MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

Deep Learning for Action Recognition in Videos

A thesis presented in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

Massey University, Albany, Auckland,

New Zealand

Ming Zong

2021

Abstract

Video action recognition is a difficult and challenging task in video processing. In this thesis, we propose three novel deep learning approaches to improve the accuracy of action recognition.

The first approach aims to learn multi-cue based spatiotemporal features by performing 3D convolutions. Previous 3D CNN models mainly perform 3D convolutions on individual cues (*e.g.*, appearance and motion cues), which lacks the effective overall integration of the appearance information and motion information of videos. To address this issue, we propose a novel multi-cue 3D convolutional neural network (named M3D model for short), which integrates three individual cues (*i.e.* an appearance cue, a direct motion cue, and a salient motion cue) directly. The proposed M3D model directly performs 3D convolutions on multiple cues instead of a single cue, which can obtain more discriminative and robust features by integrating three different cues as a whole. In particular, we propose a novel deep residual multi-cue 3D convolution model (named R-M3D for short) to enhance the representation ability by benefitting from the increasing depth of the model, which can obtain more representative spatiotemporal features.

The second approach aims to utilize the motion saliency information to enhance the accuracy of action recognition. We propose a novel motion saliency based multi-stream multiplier ResNets (named MSM-ResNets for short) for action recognition. The proposed MSM-ResNets model consists of three interactive streams: the appearance stream, motion stream and motion saliency stream. The appearance stream is responsible for capturing the appearance information with RGB video frames as input. The motion stream is responsible for capturing the motion information with optical flow frames as input. The motion saliency stream is responsible for capturing the salient motion information with motion saliency frames as input. In particular, to utilize the complementary information between different streams over time, the proposed MSM-ResNets model establishes multiplicative connections between different streams. Two kinds of different multiplicative connections are injected, the first one is to inject multiplicative connections to transmit the motion cue from the motion stream to the appearance stream, and the second one is to inject multiplicative connections to transmit the motion saliency cue from the motion saliency stream to the motion stream.

The third approach aims to explore the salient spatiotemporal information over time evolution. We propose a novel spatial and temporal saliency based four-stream network with multi-task learning (named 3M model for short) for action recognition. The proposed 3M model comprises two parts: (i) The first part is a spatial and temporal saliency based four-stream network, which comprises four streams: an appearance stream, a motion stream, a novel spatial saliency stream and a novel temporal saliency stream. The novel spatial saliency stream is used to acquire spatial saliency information and the novel temporal saliency stream is used to acquire temporal saliency information. (ii) The second part

is a multi-task learning based long short-term memory network (LSTM), which adopts the feature representations obtained by obtained convolutional neural networks (CNN) as input. The multi-task learning based LSTM can share the complementary knowledge between different streams and capture the long-term dependency relationships of consecutive frames.

Experiments verify the effectiveness of all the proposed models and show that all the proposed models achieve a better performance than the state-of-the-art.

Acknowledgements

I would like to take this opportunity to express my deepest gratitude to all the people who have supported me on my PhD journey to achieve this qualification.

Firstly, I express my sincerest gratitude and respect to my main supervisor Prof. Ruili Wang, and co-supervisor Prof. Xun Wang, for their invaluable academic guidance and support throughout my whole PhD study. They provide many constructive feedbacks to improve my research. Especially, many thanks are given for my main supervisor Prof. Wang, who has spent dedicated time and efforts in helping me improve my research ability. Except academic guidance, my supervisor Prof. Wang also taught me a lot of life lessons when I encountered difficulties. Without his help, I cannot success in finishing my PhD study. I also would like to thank to my co-supervisor Prof. Xun Wang for his warm supporting when I studied in Zhejiang Gongshang University. In addition, I would like to thank to other teachers (*e.g.* Dr Yan Tian and Dr Andrew Gilman) who have given advices for my research.

I would like to thank my friends in Prof. Ruili Wang's research team for their help and the good times.

I greatly acknowledge to the China Scholarship Council and the New Zealand China Doctoral Research Scholarships Program, which provide the financial support to help me finish my PhD study. I also would like to thank to the staff members of New Zealand Scholarships team (*e.g.* Anita and Jamie) and the faculty members at the School of Natural and Computational Sciences, for their help and support.

I am also very grateful to my previous supervisor Prof. Shichao Zhang during my Master study, who guided me into the machine learning research field and provided me with good academic guidance throughout my whole Master study. Because of his recommendation, I could meet Prof. Ruili Wang and get the opportunity to pursue my PhD at Massey University, where I have broadened my horizon and finished my PhD study.

Lastly, I would like to thank everyone who helped me and my parents for their unconditional love, understanding and support.

Publications

The following research papers have been published in or submitted to International Journals and Conferences during my PhD study:

1. **Ming Zong**, Ruili Wang, Zhe Chen, Maoli Wang, Xun Wang and Johan Potgieter. Multi-cue based 3D residual network for action recognition. *Neural Computing & Applications*. doi: https://doi.org/10.1007/s00521-020-05313-8, 2020. (Refer to Chapter 2)

2. **Ming Zong**, Ruili Wang, Xiubo Chen, Zhe Chen and Yuanhao Gong. Motion saliency based multi-stream multiplier ResNets for action recognition. *Image and Vision Computing*. doi: https://doi.org/10.1016/j.imavis.2021.104108, 2021. (Refer to Chapter 3)

3. **Ming Zong**, Ruili Wang and Wanting Ji. Spatial and temporal saliency based four-stream network with multi-task learning for action recognition. Submitted to *IEEE Transactions on Circuits and Systems for Video Technology*. (Refer to Chapter 4)

4. Ruili Wang and **Ming Zong**. Laplacian Eigenmaps based manifold regularized CNN for action recognition. Submitted to *IEEE Transactions on Cybernetics*.

5. Zhenbing Liu, Zeya Li, Ruili Wang, **Ming Zong** (corresponding author) and Wanting Ji. Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. *Neural Computing & Applications*. 32(18), 14593-14602, 2020.

6. Zhenbing Liu, Zeya Li, **Ming Zong** (corresponding author), Wanting Ji and Yan Tian. Spatiotemporal saliency based multi-stream networks for action recognition. In *Asian Conference on Pattern Recognition Workshops*, 2019.

7. Ruili Wang and **Ming Zong**. Joint self-representation and subspace learning for unsupervised feature selection. *World Wide Web*, 21(6), 1745-1758, 2018.

8. Hao Zheng, Ruili Wang, Wanting Ji, **Ming Zong**, Wai Keung Wong, Zhihui Lai and Hexin Lv. Discriminative deep multi-task learning for facial expression recognition. *Information Sciences*, 533, 60-71, 2020.

9. Zhe Chen, Ruili Wang, Wanting Ji, **Ming Zong**, Tanghuai Fan and Huibin Wang. A novel monocular calibration method for underwater vision measurement. *Multimedia Tools and Applications*, 78(14), 19437-19455, 2019.

10. Junbo Ma, Ruili Wang, Wanting Ji, Jiawei Zhao, **Ming Zong** and Andrew Gilman. Robust multi-view continuous subspace clustering. Pattern Recognition Letters. doi: https://doi.org/10.1016/j.patrec.2018.12.004, 2018.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter provides an overview of this thesis. We first briefly introduce action recognition in Section 1.1. Then the motivation of this research is presented in Section 1.2. Finally, the organization of this thesis is presented in Section 1.3.

1.1 Overview of action recognition

In computer vision, action recognition refers to the act of classifying a human action that is present in a given video to one of the pre-defined set of actions [11]. Over the past decades, this research has captured much attention due to its strength in providing personalized support for many different applications and its connection to many different fields of study such as medicine, human-computer interaction, or sociology. Currently, human action recognition has been applied to many video-related applications, such as video surveillance, abnormal event detection, and hospital patient care [4, 20, 26].

