LSTM-C recurrent neural network, the LSTM-C model can selectively extract discriminative frames from the input spatial features to formulate the final representation. We evaluated different configurations of the LSTM-C model on several databases to verify the effectiveness of the proposed framework. The empirical results demonstrate that the proposed model is an effective attention mechanism in extracting the key features from the input CNN features, and out-performed the baseline system. Action recognition from videos is a topic that has been researched extensively, and it is difficult to improve the performance of this approach further because of the significant computational cost and the lack of a more powerful model to extract compact patterns from RGB videos. In order to address this problem, we proposed using the skeleton data that is extracted from the video directly to carry out action recognition. The empirical results indicated that this approach is an efficient solution for video-based action recognition.

My third contribution of this research is that we proposed a novel multi-stream CNN model to improve the action recognition performance further. The proposed model demonstrated

good performance on small datasets with limited training data. Four fusion strategies (sum, average, multiply and max) were employed to fuse the extracted features by each CNN model and different combination options were also investigated in this contribution. The proposed framework was shown to achieve state-of-the-art performances on the UTD-MHAD dataset. (Chapter 6)

The last contribution of this thesis is that one efficient STN model is proposed, which provides an efficient data augmentation approach for skeleton-based action recognition (Chapter 7). Given an input sequence, the proposed network will output a variant of the original sequence, which contains all discriminative features but omits unrelated and redundant information that only contributes to noise. The results of the STN model on three challenging datasets demonstrate superiority of the proposed model over other state-of-the-art approaches. This framework provides us with a flexible and robust evaluation approach, which opens up more alternative classification results, and we can achieve the final optimal prediction by comparing the three output scores. The training strategies utilized in this model can efficiently prevent the error exploding and gradient vanishing problem that we encountered in Chapter 4 and Chapter 5.

8.2 Future work

The research presented in this thesis has explored various possibilities of representation learning for action recognition. The discovery raised more questions than it has addressed. The findings and discussion of our proposed models have allowed us to outline several future research directions for HAR.

Even though the proposed optical flow guided skeleton-based features contain rich information of human actions, utilising the skeleton-based geometric and kinematic features to improve recognition accuracies is still at a preliminary stage. Therefore, more options should be investigated in the future to improve the accuracy. For example, the number of geometric features needs to be determined for each situation. The best set of geometric features should also be investigated in the future. A potential research direction for future work can be feature-selection based on our proposed geometric and kinematic features. In this research, we mainly considered actions that involve one or two subjects. In

the future we may extend our researches to that involve more than two subjects. Another direction is to introduce a more powerful neural network model, including both CNNs, RNNs and other complex architectures to process the extracted geometric features.

For RGB video-based action recognition, we use the CNN model to extract spatial features from RGB images. Then we utilised a customised RNN model to discover the temporal patterns in time domain. In the future, we can explore the idea of introducing more semantic features into our framework, such as, saliency based features. This is a promising trend for video-based action recognition since modern deep neural networks make it possible to extract rich and accurate information from videos. In addition, the extracted features from videos, such as depth, skeleton, motions and so forth, can provide more options and more promising solutions for RGB video based action recognition.

Another research line involves integrating the potential attention mechanism that can be integrated into our proposed framework. This should be further investigated. In this research study, we introduced several attention mechanisms in the action recognition framework, however a more flexible and effective attention mechanism should be looked into. This can be useful for applications, such as action detection and anomaly action detection. In the future we intend to extend our proposed attention mechanism into action detection, which is much more challenging compared to action recognition. At last, a more powerful attention mechanism and a more powerful signal fusion approach should be investigated in the future for multi-modality features-based action recognition.

Bibliography

[1] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 3, pp. 257–267, 2001. DOI: 10.1109/34.910878 (cited on pages 1, 15).

[2] Y. M. Lui and J. R. Beveridge, "Tangent bundle for human action recognition", in *2011 IEEE International Conference on Automatic Face and Gesture Recognition and Workshops, FG 2011*, 2011, pp. 97–102. DOI: 10.1109/FG.2011.5771378 (cited on page 1).

[3] K. Xu, X. Jiang and T. Sun, "Two-Stream Dictionary Learning Architecture for Action Recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 3, pp. 567–576, 2017. DOI: 10.1109/TCSVT.2017.2665359 (cited on page 3).

[4] I. Laptev, "On space-time interest points", in *International Journal of Computer Vision*, vol. 64, 2005, pp. 107–123. DOI: 10.1007/s11263-005-1838-7 (cited on pages 3, 12, 14, 15).

[5] Hao Zhang and L. E. Parker, "4-dimensional local spatio-temporal features for human activity recognition", 2011. DOI: 10.1109/iros.2011.6094489 (cited on page 3).

[6] G. Willems, T. Tuytelaars and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5303 LNCS, 2008, pp. 650–663. DOI: 10.1007/978-3-540-88688-4-48 (cited on pages 3, 12, 14).

[7] N. Li, X. Cheng, S. Zhang and Z. Wu, "Realistic human action recognition by Fast HOG3D and self-organization feature map", *Machine Vision and Applications*, vol. 25, no. 7, pp. 1793–1812, 2014. DOI: 10.1007/s00138-014-0639-9 (cited on page 3).

[8] F. Murtaza, M. H. Yousaf and S. A. Velastin, "Multi-view Human Action Recognition Using Histograms of Oriented Gradients (HOG) Description of Motion History Images (MHIs)", in *Proceedings - 2015 13th International Conference on Frontiers of Information Technology, FIT 2015*, 2016, pp. 297–302. DOI: 10.1109/FIT.2015.59 (cited on page 3).

[9] G. Willems, T. Tuytelaars and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2008. DOI: 10.1007/978-3-540-88688-4-48 (cited on page 3).

[10] J. Schmidhuber, *Deep Learning in neural networks: An overview*, 2015. DOI: 10.1016/j.neunet.2014.09.003. arXiv: 1404.7828 (cited on page 4).

[11] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biological Cybernetics*, 1980. DOI: 10.1007/BF00344251 (cited on page 4).

[12] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", *arXiv preprint arXiv*, p. 1312.6229, 2013. arXiv: 1312.6229 (cited on pages 4, 23).

[13] D. Impiombato, S. Giarrusso, T. Mineo, O. Catalano, C. Gargano, G. La Rosa, F. Russo, G. Sottile, S. Billotta, G. Bonanno, S. Garozzo, A. Grillo, D. Marano and G. Romeo, "You Only Look Once: Unified, Real-Time Object Detection", *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 794, pp. 185–192, 2015. DOI: 10.1016/j.nima.2015.05.028. arXiv: arXiv:1506.02640 (cited on pages 4, 23).