Many research methods have been developed for action recognition in the past decades, which can be roughly divided into two categories as follows:

1. **Shallow representation methods for action recognition:** Shallow representation based methods mainly extract features through one hidden layer of nonlinear transformation [3]. Existing methods can be roughly classified into two categories as follows:

    - **Global feature representation based methods**: Global feature representation based methods usually extract features from global representations of human body structures and shapes [2, 6]. Generally, the motion information of a video will be captured firstly. After that, a certain template, which is predefined by pixels intensities information, is used to filter out the background

information and the human action information. Early work for human action recognition is based on global feature representation, which began from 1980s [10], while local feature representation based methods roughly began from 2000s [8]. The most influential work of global feature representation based methods was proposed by Bobick and Davis [2]. They developed two templates, which named Motion Energy Image (MEI) and Motion History Image (MHI). MEI is a motion based object recognition template while MHI is a vision based template. After that, Blank *et al.* [6] extended the MEI template by using a 3D action description to replace the previous 2D descriptions. In 2006, Weinland *et al.* [25] extended the previous 2D MHI templates into a 3D volume. The advantage of global feature representation based methods is that they can preserve the basic spatial and temporal structures of actions. However, they fail to capture fine details with the silhouette and not robust to noises.

- **Local feature representation based methods**: Local feature representation based methods usually consist of three steps [22, 23]: (i) detecting interest points from a video [17], (ii) extracting local descriptors from the video, and (iii) classifying human actions according to these descriptors. Laptev and Lindeberg *et al.* [13] proposed to detect spatiotemporal interest points to capture local image features for video interpretation. Schüldt *et al.* [15] proposed to use spatiotemporal interest points as local features and SVM is used as a classifier for action recognition. Scovanner *et al.* [16] proposed to 3D SIFT descriptor to encode spatiotemporal information for action recognition. Kläser *et al.* [12] proposed 3D gradient orientation descriptor based on HoG-based descriptors for action recognition. Improved dense trajectories (iDT) [23] is one of the classic local feature representation based methods for action recognition, which improves the performance by removing background trajectories. The advantage of local feature representation based methods is that they are robust to the background, occlusion and illumination. However, these representations lack the ability of deep abstract representation.

2. **Deep representation methods for action recognition:** In recent years, deep learning has achieved a great success in image processing and video processing [7, 19]. Action recognition also benefits from deep learning and obtains an obvious improvement [8, 29]. Existing deep representation methods for action recognition can be roughly divided into three categories as follows:

   - **Two-steam convolutional neural networks based model (two-stream CNNs based model)** [5, 18, 28]: Two-stream convolutional neural networks consist of spatial stream and temporal stream. Video frames are extracted from the action video and then they will be fed into the spatial stream for capturing the appearance information. The corresponding optical flow frames are extracted from raw RGB frames and then they will be fed into the temporal stream for capturing the motion information. The outputs of both the spatial

stream and temporal stream will be fused together for predicting the action category. The advantage of two-stream CNNs is that it can explicitly capture the appearance information and motion information of a video using two separate convolutional neural networks. *The related work of two-stream CNNs based model for action recognition can be found in Section 3.2.2 and Section 4.2.1.*

- **3D convolutional neural network based model (3D CNN based model)** [21]: 3D convolutional neural networks operate convolution operations on both spatial dimension and temporal dimension simultaneously instead of only spatial dimension, which can directly learn spatiotemporal features. The input of 3D CNN is not a single video frame or all video frames, but a few consecutive video frames. The convolution kernel of 3D CNN is a three-dimensional cube instead of a two-dimensional convolution kernel in CNN. The advantage of 3D CNN is that it can seamlessly capture spatiotemporal features along the spatial dimension and temporal dimension at the same time. *The related work of 3D CNN based model for action recognition can be found in Section 2.2.2.*
- **Recurrent neural network based model (RNN-based model)** [1, 27]: Recurrent neural network (RNN) is a kind of neural network for processing sequence data, which has been successfully applied in many sequence tasks such as speech recognition, language modelling and machine translation [9]. A common architecture of RNN-based model for action recognition usually consists of a CNN and an RNN. First, CNN is used to extract CNN features from raw video frames. Then the extracted CNN features are fed into an RNN for action recognition. The advantage of RNN based methods for action recognition is that RNN is a natural architecture for processing the sequence data, such as video data [14, 24]. *The related work of RNN-based model for action recognition can be found in Section 4.2.2.*

1.2 Motivations of this research

Action recognition is an important research topic in computer vision, which has been successfully applied to many vision-related practical applications such as human-computer interaction and smart video surveillance. However, the accuracy of current action recognition methods is still not on par with the human. Thus, we cannot apply current action recognition methods to the applications mentioned above. This motivates us to conduct research in action recognition.

Since action recognition is a classification problem, our research aim is to increase the classification accuracy of action recognition using deep learning approaches. Currently, deep learning approaches applied in action recognition are normally conducted from the following five aspects:

1. *The front-end input of networks:* Since videos contain multi-modal information such as the appearance information, motion information and acoustic information, extracting different useful information from the input videos is useful for action recognition.
2. *The design of network structure:* Different designs of network structure can capture different spatiotemporal features from different angles. Thus, a good network structure is important for action recognition.
3. *The fusion of spatial information and temporal information:* More effective spatiotemporal features, more higher accuracy of action recognition. The effective fusion methods of spatial information and temporal information can obtain effective spatiotemporal features.
4. *The back-end loss function of networks:* Loss function is used to evaluate the difference between the predicted action label and the actual action label. It is valuable to research and design a loss function that can measure video actions well.
5. *Background noise denoising:* Videos usually contain various background environments, which may impair the accuracy of action recognition. Thus, background noise denoising is important for improving action recognition.

To improve the accuracy of action recognition, three research objectives underpin this research aim:

- *Objective 1 will develop a multi-cue based 3D residual network for action recognition.* Current 3D CNN methods mainly perform 3D convolutions on individual cues (for example, perform 3D convolutions on the appearance cue or the motion cue). However, this manner lacks the effective overall integration of the appearance information and motion information. To address these issues, we propose to integrate three individual cues (*i.e.* an appearance cue, a direct motion cue, and a salient motion cue) together as an input, and directly perform 3D convolutions on a multiple cue based input instead of a single cue based input.
- *Objective 2 will develop a motion saliency based multi-stream multiplier ResNets for action recognition.* Current action recognition methods usually treat each moving pixel equally and ignore paying attention on the salient moving pixels, that is the salient actions. To address this issue, we propose a motion saliency stream with motion saliency maps as input to capture the salient motion information. In particular, to utilize the complementary information between different streams over time, we establish connections between different streams. Two kinds of different connections are injected, the first one is to inject connections to transmit the motion cue from the motion stream to the appearance stream, and the second one is to inject multiplicative connections to transmit the motion saliency cue from the motion saliency stream to the motion stream.
- *Objective 3 will develop a spatial and temporal saliency based four-stream network with multi-task learning for action recognition.* There are two key issues in current

two-stream based models, the first issue is how to utilize the salient information for action recognition, and the second issue is how to leverage full advantage of the complementary knowledge between different streams. To address these issues, we propose a spatial saliency stream and a temporal saliency stream to capture the corresponding spatial and temporal salient information for action recognition. In addition, to share the complementary knowledge between different streams and capture the long-term dependency relationships between consecutive frames, we propose a multi-task learning based long short-term memory network (LSTM). It uses the CNN feature representations of each stream as the input, and performs multi-task learning by summing all the losses of all the streams together.