[14] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016. DOI: 10.1109/TPAMI.2015.2437384. arXiv: 1311.2524 (cited on pages 4, 23).

[15] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad and S. Baik, "Action Recognition in Video Sequences using Deep Bi-directional LSTM with CNN Features", *IEEE Access*, vol. 3536, no. c, pp. 1–11, 2017. DOI: 10.1109/ACCESS.2017.2778011 (cited on pages 4, 86).

[16] A. Graves and J. J. Schmidhuber, "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks", *Advances in Neural Information Processing Systems 21, NIPS'21*, pp. 545–552, 2008. DOI: 10.1007/978-1-4471-4072-6 (cited on page 4).

[17] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink and J. Schmidhuber, *LSTM: A Search Space Odyssey*, 2016. DOI: 10.1109/TNNLS.2016.2582924. arXiv: 1503.04069 (cited on page 4).

[18] S. Hochreiter and J. Urgen Schmidhuber, "LONG SHORT-TERM MEMORY", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735. arXiv: 1206.2944 (cited on pages 4, 69, 121).

[19] S. J. Bu and S. B. Cho, "A hybrid deep learning system of CNN and LRCN to detect cyberbullying from SNS comments", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018. DOI: 10.1007/978-3-319-92639-1_47 (cited on page 5).

[20] A. Shahroudy, J. Liu, T.-T. Ng and G. Wang, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis", pp. 1010–1019, 2016. DOI: 10.1109/CVPR.2016.115. arXiv: 1604.02808 (cited on pages 5, 12, 45, 47, 48, 52, 112, 113, 117, 123, 127).

[21] G. Guo and A. Lai, "A survey on still image based human action recognition", *Pattern Recognition*, vol. 47, no. 10, pp. 3343–3361, 2014. DOI: 10.1016/j.patcog.2014.04.018 (cited on pages 9, 16).

[22] L. Lo Presti and M. La Cascia, "3D skeleton-based human action classification: A survey", *Pattern Recognition*, vol. 53, pp. 130–147, 2016. DOI: 10.1016/j.patcog.2015.11.019. arXiv: arXiv:1212.0402 (cited on pages 10, 29, 56, 116).

[23] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review", *ACM Computing Surveys*, vol. 43, no. 3, 16:1–16:43, 2011. DOI: 10.1145/1922649.1922653 (cited on pages 10, 28).

[24] L. Zelnik-Manor and M. Irani, "Event-based analysis of video", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, no. 1229, pp. 1–18, 2001. DOI: 10.1109/CVPR.2001.990935 (cited on page 11).

[25] D. Weinland, R. Ronfard and E. Boyer, "Free viewpoint action recognition using motion history volumes", *Computer Vision and Image Understanding*, vol. 104, no. 2-3 SPEC. ISS. Pp. 249–257, 2006. DOI: 10.1016/j.cviu.2006.07.013 (cited on page 11).

[26] L. Gorelick, M. Blank, E. Shechtman, M. Irani and R. Basri, "Actions as space-time shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 12, pp. 2247–2253, 2007. DOI: 10.1109/TPAMI.2007.70711 (cited on page 11).

[27] C. Chen, R. Jafari and N. Kehtarnavaz, "UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor", in *Proceedings - International Conference on Image Processing, ICIP*, 2015. DOI: 10.1109/ICIP.2015.7350781 (cited on pages 12, 45, 48, 79, 99, 123).

[28] P. Dollár, V. Rabaud, G. Cottrell and S. Belongie, "Behavior recognition via sparse spatio-temporal features", in *Proceedings - 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, VS-PETS*, vol. 2005, 2005, pp. 65–72. DOI: 10.1109/VSPETS.2005.1570899 (cited on pages 12, 14).

[29] A. Klaser, M. Marszalek and C. Schmid, "A Spatio-Temporal Descriptor Based on 3D-Gradients", *Proceedings of the British Machine Conference*, pp. 99.1–99.10, 2008. DOI: 10.5244/C.22.99 (cited on pages 12, 14).

[30] P. Scovanner, S. Ali and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition", *Proceedings of the ACM International Conference on Multimedia (MM 2007)*, no. c, p. 357, 2007. DOI: 10.1145/1291233.1291311 (cited on pages 12, 14).

[31] I. Laptev, M. Marszalek, C. Schmid and B. Rozenfeld, "Learning realistic human actions from movies", in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008. DOI: 10.1109/CVPR.2008.4587756 (cited on pages 12, 14).

[32] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories", *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 3551–3558, 2013. DOI: 10.1109/ICCV.2013.441 (cited on pages 13, 24).

[33] H. Wang, A. Kläser, C. Schmid and C. L. Liu, "Action recognition by dense trajectories", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3169–3176. DOI: 10.1109/CVPR.2011.5995407 (cited on pages 13, 16, 24, 96).

[34] H. Wang, A. Klaser, C. Schmid and C. L. Liu, "Dense trajectories and motion boundary descriptors for action recognition", *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013. DOI: 10.1007/s11263-012-0594-8 (cited on pages 13, 16, 24).

[35] L. Wiskott, P. Berkes, M. Franzius, H. Sprekeler and N. Wilbert, "Slow feature analysis", *Scholarpedia*, vol. 6, no. 2011, p. 5282, 2011. DOI: 10.4249/scholarpedia.5282 (cited on page 13).

[36] S. Arora, R. Ge, T. Ma and A. Moitra, "Simple, Efficient, and Neural Algorithms for Sparse Coding", *arXiv:1503.00778 [cs, stat]*, 2015. arXiv: 1503.00778 (cited on page 13).

[37] J. Yang, K. Yu, Y. Gong and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification", in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009, pp. 1794–1801. DOI: 10.1109/CVPRW.2009.5206757. arXiv: 1504.06897 (cited on page 13).

[38] L. Liu, L. Wang and X. Liu, "In defense of soft-assignment coding", in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2486–2493. DOI: 10.1109/ICCV.2011.6126534 (cited on page 13).

[39] F. Perronnin, J. Sánchez and T. Mensink, "Improving the Fisher kernel for large-scale image classification", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6314 LNCS, 2010, pp. 143–156. DOI: 10.1007/978-3-642-15561-1_11 (cited on page 13).

[40] J. Sánchez, F. Perronnin, T. Mensink and J. Verbeek, "Image classification with the fisher vector: Theory and practice", *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, 2013. DOI: 10.1007/s11263-013-0636-x (cited on page 13).

[41] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez and C. Schmid, "Aggregating local image descriptors into compact codes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1704–1716, 2012. DOI: 10.1109/TPAMI.2011.235 (cited on page 13).

[42] Z. Cai, L. Wang, X. Peng and Y. Qiao, "Multi-view super vector for action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 596–603. DOI: 10.1109/CVPR.2014.83 (cited on page 13).

[43] L. Wang, H. Zhou, S. C. Low and C. Leckie, "Action recognition via multi-feature fusion and Gaussian process classification", in *2009 Workshop on Applications of Computer Vision, WACV 2009*, 2009. DOI: 10.1109/WACV.2009.5403113 (cited on page 13).

[44] Y. Yang, J. Song, Z. Huang, Z. Ma, N. Sebe and A. G. Hauptman, "Multi-feature fusion via hierarchical regression for multimedia analysis", *IEEE Transactions on Multimedia*, vol. 15, no. 3, pp. 572–581, 2013. DOI: 10.1109/TMM.2012.2234731 (cited on page 13).

[45] G. Nagendar, S. Ganesh, M. Goud and C. V. Jawahar, "Action Recognition using Canonical Correlation Kernels by Action Recognition using Canonical Correlation Kernels", in *ACCV*, 2012 (cited on page 13).

[46] L. Wang and H. Sahbi, "Directed Acyclic Graph Kernels for Action Recognition", *The IEEE International Conference on Computer Vision (ICCV)*, pp. 3168–3175, 2013. DOI: 10.1109/ICCV.2013.393 (cited on page 13).

[47] C. Yuan, W. Hu, H. Wang, X. Li and N. Xie, "Spatio-temporal proximity distribution kernels for action recognition", *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, no. January, pp. 1126–1129, 2010. DOI: 10.1109/ICASSP.2010.5495360 (cited on page 13).

[48] X. Wu and Y. Jia, "View-Invariant Action Recognition Using Latent", pp. 411–424, 2012. DOI: 10.1007/978-3-642-33715-4_30 (cited on page 13).

[49] K. Kawaji, P. Spincemaille, T. D. Nguyen, N. Thimmappa, M. A. Cooper, M. R. Prince and Y. Wang, "Direct coronary motion extraction from a 2D fat image navigator for prospectively gated coronary MR angiography", *Magnetic Resonance in Medicine*, vol. 71, no. 2, pp. 599–607, 2014. DOI: 10.1002/mrm.24698 (cited on page 14).

[50] M. A. R. Ahad, J. K. Tan, H. Kim and S. Ishikawa, *Motion history image: Its variants and applications*, 2012. DOI: 10.1007/s00138-010-0298-4 (cited on page 14).

[51] C. Schüldt, I. Laptev and B. Caputo, "Recognizing human actions: A local SVM approach", in *Proceedings - International Conference on Pattern Recognition*, vol. 3, 2004, pp. 32–36. DOI: 10.1109/ICPR.2004.1334462 (cited on page 14).

[52] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2046–2053. DOI: 10.1109/CVPR.2010.5539881 (cited on pages 14, 96, 97).

[53] A. Gilbert, J. Illingworth and R. Bowden, "Action recognition using mined hierarchical compound features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 883–897, 2011. DOI: 10.1109/TPAMI.2010.144 (cited on page 14).

[54] D. Han, L. Bo and C. Sminchisescu, "Selection and context for action recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 1933–1940. DOI: 10.1109/ICCV.2009.5459427 (cited on page 14).

[55] J. C. Niebles and F. F. Li, "A hierarchical model of shape and appearance for human action classification", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. DOI: 10.1109/CVPR.2007.383132 (cited on page 14).

[56] Y. Wang and G. Mori, "Hidden part models for human action recognition: Probabilistic versus max margin", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1310–1323, 2011. DOI: 10.1109/TPAMI.2010.214 (cited on page 15).

[57] Y. Xie, H. Chang, Z. Li, L. Liang, X. Chen and D. Zhao, "A unified framework for locating and recognizing human actions", *Cvpr 2011*, pp. 25–32, 2011. DOI: 10.1109/CVPR.2011.5995648 (cited on page 15).

[58] Y. Tian, R. Sukthankar and M. Shah, "Spatiotemporal deformable part models for action detection", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2642–2649. DOI: 10.1109/CVPR.2013.341 (cited on page 15).

[59] L. Xia and J. K. Aggarwal, "Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2834–2841. DOI: 10.1109/CVPR.2013.365 (cited on pages 15, 20).

[60] Zhang Zhang and Dacheng Tao, "Slow Feature Analysis for Human Action Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 436–450, 2012. DOI: 10.1109/TPAMI.2011.157 (cited on pages 15, 16).

[61] R. Messing, C. Pal and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints", in *Proceedings of the IEEE International Conference on Computer Vision*, 2009, pp. 104–111. DOI: 10.1109/ICCV.2009.5459154 (cited on page 16).

[62] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Imaging*, vol. 130, no. x, pp. 674–679, 1981. DOI: 10.1109/HPDC.2004.1323531. arXiv: 3629719 (cited on page 16).

[63] R. Souvenir and J. Babbs, "Learning the viewpoint manifold for action recognition", in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008. DOI: 10.1109/CVPR.2008.4587552 (cited on page 17).

[64] O. Oreifej and Z. Liu, "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 716–723. DOI: 10.1109/CVPR.2013.98 (cited on pages 17, 20, 47).

[65] W. Li, Z. Zhang and Z. Liu, "Action recognition based on a bag of 3D points", in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops, CVPRW 2010*, 2010, pp. 9–14. DOI: 10.1109/CVPRW.2010.5543273 (cited on pages 17, 75).

[66] L. Xia, C. C. Chen and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 20–27. DOI: 10.1109/CVPRW.2012.6239233 (cited on pages 17, 30, 34, 73).

[67] Y. Song, J. H. Tang, F. Liu and S. C. Yan, "Body Surface Context: A New Robust Feature for Action Recognition From Depth Videos", *Ieee Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 952–964, 2014. DOI: Doi10.1109/Tcsvt.2014.2302558 (cited on page 18).

[68] C. Lu, J. Jia and C. K. Tang, "Range-sample depth feature for action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 772–779. DOI: 10.1109/CVPR.2014.104 (cited on page 18).

[69] C. Wang, Y. Wang and A. L. Yuille, "An approach to pose-based action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 915–922. DOI: 10.1109/CVPR.2013.123 (cited on pages 18, 19, 25).

[70] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 471–478. DOI: 10.1109/CVPRW.2013.153 (cited on page 19).

[71] R. Vemulapalli, F. Arrate and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588–595. DOI: 10.1109/CVPR.2014.82 (cited on pages 19, 30, 47, 75, 125).

[72] F. Lv and R. Nevatia, "Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3954 LNCS, 2006, pp. 359–372. DOI: 10.1007/11744085_28 (cited on page 19).

[73] D. Wu and L. Shao, "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 724–731. DOI: 10.1109/CVPR.2014.98 (cited on page 19).