1.3 Organization of this thesis

Literature reviews of the deep learning based action recognition methods are presented in each chapter corresponding to the proposed three novel deep learning approaches.

The rest of this thesis is organized as follows:

Chapter 2 presents the proposed multi-cue 3D convolutional neural network.

Chapter 3 presents the proposed motion saliency based multi-stream multiplier ResNets.

Chapter 4 presents the proposed spatial and temporal saliency based four-stream network with multi-task learning.

Chapter 5 concludes this thesis and discusses the future work.

Note that references related to each chapter are listed at the end of each chapter.

References

[1] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.

[2] Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[3] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019.

[5] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[6] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2247–2253, 2007.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Samitha Herath, Mehrtash Harandi, and Fatih Porikli. Going deeper into action recognition: A survey. *Image and Vision Computing*, 60:4–21, 2017.

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] David Hogg. Model-based vision: a program to see a walking person. *Image and Vision Computing*, 1(1):5–20, 1983.

[11] Soo Min Kang and Richard P Wildes. Review of action recognition and detection methods. *arXiv preprint arXiv:1610.06906*, 2016.

[12] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3D-gradients. 2008.

[13] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005.

[14] Junhua Mao, Wei Xu, Yi Yang, Jiang Wang, Zhiheng Huang, and Alan Yuille. Deep captioning with multimodal recurrent neural networks (m-RNN). *arXiv preprint arXiv:1412.6632*, 2014.

[15] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.

[16] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM International Conference on Multimedia*, pages 357–360, 2007.

[17] Ling Shao, Xiantong Zhen, Dacheng Tao, and Xuelong Li. Spatio-temporal laplacian pyramid coding for action recognition. *IEEE Transactions on Cybernetics*, 44(6):817–827, 2013.

[18] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[20] Yan Tian, Yifan Cao, Jiachen Wu, Wei Hu, Chao Song, and Tao Yang. Multi-cue combination network for action-based video classification. *IET Computer Vision*, 13(6):542–548, 2019.

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[22] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176. IEEE, 2011.

[23] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[24] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[25] Daniel Weinland, Remi Ronfard, and Edmond Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2006.

[26] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 305–321, 2018.

[27] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[28] Ming Zong, Ruili Wang, Xiubo Chen, Zhe Chen, and Yuanhao Gong. Motion saliency based multi-stream multiplier ResNets for action recognition. *Image and Vision Computing*, (https://doi.org/10.1016/j.imavis.2021.104108), 2021.

[29] Ming Zong, Ruili Wang, Zhe Chen, Maoli Wang, Xun Wang, and Johan Potgieter. Multi-cue based 3D residual network for action recognition. *Neural Computing and Applications*, (https://doi.org/10.1007/s00521-020-05313-8), 2020.

Chapter 2

Multi-cue based 3D residual network for action recognition

Convolutional neural network (CNN) is a natural structure for video modelling that has been successfully applied in the field of action recognition. The existing 3D CNN based action recognition methods mainly perform 3D convolutions on individual cues (e.g., appearance and motion cues) and rely on the design of subsequent networks to fuse these cues together. In this chapter, we propose a novel multi-cue 3D convolutional neural network (M3D), which integrates three individual cues (i.e. an appearance cue, a direct motion cue, and a salient motion cue) directly. Different from the existing methods, the proposed M3D model directly performs 3D convolutions on multiple cues instead of a single cue. Compared with the previous methods, this model can obtain more discriminative and robust features by integrating three different cues as a whole. Further, we propose a novel residual multi-cue 3D convolution model (R-M3D) to improve the representation ability to obtain representative video features. Experimental results verify the effectiveness of the proposed M3D model, and the proposed R-M3D model (pre-trained on the Kinetics dataset) achieves competitive performance compared with the state-of-the-art models on UCF101 and HMDB51 datasets.

The organization of this chapter is as follows. In Section 2.1, we introduce the motivation of the proposed approach. The background and related work are introduced in Section 2.2. Then two novel multi-cue 3D convolutional neural network (M3D and R-M3D) are proposed in Section 2.3. Experimental results and analysis will be discussed in Section 2.4. Lastly, a conclusion and future work will be summarized in Section 2.5.

2.1 Introduction

Human action recognition aims to automatically identify specified actions in a video [28, 29]. It has many applications such as intelligent video surveillance, human-computer interaction, human behaviour analysis, and smart hospital care [31]. Different from images that only contain an appearance cue, videos contain not only the appearance cue extracted from still video frames but also a motion cue extracted from stacked video frames. Therefore, the motion cue plays an important role in action recognition.

Compared with traditional shallow hand-crafted models [40, 41], convolutional neural networks (CNNs) [46] have shown a superior ability to capture appearance information in many visual related tasks such as image classification [26, 45], object detection [13], and image segmentation [30].

To take advantage of CNNs, many 2D CNN-based methods [4, 9, 24, 34, 38, 44] were developed for action recognition. 2D CNN-based action recognition models can be roughly divided into two categories: (i) frame-based aggregation models [4, 9, 24, 44], and (ii) two-stream CNN-based models [34]. Frame-based aggregation models use CNNs to extract features from each frame and then aggregate frame-level information to obtain video-level information by using different aggregating strategies such as recurrent neural networks (RNNs). However, such models ignore the temporal structure and mainly rely on subsequent aggregation strategies for obtaining a motion cue. Two-stream CNN-based models consist of a spatial CNN stream and a temporal CNN stream for capturing the appearance cue and motion cue, respectively. Optical flow based methods usually are adopted for extracting the motion features, which can effectively provide the velocity information (including the speed and direction) of each pixel. However, such models lack effective information interaction over time between the appearance cue and motion cue.

Recently, a variety of 3D CNN-based action recognition models have been developed for modelling spatio-temporal features [22, 36], which has shown more promising results than the previous CNN-based models for action recognition on a sufficiently large video dataset such as Kinetics dataset [4, 16]. According to the input cue, we categorize the existing 3D CNN-based action recognition models into three classes: (i) single-cue 3D CNN model [24], (ii) two-cue 3D CNN model [34] and (iii) three-cue 3D CNN model [22].

- **Single-cue 3D CNN model:** The input of a single-cue 3D CNN model is stacked video frames that only provide an appearance cue at the input level. The motion information is indirectly obtained by 3D convolutions through the inferences between stacked video frames. Thus, such models lack the ability to provide direct motion cues at the input level [34].
- **Two-cue 3D CNN model:** Two-cue 3D CNN models [4, 39] perform 3D convolutions on video frames and optical flow frames separately, and then the obtained appearance and motion cues are integrated by using various fusion strategies. However, since the 3D convolutions are separately operated on individual cues respectively

and the fusion relies on specific designed networks, this model lacks the overall integration of the appearance information and motion information of videos.

- **Three-cue 3D CNN model:** Three-cue 3D CNN models [22] generate different cues (such as gray, gradient, and optical flow) from stacked video frames, and performed 3D convolutions on these cues separately. Then a full connection layer is applied to combine all cues. Late fusion is also adopted for aggregating different cues for action recognition. Although optical flow [3, 34] is capable of capturing motion information (*i.e.* speed and direction information) directly, it mainly focuses on instantaneous motion velocity information that is sensitive to background motion noises such as slight leaf jittering and water rippling [19, 32].

To overcome the above challenges, we propose a novel multi-cue 3D convolutional neural network (M3D). The proposed M3D model integrates an appearance cue, a direct motion cue and a salient motion cue simultaneously as the input. The appearance cue can extract features from still video frames, the motion cue can extract features from stacked video frames, and the salient motion cue can effectively suppress background motion noises and highlight salient motion. Different from the previous action recognition methods (single-, two-, and three-cue CNN models), our proposed M3D model directly performs 3D convolutions on a multi-cue input instead of performing 3D convolutions on different single-cue inputs separately. Further, we propose a novel residual multi-cue 3D convolution model (R-M3D) to improve the representation ability to obtain representative video features.