[74] D. Gong, G. Medioni and X. Zhao, "Structured Time Series Analysis for Human Action Segmentation and Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1414–1427, 2014. DOI: 10.1109/TPAMI.2013.244 (cited on page 19).

[75] J. Wang, Z. Liu, J. Chorowski, Z. Chen and Y. Wu, "Robust 3D action recognition with random occupancy patterns", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7573 LNCS, 2012, pp. 872–885. DOI: 10.1007/978-3-642-33709-3_62 (cited on pages 20, 29).

[76] X. Yang and Y. Tian, "Super normal vector for activity recognition using depth sequences", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 804–811. DOI: 10.1109/CVPR.2014.108 (cited on pages 20, 51).

[77] S. Yan, Y. Xiong and D. Lin, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition", 2018. arXiv: 1801.07455 (cited on pages 20, 127).

[78] S. Hadfield and R. Bowden, "Hollywood 3D: Recognizing actions in 3D natural scenes", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3398–3405. DOI: 10.1109/CVPR.2013.436 (cited on page 20).

[79] L. Liu and L. Shao, "Learning discriminative representations from RGB-D video data", in *IJCAI International Joint Conference on Artificial Intelligence*, 2013, pp. 1493–1500 (cited on page 20).

[80] C. Jia, Y. Kong, Z. Ding and Y. Fu, "Latent tensor transfer learning for RGB-D action recognition", *MM 2014 - Proceedings of the 2014 ACM Conference on Multimedia*, no. November, pp. 87–96, 2014. DOI: 10.1145/2647868.2654928 (cited on pages 20, 21).

[81] Y. Kong and Y. Fu, "Heterogeneous Information Machine for RGB-D Action Recognition", pp. 1054–1062, 2015 (cited on page 21).

[82] A. A. Chaaraoui, J. R. Padilla-López and F. Flórez-Revuelta, "Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices", in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 91–97. DOI: 10.1109/ICCVW.2013.19 (cited on page 21).

[83] Y. Y. Lin, J. H. Hua, N. C. Tang, M. H. Chen and H. Y. M. Liao, "Depth and skeleton associated action recognition without online accessible RGB-D cameras", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2617–2624. DOI: 10.1109/CVPR.2014.335 (cited on page 21).

[84] M. Yu, L. Liu and L. Shao, "Structure-preserving binary representations for RGB-D action recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1651–1664, 2016. DOI: 10.1109/TPAMI.2015.2491925 (cited on page 21).

[85] H. Kim, J. Lee and H. Yang, "Human action recognition using a modified convolutional neural network", *Advances in Neural Networks-ISNN 2007*, pp. 715–723, 2007. DOI: 10.1007/978-3-540-72393-6_85 (cited on page 22).

[86] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia and A. Baskurt, "Action classification in soccer videos with long short-term memory recurrent neural networks", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6353 LNCS, 2010, pp. 154–159. DOI: 10.1007/978-3-642-15822-3_20 (cited on pages 22, 23).

[87] M. Baccouche, F. Mamalet and C. Wolf, "Sequential deep learning for human action recognition", *Proc. Int. Conf. Human Behavior Understanding (HBU)*, pp. 29–39, 2011. DOI: 10.1007/978-3-642-25446-8. arXiv: 1506.03607v1 (cited on pages 22, 24).

[88] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos", *Advances in neural information processing systems*, pp. 568–576, 2014. DOI: 10.1017/CBO9781107415324.004. arXiv: arXiv:1406.2199v1 (cited on pages 22–24, 86).

[89] Q. V. Le, W. Y. Zou, S. Y. Yeung and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3361–3368, 2011. DOI: 10.1109/CVPR.2011.5995496. arXiv: 1312.5602 (cited on pages 22, 23, 25, 96).

[90] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn and D. Yu, "Convolutional Neural Networks for Speech Recognition", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014. DOI: 10.1109/TASLP.2014.2339736. arXiv: 1408.5882v1 (cited on pages 23, 29, 34).

[91] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar and F. F. Li, "Large-scale video classification with convolutional neural networks", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223. arXiv: 1412.0767 (cited on page 23).

[92] Y. Kim, "Convolutional Neural Networks for Sentence Classification", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1746–1751, 2014. DOI: 10.1109/LSP.2014.2325781. arXiv: arXiv:1408.5882v1 (cited on page 23).

[93] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965. arXiv: 1411.4038 (cited on page 23).

[94] A. Severyn and A. Moschitti, "Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks Categories and Subject Descriptors", *Sigir*, pp. 373–382, 2015. DOI: 10.1145/2766462.2767738 (cited on page 23).

[95] Y. Shen, X. He, J. Gao, L. Deng and G. Mesnil, "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval", *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, pp. 101–110, 2014. DOI: 10.1145/2661829.2661935. arXiv: 1502.02367 (cited on page 23).

[96] Y. Sun, X. Wang and X. Tang, "Deep Learning Face Representation by Joint Identification-Verification", in *NIPS*, 2014, pp. 1–9. DOI: 10.1109/CVPR.2014.244. arXiv: 1406.4773 (cited on page 23).

[97] 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*, vol. 11-18-Dece, 2016, pp. 4489–4497. DOI: 10.1109/ICCV.2015.510. arXiv: 1412.0767 (cited on pages 23, 25).

[98] T. Zeng and S. Ji, "Deep Convolutional Neural Networks for Multi-instance Multi-task Learning", *2015 IEEE International Conference on Data Mining*, pp. 579–588, 2015. DOI: 10.1109/ICDM.2015.92 (cited on page 23).

[99] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 2625–2634. DOI: 10.1109/CVPR.2015.7298878. arXiv: arXiv:1411.4389v3 (cited on pages 24, 25).

[100] 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 *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016. DOI: 10.1007/978-3-319-46484-8_2 (cited on page 24).

[101] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye and X. Xue, "Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification", *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 461–470, 2015. DOI: 10.1145/2733373.2806222. arXiv: 1504.01561 (cited on page 24).

[102] L. Wang, Y. Qiao and X. Tang, "Action recognition with trajectory-pooled deep-convolutional descriptors", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 4305–4314. DOI: 10.1109/CVPR.2015.7299059. arXiv: 1505.04868 (cited on page 24).

[103] J. Y. H. 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 Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 4694–4702. DOI: 10.1109/CVPR.2015.7299101. arXiv: 1503.08909 (cited on page 24).

[104] L. Sun, K. Jia, T. H. Chan, Y. Fang, G. Wang and S. Yan, "DL-SFA: Deeply-learned slow feature analysis for action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2625–2632. DOI: 10.1109/CVPR.2014.336 (cited on page 25).

[105] N. Srivastava, E. Mansimov and R. Salakhutdinov, "Unsupervised Learning of Video Representations using LSTMs", *Bmvc2015*, p. 2009, 2015. DOI: citeulike-article-id:13519737. arXiv: 1502.04681 (cited on pages 26, 121).

[106] M. Hasan and A. K. Roy-Chowdhury, "Continuous Learning of Human Activity Models Using Deep Nets", *Eccv*, vol. 8691, pp. 705–720, 2014. DOI: 10.1007/978-3-319-10578-9_46 (cited on page 26).

[107] B. Liang and L. Zheng, "A Survey on Human Action Recognition Using Depth Sensors", in *2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2015, pp. 1–8. DOI: 10.1109/DICTA.2015.7371223 (cited on page 28).

[108] A. Krizhevsky, I. Sutskever and H. Geoffrey E., "ImageNet Classification with Deep Convolutional Neural Networks", *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pp. 1–9, 2012. DOI: 10.1109/5.726791. arXiv: 1102.0183 (cited on page 29).

[109] S. Liao, J. Wang, R. Yu, K. Sato and Z. Cheng, "CNN for situations understanding based on sentiment analysis of twitter data", in *Procedia Computer Science*, vol. 111, 2017, pp. 376–381. DOI: 10.1016/j.procs.2017.06.037 (cited on page 29).

[110] G. Cheron, I. Laptev and C. Schmid, "P-CNN: Pose-based CNN features for action recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 11-18-Dece, 2016, pp. 3218–3226. DOI: 10.1109/ICCV.2015.368. arXiv: 1506.03607 (cited on page 29).

[111] Y. Du, Y. Fu and L. Wang, "Skeleton based action recognition with convolutional neural network", in *Proceedings - 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015*, 2016, pp. 579–583. DOI: 10.1109/ACPR.2015.7486569 (cited on pages 29, 30, 35, 36, 57, 60, 61).

[112] R. Dunne and N. Campbell, "On the pairing of the Softmax activation and cross-entropy penalty functions and the derivation of the Softmax activation function", *Proc. 8th Aust. Conf. on the Neural Networks*, pp. 1–5, 1997. DOI: 10.1.1.49.6403 (cited on page 29).

[113] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal and R. Bajcsy, "Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition", *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 24–38, 2014. DOI: 10.1016/j.jvcir.2013.04.007 (cited on pages 30, 58).

[114] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and HOG2 for action recognition", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 465–470. DOI: 10.1109/CVPRW.2013.76 (cited on pages 30, 34).

[115] A. Yao, J. Gall, G. Fanelli and L. V. Gool, "Does Human Action Recognition Benefit from Pose Estimation?", *Procedings of the British Machine Vision Conference 2011*, pp. 67.1–67.11, 2011. DOI: 10.5244/C.25.67 (cited on page 30).

[116] J. Ren, N. Reyes, A. Barczak, C. Scogings and M. Liu, "Toward three-dimensional human action recognition using a convolutional neural network with correctness-vigilant regularizer", *Journal of Electronic Imaging*, 2018. DOI: 10.1117/1.jei.27.4.043040 (cited on page 30).

[117] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang and G. Wang, "Recent Advances in Convolutional Neural Networks", *arXiv*, pp. 1–14, 2015. DOI: 10.3389/fpsyg.2013.00124. arXiv: arXiv:1512.07108v1 (cited on page 31).

[118] G. Obozinski, B. Taskar and M. Jordan, "Multi-task feature selection", *Statistics Department, UC Berkeley*, pp. 1–15, 2006 (cited on page 31).

[119] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. DOI: 10.1214/12-AOS1000. arXiv: 1102.4807 (cited on page 31).

[120] C. S. S Ioffe and C. S. Sergey Ioffe, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", *Uma ética para quantos?*, vol. XXXIII, no. 2, pp. 81–87, 2014. DOI: 10.1007/s13398-014-0173-7.2. arXiv: arXiv:1011.1669v3 (cited on page 31).

[121] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville and Y. Bengio, "Maxout Networks", 2013. DOI: 10.1093/bib/bbw065.. arXiv: 1302.4389 (cited on page 31).

[122] L. Xie, J. Wang, Z. Wei, M. Wang and Q. Tian, "DisturbLabel: Regularizing CNN on the loss layer", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 4753–4762. DOI: 10.1109/CVPR.2016.514. arXiv: 1605.00055 (cited on pages 31, 41).

[123] G. Hinton, O. Vinyals and J. Dean, "Distilling the Knowledge in a Neural Network", pp. 1–9, 2015. DOI: 10.1063/1.4931082. arXiv: 1503.02531 (cited on page 31).

[124] T. S. Kim and A. Reiter, "Interpretable 3D Human Action Analysis with Temporal Convolutional Networks", *arXiv preprint arXiv*, 2017. DOI: 10.1109/CVPRW.2017.207. arXiv: 1704.04516 (cited on pages 34, 46, 47).

[125] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", 2015. DOI: 10 . 1109 / CVPR . 2016 . 308. arXiv: 1512.00567 (cited on page 41).

[126] X. Glorot, A. Bordes and Y. Bengio, "Deep sparse rectifier neural networks", *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, vol. 15, pp. 315–323, 2011. DOI: 10.1.1.208.6449. arXiv: 1502.03167 (cited on page 43).

[127] D. Yu, K. Yao, H. Su, G. Li and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013. DOI: 10.1109/ICASSP.2013.6639201 (cited on page 44).

[128] J. Wang, X. Nie, Y. Xia, Y. Wu and S. C. Zhu, "Cross-view action modeling, learning, and recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2649–2656. DOI: 10 . 1109 / CVPR . 2014 . 339. arXiv: arXiv:1405.2941v1 (cited on pages 45, 47, 49, 125).

[129] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *International Conference on Learning Representations (ICRL)*, pp. 1–14, 2015. DOI: 10.1016/j.infsof.2008.09.005. arXiv: 1409.1556 (cited on page 46).

[130] F. Chollet, "Xception: Deep learning with depthwise separable convolutions", *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 1800–1807, 2017. DOI: 10.1109/CVPR.2017.195. arXiv: 1610.02357 (cited on page 46).

[131] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image steganalysis", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1–17. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385 (cited on page 46).

[132] C. Li, Y. Hou, P. Wang and W. Li, "Joint Distance Maps Based Action Recognition with Convolutional Neural Networks", *IEEE Signal Processing Letters*, 2017. DOI: 10.1109/LSP.2017.2678539 (cited on pages 46–48, 113, 123).

[133] H. Rahmani, A. Mian and M. Shah, "Learning a Deep Model for Human Action Recognition from Novel Viewpoints", pp. 1–14, 2016. DOI: 10.1103/PhysRevD.94.065007. arXiv: 1602.00828 (cited on pages 46, 47, 51, 125).