The key contributions of this chapter can be summarized as follows:

- This chapter proposes a novel M3D model, which first directly performs 3D convolutions on a multi-cue input instead of a single-cue input.
- This chapter develops a novel video frame representation for performing 3D convolutions on a multi-cue input.
- The proposed M3D model adds the salient motion cue for action recognition, which can suppress background motion noises and highlight salient motion.
- To further improve the representation ability, we propose the deeper R-M3D model based on 3D ResNet, which can significantly improve the performance of action recognition.

2.2 Background and related work

A video contains both time and space information. With the development of deep learning techniques, deep neural network-based action recognition methods, especially CNN based action recognition methods, have obtained better performances than the conventional shallow bag-of-visual-words based models [22]. 2D CNNs and 3D CNNs are two types of neural

networks commonly used in CNN based action recognition methods.

Section 2.2.1 first introduces some 2D CNN-based action recognition works. Then we present the general 3D CNNs architecture and some 3D CNN-based action recognition works in Section 2.2.2. Finally, we present the principle of 3D convolutions performed on a single-cue input for action recognition in Section 2.2.3.

2.2.1 2D CNN for action recognition

2D CNN-based action recognition methods usually extract features from single video frame using 2D CNNs, and then aggregate temporal information across video frames using different fusion strategies. In general, these methods average the output of the utilized 2D CNNs on each frame and utilise the average frame-level result to represent video-level results [24]. However, these methods cannot make full use of video information. They only extract appearance information from videos while ignoring motion information between video frames.

To capture temporal motion information between video frames, two-stream CNN-based models were developed for action recognition. They usually consist of two streams. The spatial stream is responsible for capturing the appearance cue and the temporal stream is responsible capturing the motion cue. Then various fusion strategies are adopted to aggregate the output of these two streams [10–12, 34]. In addition, some methods utilize recurrent neural networks (RNN) [4, 9, 44], especially Long Short-Term Memory (LSTM) networks, were utilized to capture temporal information between stacked video frames. A pipeline of RNN-based action recognition methods is that (i) 2D CNNs are used for extracting features from each video frame, and (ii) LSTM networks are used for encoding states and learning temporal relations between stacked video frames.

2.2.2 3D CNN for action recognition

In contrast to 2D CNN for action recognition, some action recognition methods utilize 3D CNNs to model spatio-temporal features from videos. A common architecture of 3D CNNs for action recognition can be illustrated in Figure 2.1, which consists of five convolutional layers, two fully-connected layers, and a softmax layer. Different from the input of 2D CNNs that is a single frame, the input of 3D CNNs is stacked video frames. Different from 2D CNNs, which only slide a convolution filter kernel along the spatial direction, the 3D convolution filter kernel slides along with both spatial and temporal directions simultaneously. Thus, 3D CNNs can naturally capture spatio-temporal information, while 2D CNNs need to aggregate multiple frame-level information over long time periods to capture the temporal information of videos.

It has been proven that 3D CNN-based models have shown competitive performance than

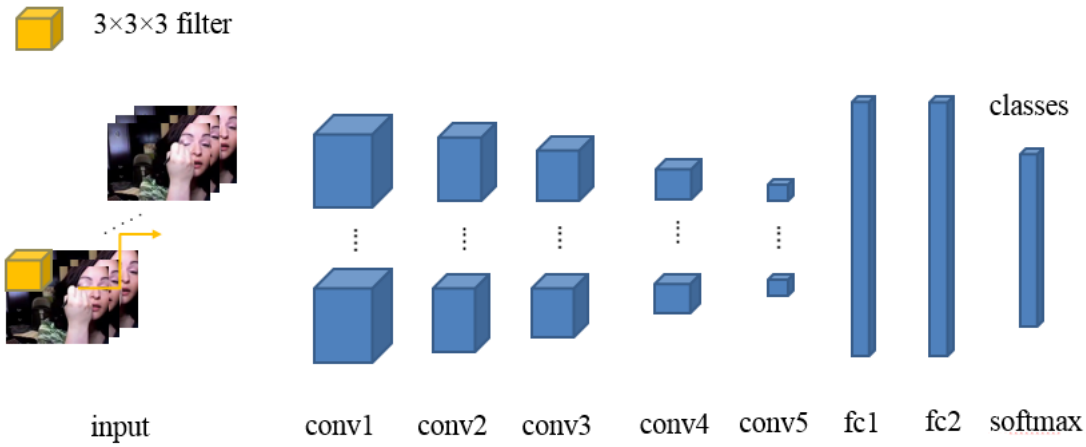

Figure 2.1: The architecture of 3D CNN for action recognition, which consists of five convolutional layers, two fully-connected layers, and a softmax layer. The kernel size is $3 \times 3 \times 3$.

2D CNN-based models [16]. Early work of 3D CNN based action recognition models were proposed by Baccouche *et al.* [2], which extracted 3D CNN-based features for each frame and input them sequentially into an LSTM network for classification. Ji *et al.* [22] proposed to separately perform 3D convolutions on multiple channels. Then, a late fusion was used for all these channels. These 3D CNNs are not obviously deep, which typically contain 3 convolutional layers at most.

Recently, a variety of deeper 3D CNN-based models were developed. Karpathy *et al.* [24] proposed a slow fusion strategy to aggregate frame-level information across temporal domain. Tran *et al.* [36] proposed to use a small $3 \times 3 \times 3$ convolution kernel in all convolutional layers and got better performance than other large sizes of convolution kernels. Similar to [5], 3D convolutions were performed separately on a spatial stream and a temporal stream in [4], the output results of two streams were aggregated later. Hara *et al.* [15] proposed to use a very deep 3D residual network instead of shallow 3D convolutional networks. Further, deep 3D CNNs architectures on sufficiently large datasets can get better performances than those complex 2D CNNs architectures [16]. Arunnehru *et al.* [1] extracted motion information from the video and regarded them as 3D motion cuboid and then applied a 3D convolutional neural network for action recognition. Zhou *et al.* [48] proposed a novel Mixed Convolutional Tube to combine 2D CNNs with the 3D convolution together for action recognition, which can generate more discriminative feature maps. Qiu *et al.* [33] considered using a $1 \times 3 \times 3$ spatial convolution kernel plus a temporal $3 \times 1 \times 1$ convolution kernel to simulate the previous $3 \times 3 \times 3$ convolution kernel, which is helpful for reducing the computational cost and training deeper neural networks. Based on 3D ResNets, Tran *et al.* [37] decomposed 3D convolution kernel into spatial convolution and temporal convolution and designed a novel spatiotemporal convolutional block R(2+1)D, which can achieve superior performance for action recognition compared with the state-of-the-art.

Although current 3D convolution models can achieve competitive performance, most of them only directly perform 3D convolutions on a single-cue input, *e.g.*, on stacked video frames, and optical flow frames, separately. To obtain richer information in the input level such as motion velocity and motion saliency, we propose to directly perform 3D convolutions on a multi-cue input in Section 2.3.3.

2.2.3 3D convolutions performed on a single-cue input

Current 3D convolutions usually are performed on a single-cue input, *i.e.* stacked video frames or stacked optical flow frames. Specifically, taking stacked video frames as an example, we denote n stacked video frames as $\{vf_1, vf_2, ..., vf_n\}$. An image in the RGB format usually consists of three different colour channels: the red channel, green channel, and blue channel. Thus each video frame vf can be represented as $\{vf\_r, vf\_g, vf\_b\}$, where vf_r represents a red channel; vf_g represents a green channel, and vf_b represents a blue channel. Correspondingly, the stacked video frames can be represented as $\{vf_1\_r, vf_1\_g, vf_1\_b, ..., vf_n\_r, vf_n\_g, vf_n\_b\}$, where n denotes the number of input video frames. The number of total channels of the input stacked video frames is $n \times 3$. The computation process of 3D convolutions performed on a single-cue input (*i.e.* stacked video frames) is illustrated in Figure 2.2.