[134] Q. Ke, S. An, M. Bennamoun, F. Sohel and F. Boussaid, "SkeletonNet: Mining Deep Part Features for 3-D Action Recognition", *IEEE Signal Processing Letters*, vol. 24, no. 6, pp. 731–735, 2017. DOI: 10.1109/LSP.2017.2690339 (cited on pages 46, 47, 75).

[135] P. Wang, W. Li, C. Li and Y. Hou, "Action recognition based on joint trajectory maps with convolutional neural networks", *Knowledge-Based Systems*, 2018. DOI: 10.1016/j.knosys.2018.05.029 (cited on pages 47, 48, 113, 123).

[136] Y. Hou, Z. Li, P. Wang and W. Li, "Skeleton Optical Spectra-Based Action Recognition Using Convolutional Neural Networks", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 3, pp. 807–811, 2018. DOI: 10.1109/TCSVT.2016.2628339 (cited on pages 47, 113, 123).

[137] B. Zhang, Y. Yang, C. Chen, L. Yang, J. Han and L. Shao, "Action Recognition Using 3D Histograms of Texture and A Multi-Class Boosting Classifier", *IEEE Transactions on Image Processing*, 2017. DOI: 10.1109/TIP.2017.2718189 (cited on pages 47, 113, 123).

[138] M. E. Hussein, M. Torki, M. A. Gowayyed and M. El-Saban, "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations", *IJCAI International Joint Conference on Artificial Intelligence*, pp. 2466–2472, 2013 (cited on pages 47, 48, 113).

[139] M. Liu, H. Liu and C. Chen, "Enhanced skeleton visualization for view invariant human action recognition", *Pattern Recognition*, vol. 68, pp. 346–362, 2017. DOI: 10.1016/j.patcog.2017.02.030 (cited on pages 47, 112).

[140] X. Yang and Y. L. Tian, "Super Normal Vector for Human Activity Recognition with Depth Cameras", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. DOI: 10.1109/TPAMI.2016.2565479 (cited on page 47).

[141] Y. Du, W. Wang and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 1110–1118. DOI: 10.1109/CVPR.2015.7298714. arXiv: 1409.7495 (cited on pages 47, 75, 78, 103, 110, 117, 125, 127).

[142] J. Wang, Z. Liu, Y. Wu, J. Yuan, S. S. Member, Z. Liu, S. S. Member, Y. Wu and S. S. Member, "Learning Actionlet Ensemble for 3D Human Action Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 5, pp. 914–927, 2014. DOI: 10.1007/978-3-319-04561-0 (cited on page 47).

[143] R. Vemulapalli and R. Chellappa, "Rolling Rotations for Recognizing Human Actions from 3D Skeletal Data", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4471–4479. DOI: 10.1109/CVPR.2016.484 (cited on page 47).

[144] Z. Huang, C. Wan, T. Probst and L. Van Gool, "Deep Learning on Lie Groups for Skeleton-based Action Recognition", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 1243–1252. DOI: 10.1109/CVPR.2017.137. arXiv: 1612.05877 (cited on page 47).

[145] S. Laraba, M. Brahimi, J. Tilmanne and T. Dutoit, "3D skeleton-based action recognition by representing motion capture sequences as 2D-RGB images", in *Computer Animation and Virtual Worlds*, vol. 28, 2017. DOI: 10.1002/cav.1782 (cited on pages 47, 51).

[146] K. Cho and X. Chen, "Classifying and Visualizing Motion Capture Sequences using Deep Neural Networks", *Computer Vision Theory and Applications (VISAPP)*, to appear, 2014. arXiv: arXiv:1306.3874v2 (cited on page 51).

[147] J. K. Aggarwal and L. Xia, *Human activity recognition from 3D data: A review*, 2014. DOI: 10.1016/j.patrec.2014.04.011 (cited on pages 56, 85, 116).

[148] H. Wang and L. Wang, "Beyond Joints: Learning Representations from Primitive Geometries for Skeleton-Based Action Recognition and Detection", *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4382–4394, 2018. DOI: 10.1109/TIP.2018.2837386 (cited on page 56).

[149] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition", *International Journal of Computer Vision*, vol. 101, no. 3, pp. 420–436, 2013. DOI: 10.1007/s11263-012-0550-7 (cited on page 57).

[150] T. R. Meinard Müller and M. Clausen, "Efficient Content-Based Retrieval of Motion Capture Data", *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 677–685, 2005. DOI: 10.1145/1073204.1073247 (cited on page 57).

[151] L. Seidenari, V. Varano, S. Berretti, A. Del Bimbo and P. Pala, "Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 479–485. DOI: 10.1109/CVPRW.2013.77 (cited on page 58).

[152] R. Slama, H. Wannous, M. Daoudi and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold", *Pattern Recognition*, vol. 48, no. 2, pp. 556–567, 2015. DOI: 10.1016/j.patcog.2014.08.011 (cited on page 58).

[153] J. Liu, A. Shahroudy, D. Xu and G. Wang, "Spatio-temporal LSTM with trust gates for 3D human action recognition", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, 2016, pp. 816–833. DOI: 10.1007/978-3-319-46487-9_50. arXiv: 1607.07043 (cited on pages 60, 75, 110).

[154] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen and X. Xie, "Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks", *Aaai*, no. ii, pp. 3697–3703, 2016. arXiv: 1603.07772 (cited on pages 61, 75, 78, 110).

[155] C. Li, S. Sun, X. Min, W. Lin, B. Nie and X. Zhang, "End-to-end learning of deep convolutional neural network for 3D human action recognition", in *2017 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2017*, 2017, pp. 609–612. DOI: 10.1109/ICMEW.2017.8026281 (cited on page 61).

[156] B. Zhang, L. Wang, Z. Wang, Y. Qiao and H. Wang, "Real-Time Action Recognition with Deeply Transferred Motion Vector CNNs", *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2326–2339, 2018. DOI: 10.1109/TIP.2018.2791180. arXiv: 1604.07669 (cited on page 61).

[157] I. Lee, D. Kim, S. Kang and S. Lee, "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks", in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1012–1020. DOI: 10.1109/ICCV.2017.115 (cited on pages 61, 75, 76, 112, 125).

[158] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks", *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: 10.1109/78.650093. arXiv: arXiv:1011.1669v3 (cited on page 71).

[159] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 28–35. DOI: 10.1109/CVPRW.2012.6239234 (cited on pages 73, 75, 108, 110).

[160] Y. Zhu, W. Chen and G. Guo, "Fusing spatiotemporal features and joints for 3D action recognition", in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 486–491. DOI: 10.1109/CVPRW.2013.78 (cited on pages 75, 76).