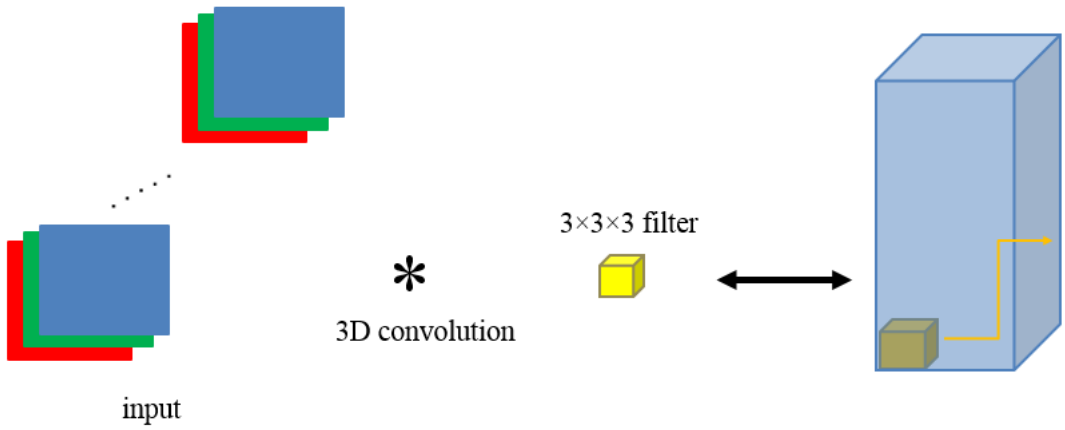

Figure 2.2: The computation process of 3D convolutions performed on a single-cue input, *i.e.* stacked video frames. The red rectangle represents the red colour channel, the green rectangle represents the green colour channel, and the blue rectangle represents the blue colour channel.

In the temporal dimension, as a $3 \times 3 \times 3$ filter kernel is applied, total 9 colour channels are involved for each computation process of 3D convolutions performed on a single-cue input. Three stacked video frames are involved in each computation, and each video frame contains three different colour channels. Since 3D convolutions are only applied to the

single-cue input, *i.e.* stacked video frames, it lacks the ability to provide direct motion information in the input level. For two-cue 3D CNN models, they still separately perform 3D convolutions on individual single-cue input such as stacked video frames or stacked optical flow frames, and then a late fusion is adopted. However, late fusion is lack of cues evolution over time.

2.3 Our models: M3D & R-M3D

In this section, we will first present a motion saliency detection method to suppress background motion noises and provide salient motion cue in Section 2.3.1, which will be used in our proposed M3D model. Then we present the novel triple video representation in Section 2.3.2. Based on novel triple video representation, 3D convolutions performed on a multi-cue input is proposed in Section 2.3.3. Finally, a novel M3D model is presented in Section 2.3.4 and a novel deep R-M3D model is presented in Section 2.3.5.

2.3.1 Motion saliency detection

Saliency detection aims to detect the salient object from the whole image. For example, Ji *et al.* [23] proposed to use an attention CNN layer to capture the context information between different feature maps to improve the quality of obtained saliency maps. In contrast to traditional saliency detection methods that identify salient objects on a still image, motion saliency detection methods [5, 6, 8, 43] focus on identifying salient motion information (named motion saliency for short [6]) from a video. However, conventional motion saliency detection methods have expensive time cost and are not suitable for large video datasets. Compared with conventional motion saliency detection methods [7, 17, 21], spectrum-based motion saliency detection methods have a good balance between performance and computation cost [6].

The spectral theory has been widely used in the saliency detection field [6, 8, 14, 20]. The first spectrum-based saliency detection work was proposed by Hou *et al.* [20], which used amplitude spectral residual of an image to represent the novelty part. Guo *et al.* [14] proposed to use the phase spectrum of an image to represent the novelty part instead of the amplitude spectrum. According to Spectral Residual (SR) theory [20], the image information contains the novelty part and the redundant information. Therefore, an image can be represented as follows [20]:

$$\gamma(Image) = \gamma(Innovation) + \gamma(PriorKnowledge) \tag{2.1}$$

where $\gamma(Image)$ represents the image information; $\gamma(PriorKnowledge)$ represents the redundant information, which denotes the statistical invariant properties in the image; $\gamma(Innovation)$ represents the novelty part, which denotes the statistical variant properties

in the image [20]. In the field of motion saliency detection, $\gamma(Innovation)$ corresponds to the foreground objects, and $\gamma(PriorKnowledge)$ corresponds to the background [6, 14].

In this section, a phase spectrum-based motion saliency detection method [6] is introduced for capturing the salient motion information and suppressing the background motion noises. It distinguishes the salient motion and background motion noises by identifying phase spectrum variations of each pixel through a temporal Fourier transform. The key procedure of this method can be summarized as follows:

Procedure:	Phase spectrum-based motion saliency detection.
Step1:	Establishing a temporal sequence $S_{x,y}(t)$ for each pixel in the same position (x, y) through stacked video frames, here the size of a video frame is $M \times N$, $x = 1, 2, ..., M$, $y = 1, 2, ..., N$, and $t = 1, 2, ..., T$. Thus $M \times N$ temporal sequences are established.
Step2:	Calculating the Fourier transform for each temporal sequence $S_{x,y}(t)$: $f_{x,y}(t) = F(S_{x,y}(t))$, here F denotes the Fourier transform.
Step3:	Calculating phase spectrum $p_{x,y}(t)$ for each temporal sequence $S_{x,y}(t)$: $p_{x,y}(t) = angle(f_{x,y}(t))$, here $angle$ denotes a function of obtaining phase values of a temporal sequence.
Step4:	Calculating inverse Fourier transform $f^{-1}_{x,y}(t)$ for each temporal sequence $S_{x,y}(t)$: $f^{-1}_{x,y}(t) = g(t) * F^{-1}(p_{x,y}(t))$, here F^{-1} denotes inverse Fourier transform and $g(t)$ denotes a one-dimensional Gaussian filter for smoothing noises.

Through the above steps, the salient motion can be identified by obvious phase spectrum variations, and noises can be identified by slight phase spectrum variations. Therefore, background and background motion noises are easily suppressed because the corresponding values of the phase spectrum are much smaller than the values produced by salient motions. Figure 2.3 illustrates a video frame extracted from a makeup video clip from the UCF101 dataset [35], and its corresponding optical flow and motion saliency results.

2.3.2 Novel triple video representation

A traditional video representation consists of stacked video frames, *i.e.* stacked video frames are used to describe video information. However, this video representation manner

Figure 2.3: The left picture denotes a video frame; the middle picture denotes the corresponding optical flow result, and the right picture denotes the corresponding motion saliency result.

only can provide the appearance cue as input for current single-cue 3D CNN models in action recognition, which lacks the ability to provide a direct motion cue in the input level.

For two-cue 3D CNN models, they usually adopt video frames and optical flow frames as input to provide the appearance cue and the direct motion cue, *i.e.* they use video frame and optical flow to describe video information. However, since they separately use video frames and optical flow frames as input to operate 3D convolutions, this manner relies on specific designed networks to fuse the appearance cue and motion cue instead of directly operating 3D convolutions on a multi-cue input. Besides, optical flow-based motion detection methods are sensitive to background motion noises.