[161] R. Anirudh, P. Turaga, J. Su and A. Srivastava, "Elastic functional coding of human actions: From vector-fields to latent variables", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. DOI: 10.1109/CVPR.2015.7298934 (cited on page 75).

[162] Y. Ji, G. Ye and H. Cheng, "Interactive body part contrast mining for human interaction recognition", in *2014 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2014*, 2014. DOI: 10.1109/ICMEW.2014.6890714 (cited on page 75).

[163] W. Li, L. Wen, M. C. Chuah and S. Lyu, "Category-blind human action recognition: A practical recognition system", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015. DOI: 10.1109/ICCV.2015.505 (cited on pages 75, 110).

[164] M. Li and H. Leung, "Multiview Skeletal Interaction Recognition Using Active Joint Interaction Graph", *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2293–2302, 2016. DOI: 10.1109/TMM.2016.2614228 (cited on pages 75, 78).

[165] Y. A. LeCun, Y. Bengio and G. E. Hinton, "Deep learning", *Nature*, vol. 521, no. 7553, pp. 436–444, 2015. DOI: 10.1038/nature14539. arXiv: arXiv:1312.6184v5 (cited on page 84).

[166] 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*, vol. 35, no. 7, pp. 1635–1648, 2013. DOI: 10.1109/TPAMI.2012.253 (cited on page 84).

[167] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017. DOI: 10.1109/TPAMI.2016.2599174. arXiv: 1411.4389 (cited on page 84).

[168] V. F. Mota, E. A. Perez, L. M. Maciel, M. B. Vieira and P. H. Gosselin, "A tensor motion descriptor based on histograms of gradients and optical flow", *Pattern Recognition Letters*, vol. 39, no. 1, pp. 85–91, 2014. DOI: 10.1016/j.patrec.2013.08.008 (cited on pages 85, 96).

[169] M. Al Ghamdi, L. Zhang and Y. Gotoh, "Spatio-temporal SIFT and its application to human action classification", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2012. DOI: 10.1007/978-3-642-33863-2_30 (cited on page 85).

[170] P. Liu, J. Wang, M. She and H. Liu, "Human action recognition based on 3D SIFT and LDA model", in *IEEE SSCI 2011: Symposium Series on Computational Intelligence - RIISS 2011: 2011 IEEE Workshop on Robotic Intelligence in Informationally Structured Space*, 2011. DOI: 10.1109/RIISS.2011.5945790 (cited on page 85).

[171] A. Sargano, P. Angelov and Z. Habib, "A Comprehensive Review on Handcrafted and Learning-Based Action Representation Approaches for Human Activity Recognition", *Applied Sciences*, 2017. DOI: 10.3390/app7010110 (cited on page 86).

[172] S. R. Sreela and S. M. Idicula, "Action recognition in still images using residual neural network features", in *Procedia Computer Science*, 2018. DOI: 10.1016/j.procs.2018.10.432 (cited on page 86).

[173] S. Ji, W. Xu, M. Yang, K. Yu and W. Xu, "3D convolutional neural networks for human action recognition", *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 1, pp. 221–31, 2013. DOI: 10.1109/TPAMI.2012.59 (cited on pages 86, 103).

[174] M. D. Rodriguez, J. Ahmed and M. Shah, "Action MACH: A spatio-temporal maximum average correlation height filter for action recognition", in *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2008. DOI: 10.1109/CVPR.2008.4587727 (cited on page 94).

[175] V. F. Mota, J. I. Souza, A. D. A. Araujo and M. B. Vieira, "Combining orientation tensors for human action recognition", in *Brazilian Symposium of Computer Graphic and Image Processing*, 2013. DOI: 10.1109/SIBGRAPI.2013.52 (cited on page 96).

[176] J. Cho, M. Lee, H. J. Chang and S. Oh, "Robust action recognition using local motion and group sparsity", *Pattern Recognition*, 2014. DOI: 10.1016/j.patcog.2013.12.004 (cited on page 96).

[177] S. Sharma, R. Kiros and R. Salakhutdinov, "Action Recognition using Visual Attention", pp. 1–11, 2016. arXiv: 1511.04119 (cited on page 96).

[178] G. K. Yadav, P. Shukla and A. Sethfi, "Action recognition using interest points capturing differential motion information", in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016. DOI: 10.1109/ICASSP.2016.7472003 (cited on page 96).

[179] J. Monteiro, R. Granada, J. P. Aires and R. C. Barros, "Evaluating the Feasibility of Deep Learning for Action Recognition in Small Datasets", in *Proceedings of the International Joint Conference on Neural Networks*, 2018. DOI: 10.1109/IJCNN.2018.8489297 (cited on page 96).

[180] H. Yang, J. Zhang, S. Li, J. Lei and S. Chen, "Attend It Again: Recurrent Attention Convolutional Neural Network for Action Recognition", *Applied Sciences*, 2018. DOI: 10.3390/app8030383 (cited on page 96).

[181] N. Souly and M. Shah, "Visual Saliency Detection Using Group Lasso Regularization in Videos of Natural Scenes", *International Journal of Computer Vision*, 2016. DOI: 10.1007/s11263-015-0853-6 (cited on pages 96, 97).

[182] H. Wang, M. M. Ullah, A. Klaser, I. Laptev and C. Schmid, "Evaluation of local spatio-temporal features for action recognition", in *Procedings of the British Machine Vision Conference 2009*, 2009. DOI: 10.5244/C.23.124 (cited on pages 96, 97).

[183] P. Weinzaepfel, Z. Harchaoui and C. Schmid, "Learning to track for spatio-temporal action localization", in *Proceedings of the IEEE International Conference on Computer Vision*, 2015. DOI: 10.1109/ICCV.2015.362 (cited on pages 96, 97).

[184] T. Lan, Y. Wang and G. Mori, "Discriminative figure-centric models for joint action localization and recognition", in *Proceedings of the IEEE International Conference on Computer Vision*, 2011. DOI: 10.1109/ICCV.2011.6126472 (cited on page 97).

[185] Z. Cao, T. Simon, S. E. Wei and Y. Sheikh, "Realtime multi-person 2D pose estimation using part affinity fields", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. DOI: 10.1109/CVPR.2017.143 (cited on page 99).

[186] A. Ben Mahjoub and M. Atri, "Human action recognition using RGB data", *International Design and Test Workshop*, pp. 83–87, 2017. DOI: 10.1109/IDT.2016.7843019 (cited on page 99).

[187] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang and P. O. Ogunbona, "Action Recognition from Depth Maps Using Deep Convolutional Neural Networks", *IEEE Transactions on Human-Machine Systems*, 2016. DOI: 10.1109/THMS.2015.2504550 (cited on page 99).

[188] H. Shi, M. Xu and R. Li, "Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN", *IEEE Transactions on Smart Grid*, 2018. DOI: 10.1109/TSG.2017.2686012 (cited on page 103).