To overcome the aforementioned challenges in current 3D CNN models, we develop a novel triple video representation composed of the original video frame, the corresponding optical flow and motion saliency, which is illustrated in Figure 2.4. The proposed triple video representation integrates an appearance cue, a direct motion cue and a salient motion cue simultaneously as the input. The appearance cue can extract features from still video frames, the motion cue can extract features from stacked video frames, and the salient motion cue can effectively suppress background motion noises and highlight salient motion.

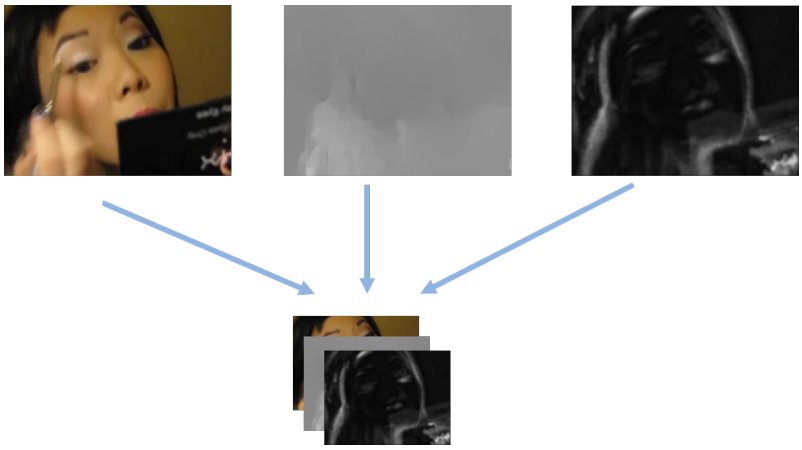

Figure 2.4: The triple video representation consists of video frame, optical flow and motion saliency.

The triple video representation consists of video frame, optical flow and motion salien-

cy. The stacked video frames $\{vf_1, vf_2, ..., vf_n\}$ provide the appearance cue. The optical flow provides direct motion information instead of the inferences between stacked video frames. A popular optical flow-based motion detection method [3] is used to extract optical flow $\{opf_1\_x, opf_2\_x, ..., opf_n\_x\}$ and $\{opf_1\_y, opf_2\_y, ..., opf_n\_y\}$ from the stacked video frames, here opf_i_x denotes the horizontal optical flow of the ith video frame and opf_i_y denotes the vertical optical flow of the ith video frame. The motion saliency captures the salient motion information and suppresses background motion noises. The phase spectrum-based motion saliency detection method [6] is used to obtain motion saliency information $\{ms_1, ms_2, ..., ms_n\}$ from the n stacked video frames, here ms_i denotes the motion saliency of the ith video frame.

Based on the above fundamentals, the triple video frame can be represented as $tvf : \{vf, opf, ms\}$, *i.e.* $\{vf_r, vf_g, vf_b, opf_x, opf_y, ms\}$. Compared with the previous video frame vf which only provides the appearance information as an input, the novel triple video frame tvf can provide richer input information, including appearance information, direct motion information and salient motion information.

2.3.3 3D convolutions on a multi-cue input

Based on the novel triple video representation, we propose to perform 3D convolutions on the stacked triple video frames, *i.e.* perform 3D convolutions on a multi-cue input instead of a single-cue input. Three different cues are adopted as an input, which includes the appearance cue, direct motion cue and salient motion cue. The computation process of the proposed novel 3D convolutions on a multi-cue input can be illustrated in Figure 2.5.

From the perspective of the temporal dimension, as a $3 \times 3 \times 3$ filter kernel is applied, total 18 colour channels are used for each computation of the proposed 3D convolution on a multi-cue input. Three stacked triple video frames are involved for each computation, and each triple video frame contains 6 channels: three colour channels, a horizontal optical flow channel, a vertical optical flow channel and a motion saliency channel. An obvious difference between the single-cue input and the multi-cue input is the number of input channels. The single-cue input of stacked video frames usually provides 3 input channels while the proposed multi-cue input usually provides 6 input channels. The added channels focus on providing motion information, including motion velocity and motion saliency.

2.3.4 M3D model

To evaluate the effectiveness of our proposed 3D convolutions on a multi-cue input, we design our multi-cue 3D convolution based on C3D model (M3D for short) [36]. The C3D model is a benchmark architecture of 3D convolution neural networks, which contains 8 convolutional layers, 5 max-pooling layers, 2 fully-connected layers and 1 softmax layer. The architecture can be denoted as (conv1, pool1, conv2, pool2, conv3, conv4, pool3,

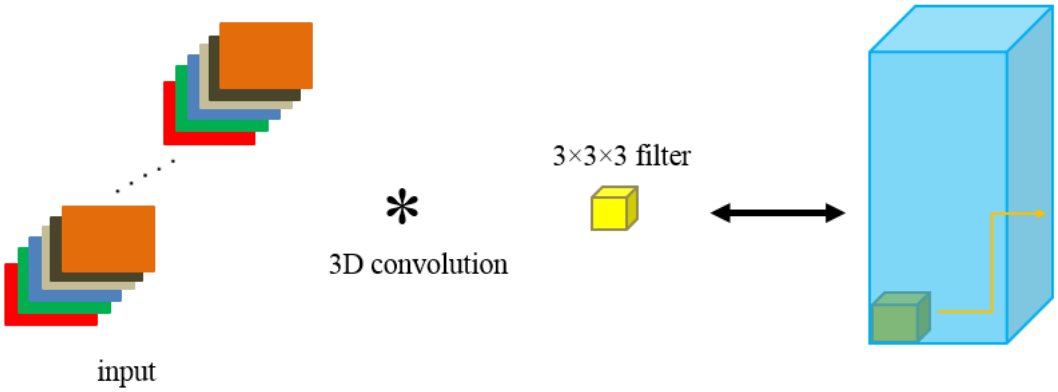

Figure 2.5: An illustration of the 6 involved channels for each computation of the proposed novel 3D convolutions on a multi-cue input along the temporal direction. The 6 channel are shown in different colours. The red rectangle, green rectangle and blue rectangle represent three different colour channels. The shallow grey rectangle and dark grey rectangle represent the corresponding horizontal optical flow channel and the vertical optical flow channel. The orange rectangle represents the corresponding motion saliency channel.

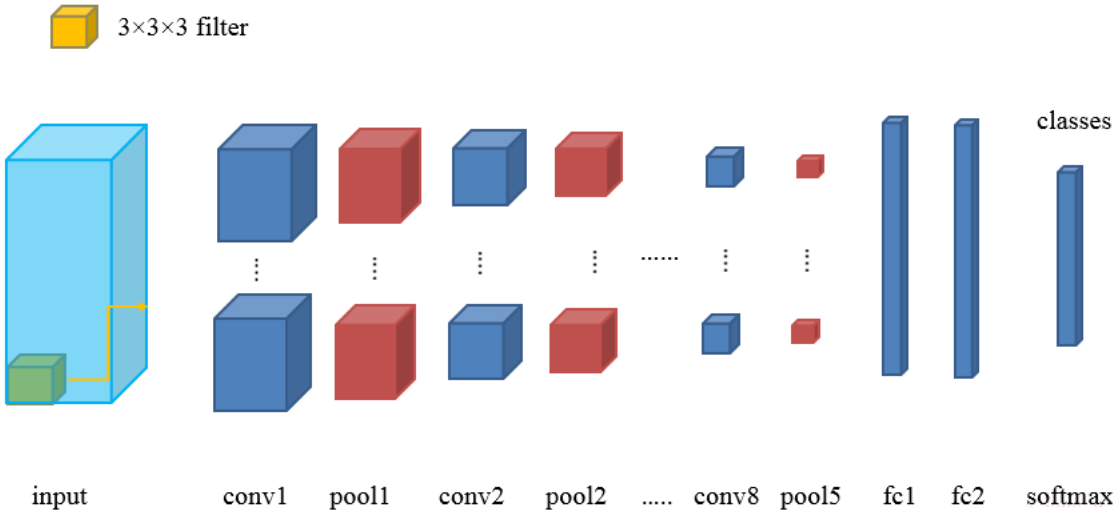

Figure 2.6: The architecture of the proposed M3D model, which can be denoted as (conv1, pool1, conv2, pool2, conv3, conv4, pool3, conv5, conv6, pool4, conv7, conv8, pool5, fc1, fc2, softmax).