[189] J. Liu, G. Wang, P. Hu, L. Y. Duan and A. C. Kot, "Global context-aware attention LSTM networks for 3D action recognition", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 3671–3680. DOI: 10.1109/CVPR.2017.391. arXiv: 1707.05740 (cited on pages 112, 127).

[190] J. Zhang, W. Li, P. O. Ogunbona, P. Wang and C. Tang, "RGB-D-based action recognition datasets: A survey", *Pattern Recognition*, vol. 60, pp. 86–105, 2016. DOI: 10.1016/j.patcog.2016.05.019. arXiv: 1601.05511 (cited on page 116).

[191] H. Wang and L. Wang, "Modeling Temporal Dynamics and Spatial Configurations of Actions Using Two-Stream Recurrent Neural Networks", *Cvpr2017*, pp. 499–508, 2017. DOI: 10.1109/CVPR.2017.387. arXiv: 1704.02581 (cited on pages 117, 118).

[192] Y. Du, Y. Fu and L. Wang, "Representation Learning of Temporal Dynamics for Skeleton-Based Action Recognition", *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3010–3022, 2016. DOI: 10.1109/TIP.2016.2552404 (cited on page 118).

[193] B. Li, Y. Dai, X. Cheng, H. Chen, Y. Lin and M. He, "Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN", in *2017 IEEE International Conference on Multimedia and Expo Workshops, ICMEW 2017*, 2017. DOI: 10.1109/ICMEW.2017.8026282 (cited on page 118).

[194] J. Liu, A. Shahroudy, D. Xu, A. Kot Chichung and G. Wang, *Skeleton-Based Action Recognition Using Spatio-Temporal LSTM Network with Trust Gates*, 2017. DOI: 10.1109/TPAMI.2017.2771306. arXiv: 1706.08276 (cited on page 127).

[195] Q. Ke, M. Bennamoun, S. An, F. Sohel and F. Boussaid, "A new representation of skeleton sequences for 3D action recognition", in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 4570–4579. DOI: 10.1109/CVPR.2017.486. arXiv: 1703.03492 (cited on page 127).

[196] S. Li, W. Li, C. Cook, C. Zhu and Y. Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN", in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. DOI: 10.1109/CVPR.2018.00572 (cited on page 127).

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jun Ren
Name/title of Primary Supervisor:	Napoleon Reyes
Name of Research Output and full reference:	
Ren, J., Reyes, N., Barczak, A., Scogings, C. \& Liu, M. (2018). Toward three-dimensional human action recognition using a convolutional neural network with correctness-vigilant regularizer. Journal of Electronic Imaging, 27(4), 043040.	
In which Chapter is the Manuscript /Published work:	Chapter3
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Jun Ren is the first author of this paper, with 85% contribution. Jun Ren designed and implemented the algorithms, performed the experiments and wrote the papers. Other co-authors reviewed the papers and provided useful feedback.	
For manuscripts intended for publication please indicate target journal:	
Candidate's Signature:	Jun Ren
Date:	22/08/2019
Primary Supervisor's Signature:	
Date:	22/08/2019

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jun Ren
Name/title of Primary Supervisor:	Napoleon Reyes
Name of Research Output and full reference:	
J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings & M. Liu (2018). Towards 3D Human Action Recognition Using a Distilled CNN Model. In 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP) (pp. 7-12). IEEE.	
In which Chapter is the Manuscript /Published work:	Chapter3
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Jun Ren is the first author of this paper, with 85% contribution. Jun Ren designed and implemented the algorithms, performed the experiments and wrote the papers. Other co-authors reviewed the papers and provided useful feedback.	
For manuscripts intended for publication please indicate target journal:	
Candidate's Signature:	Jun Ren
Date:	22/08/2019
Primary Supervisor's Signature:	
Date:	22/08/2019

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jun Ren
Name/title of Primary Supervisor:	Napoleon Reyes
Name of Research Output and full reference:	
Spatio-temporal Kernel based Temporal Convolutional Network for Human Action Recognition	
In which Chapter is the Manuscript /Published work:	Chapter4
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Jun Ren is the first author of this paper, with 85% contribution. Jun Ren designed and implemented the algorithms, performed the experiments and wrote the manuscripts. Other co-authors provided useful feedback.	
For manuscripts intended for publication please indicate target journal:	
IEEE Access	
Candidate's Signature:	Jun Ren
Date:	22/08/2019
Primary Supervisor's Signature:	
Date:	22/08/2019

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jun Ren
Name/title of Primary Supervisor:	Napoleon Reyes
Name of Research Output and full reference:	
J. Ren, N. H. Reyes, A. L. C. Barczak, C. Scogings, & M. Liu (2018). An Investigation of Skeleton-Based Optical Flow-Guided Features for 3D Action Recognition Using a Multi-Stream CNN Model. In 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC) (pp. 199-203). IEEE.	
In which Chapter is the Manuscript /Published work:	Chapter6
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Jun Ren is the first author of this paper, with 85% contribution. Jun Ren designed and implemented the algorithms, performed the experiments and wrote the papers. Other co-authors reviewed the papers and provided useful feedback.	
For manuscripts intended for publication please indicate target journal:	
Candidate's Signature:	Jun Ren
Date:	22/08/2019
Primary Supervisor's Signature:	
Date:	22/08/2019

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)

MASSEY UNIVERSITY
GRADUATE RESEARCH SCHOOL

STATEMENT OF CONTRIBUTION
DOCTORATE WITH PUBLICATIONS/MANUSCRIPTS

We, the candidate and the candidate's Primary Supervisor, certify that all co-authors have consented to their work being included in the thesis and they have accepted the candidate's contribution as indicated below in the *Statement of Originality*.

Name of candidate:	Jun Ren
Name/title of Primary Supervisor:	Napoleon Reyes
Name of Research Output and full reference:	
Action Recognition with CNN Features using LSTM-C RNN Model	
In which Chapter is the Manuscript /Published work:	Chapter5
Please indicate:	
• The percentage of the manuscript/Published Work that was contributed by the candidate:	85%
and	
• Describe the contribution that the candidate has made to the Manuscript/Published Work:	
Jun Ren is the first author of this paper, with 85% contribution. Jun Ren designed and implemented the algorithms, performed the experiments and wrote the manuscript. Other co-authors provided useful feedback.	
For manuscripts intended for publication please indicate target journal:	
Computer Vision and Image Understanding	
Candidate's Signature:	Jun Ren
Date:	22/08/2019
Primary Supervisor's Signature:	
Date:	22/08/2019

(This form should appear at the end of each thesis chapter/section/appendix submitted as a manuscript/ publication or collected as an appendix at the end of the thesis)