conv5, conv6, pool4, conv7, conv8, pool5, fc1, fc2, and softmax). The main difference between the C3D model and our proposed M3D model is in the input part. The C3D model performs 3D convolutions on a single-cue input, *i.e.* each video clip as a sample. However, our proposed M3D model performs 3D convolutions on a multi-cue input using the proposed triple video frame input representation, which stacks video frames (including

RGB three colour channels), optical flow frames (including horizontal and vertical two directions channels) and motion saliency frames (including one channel) together as one input. The architecture of our proposed M3D model can be illustrated in Figure 2.6.

The procedure of the M3D model can be summarized in Algorithm 1 as follows:

Algorithm 1:	M3D model
Step1:	Extracting optical flow $\{opf_1\_x, opf_2\_x, ..., opf_n\_x\}$ and $\{opf_1\_y, opf_2\_y, ..., opf_n\_y\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step2:	Extracting motion saliency $\{ms_1, ms_2, ..., ms_n\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step3:	Expanding each video frame as a triple video frame representation composed of the original video frame, the corresponding optical flow and motion saliency. The triple video frame can be represented as tvf : $\{vf_r, vf_g, vf_b, opf_x, opf_y, ms\}$.
Step4:	Obtaining the corresponding stacked triple video frames$\{tvf_1, tvf_2, ..., tvf_n\}$ from the stacked video frames $\{vf_1, vf_2, ..., vf_n\}$.
Step5:	Performing 3D convolutions on the stacked triple video frames $\{tvf_1, tvf_2, ..., tvf_n\}$ and training our proposed M3D network for classification.

2.3.5 Deep R-M3D model

To improve the representation ability to obtain more representative video features. We further explore to apply our proposed multi-cue 3D convolutions on deep network architectures. 3D Residual Network (3D ResNet for short) [15, 18] is an excellent deep network, which can effectively alleviate the degradation problem [18]. The degradation problem indicates that as the network depth increasing, the training accuracy will get saturated but then degrade rapidly. Standard deep neural networks usually use multiple stacked layers to approximate the desired underlying mapping and transit information layer by layer. Formally, a building block of stacked layers in standard deep neural networks can be defined as follows:

$$H(\mathbf{x}) \approx F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) \tag{2.2}$$

where $\mathbf{x}$ denotes the input, $H(\mathbf{x})$ denotes the desired underlying mapping function, $F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\})$ denotes the actual output function of multiple stacked nonlinear layers which can approximate the desired underlying mapping function $H(\mathbf{x})$, $\mathbf{W}$ denotes the weights and $\mathbf{b}$ denotes the biases. For example, a building block of two stacked layers is illustrated in Figure 2.7.

We can find that $F = W_2 f(W_1 \mathbf{x} + b_1) + b_2$ where $f(\cdot)$ denotes the nonlinear activation function ReLU [18].

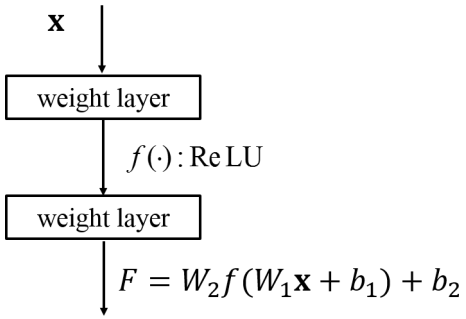

Figure 2.7: The building block of standard deep neural networks.

In contrary to standard deep neural networks, ResNets consider using multiple stacked layers to approximate a residual mapping by directly pass gradient flows from front layers to back layers, which can effectively ease the degradation problem. Formally, a residual building block can be defined as follows:

$$H(\mathbf{x}) \approx F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) + \mathbf{x} \tag{2.3}$$

where $H(\mathbf{x}) - \mathbf{x}$ denotes the residual mapping function and $F(\mathbf{x}, \{\mathbf{W}, \mathbf{b}\}) + \mathbf{x}$ is regarded as injecting a shortcut connection from the input to the output. For example, a residual building block of ResNets can be illustrated in Figure 2.8.

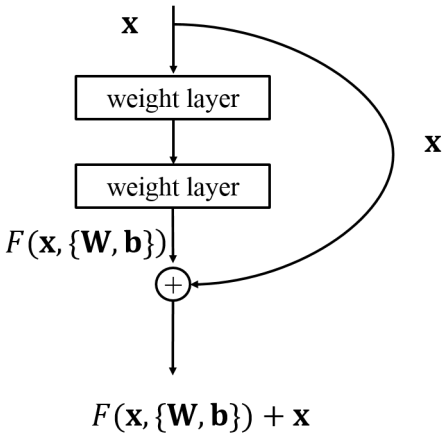

Figure 2.8: The residual building block of ResNets.

Based on 3D ResNet, we design a deep residual multi-cue 3D convolution model (R-M3D for short). Compared with the M3D model consisting of 8 layers, the number of layers of R-M3D can reach to {18, 34, 50, 101 and 152}. Similar to the M3D model based on C3D model, the main difference between R-M3D and 3D ResNet depends on whether

3D convolutions operating on a multi-cue input or a single-cue input. In addition, in contrast to 3D ResNet directly using a fully-connected layer as the output layer (*i.e.* the softmax layer), we add a new fully-connected layer between the last convolutional layer and the output layer in our proposed R-M3D, which can synthesize all feature maps of the last convolutional layer together to improve the performance of action recognition, more details can be found in Section 2.4.2.1. The procedure of R-M3D is similar to the procedure of M3D model as illustrated in Algorithm1.

2.4 Experimental analysis

In this section, we present the experimental details. First, we introduce the UCF101 and HMDB51 datasets in Section 2.4.1. Then discuss the implementation for network architecture, training process and recognition process in Section 2.4.2. After that, we mainly analysis the experimental results of the single-cue model, two-cue models and the proposed M3D model in Section 2.4.3. Finally, the experimental results of the proposed deep R-M3D model will be presented and compared with other state-of-the-art models in Section 2.4.4.

2.4.1 Datasets

We mainly evaluate our model and other models on UCF101 [35] and HMDB51 [27] datasets.

- UCF101 dataset is a temporal trimmed action dataset, which contains 13320 videos and has 101 action categories (average about 130 videos for each action category). Three train/test splits are provided for distinguishing training dataset and test dataset. There are about 9500 videos for training and about 3800 videos for testing according to each UCF101 split.
- HMDB51 dataset contains 6766 videos and 51 action categories, each action category contains at least 101 videos. These action categories can be grouped into 5 different types: general facial actions such as smile, facial actions with object manipulation such as smoke, general body movements such as climb, body movements with object interaction such as golf, body movements for human interaction such as hug. Similar to UCF101 dataset, this dataset also provides three train/test splits for distinguishing training dataset and test dataset (about 70% training dataset and 30% test dataset).

2.4.2 Implementation

The experimental environment is Ubuntu 16.04 and Python 3.5. We implement our proposed M3D model and R-M3D model based on the deep learning framework Pytorch 0.3.1. To train our proposed M3D model and R-M3D model on UCF101 and HMDB51 datasets, stochastic gradient descent (SGD) is adopted. We train our model with a batch size of 32 on 4 GPUs (Nvidia GTX 1080Ti). The learning rate is set to 0.1 and it will be divided by 10 if the validation loss saturates. The weight decay is set to 0.001. The network architecture of the proposed M3D model can be found in Section 2.3. We present the network architecture of the proposed R-M3D model in Section 2.4.2.1. Then the training process and recognition process in detail are presented in Section 2.4.2.2 and Section 2.4.2.3, respectively.

2.4.2.1 Network architecture of R-M3D

To explore a deep network, we choose a deep 3D ResNet [16] as the backbone network instead of C3D network. According to [16], 3D ResNet with 34 layers is enough and suitable for UCF101 and HMDB51 datasets. In our implementation, we adopt 3D ResNet with 34 layers and modify 3D ResNet into R-M3D, the main differences between them contains two aspects: (i) The input channels of R-M3D are 6 (a multi-cue triple video input) while the input channels of 3D ResNet are 3 (a single-cue RGB input), thus the proposed R-M3D perform 3D convolutions on a multi-cue input; (ii) A new fully-connected layer of 1024 neurons is added between the last convolutional layer and the output layer (*i.e.* the softmax layer) in our proposed R-M3D, which can synthesize all feature maps of the last convolutional layer together to improve the performance of action recognition. The number of neurons in the softmax layer is 101 or 51, which is corresponding to the number of classes on UCF101 or HMDB51 datasets, respectively. Thus, the proposed R-M3D contains 35 layers, which can be denoted as (conv1, maxpool, conv2_3, conv3_4, conv4_6, conv5_3, averagepool, fc and softmax), here x in conv2, 3, 4, 5_x denotes the multiple of the corresponding residual building block. Each frame is resized into 112 $\times$ 112 and the input size of each sample clip is 16 $\times$ 6 $\times$ 112 $\times$ 112, where 16 denotes each sample clip contains 16 frames and 6 denotes the number of channels (more detail can be found in Section 2.4.2.2). The detail of R-M3D network architecture is illustrated in Table 2.1.

2.4.2.2 Training process

To perform data augmentation, we first randomly select a video clip of 16 consecutive frames as a training sample from a raw video. Then similar to [42], the training sample is randomly cropped from 5 positions: top left corner, top right corner, bottom left corner, bottom right corner and center. We also randomly horizontally flip the training sample with 50% probability to perform data augmentation. Each training sample size is resized

Table 2.1: The network architecture of the proposed R-M3D (35 layers) in detail is illustrated. A residual building block is illustrated in brackets.

Layer name	Layer architecture	Output size
conv1	$[7 \times 7 \times 7, 64]$, kernel size: $7 \times 7 \times 7$, number of feature maps: 64, stride size: $2 \times 2 \times 2$, padding size: $3 \times 3 \times 3$	$56 \times 56 \times 16$, feature map size: 56×56, number of input frames: 16
maxpool	kernel size: $3 \times 3 \times 3$, stride size: $2 \times 2 \times 2$, padding size: $1 \times 1 \times 1$	$28 \times 28 \times 8$
conv2_x	$\begin{bmatrix} 3 \times 3 \times 3, 64 \\ 3 \times 3 \times 3, 64 \end{bmatrix} \times 3$ for all convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$28 \times 28 \times 8$
conv3_x	$\begin{bmatrix} 3 \times 3 \times 3, 128 \\ 3 \times 3 \times 3, 128 \end{bmatrix} \times 4$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$14 \times 14 \times 4$
conv4_x	$\begin{bmatrix} 3 \times 3 \times 3, 256 \\ 3 \times 3 \times 3, 256 \end{bmatrix} \times 6$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$7 \times 7 \times 2$
conv5_x	$\begin{bmatrix} 3 \times 3 \times 3, 512 \\ 3 \times 3 \times 3, 512 \end{bmatrix} \times 3$ for the first convolutional layers: stride size: $2 \times 2 \times 2$ padding size: $1 \times 1 \times 1$, for other convolutional layers: stride size: $1 \times 1 \times 1$ padding size: $1 \times 1 \times 1$	$4 \times 4 \times 1$
averagepool	kernel size: $4 \times 4 \times 1$, stride size: $1 \times 1 \times 1$ padding size: $0 \times 0 \times 0$	$1 \times 1 \times 1$
fc	512×1024	-
softmax	$1024 \times \{101 \quad or \quad 51\}$	-

to 112 × 112. Thus, for 3D convolutions performed on a single-cue input, the input is a video clip with a size of 16 × 3 × 112 × 112 composed of 16 video frames. However, for 3D convolutions performed on a multi-cue input, the input is a new reformed video clip with a size of 16 × 6 × 112 × 112 composed of 16 triple video frames. For the validation process, we uniformly split the video into three video clips to perform data augmentation, and each video clip represents the same video with the same class label.

2.4.2.3 Recognition process

After training our proposed M3D model and R-M3D model, we use them to recognize the actions in the test dataset. As action recognition is a classification problem [47], classification accuracy is adopted as the main evaluation metric. We train and test our proposed M3D model and R-M3D model on each training/test split of UCF101 dataset and HMDB51 dataset. The average classification accuracies over all three training/test splits are adopted as the final report results. To evaluate the classification accuracy of action recognition for a test video, we split the test video into non-overlapped video clips with a size of 16. Each video clip is cropped from the center position and resized into 112 × 112. We use our trained model to classify each video clip to obtain the probabilities for each class label, and average the probabilities of all the video clips corresponding to the test video for classification.

2.4.3 Analysis of experimental results of the M3D model

In this section, we first set up two sets of ablation comparison experiments, we first compare our proposed M3D model with different input modalities in Section 2.4.3.1. Then we compare the effects of different numbers of layers of CNN for M3D in Section 2.4.3.2. Finally, we analyse the computation cost of our proposed M3D model with the C3D model in Section 2.4.3.3.

2.4.3.1 Ablation experiments: different input modalities

To compare the different effects of different inputs, we select different input combinations as different comparison models. For fair comparison, all models use the C3D model as the basic model and the only difference is the input part. All the comparison models can be summarized as follows: (i) Adopting video frames as a single-cue input, *i.e.* C3D baseline model [36]. (ii) Adopting video frame and optical flow as a two-cue input, we call it T3D-I model for short. (iii) Adopting video frame and motion saliency as a two-cue input, we call it T3D-II model for short. (iv) Adopting video frame, optical flow and motion saliency as a multi-cue input, *i.e.* our proposed M3D model. The corresponding best classification

Table 2.2: Top-1 classification accuracies of different input modalities of 3D convolution models on UCF101 and HMDB51 datasets (Mean of all accuracies over 3 splits).

Input Modalities	UCF101	HMDB51
Video frame (C3D from scratch)	43.2%	16.2%
Video frame + optical flow frame (T3D-I)	48.5%	20.1%
Video frame + motion saliency frame (T3D-II)	44.8%	17.6%
All modalities (M3D)	**49.0%**	**20.8%**

accuracies of different input modalities on the UCF101 and HMDB51 datasets are reported in Table 2.2. We can get these observations as follows:

- As experimental results on the UCF101 dataset illustrated, our proposed M3D model, the T3D-I model, T3D-II model and C3D model obtain 49.0%, 48.5%, 44.8% and 43.2% classification accuracy on the UCF101 dataset, respectively. The first observation is that two-cue input-based or multi-cue input-based models (T3D-I, T3D-II and M3D) outperform the baseline C3D model because more motion related cues can be provided in the input level. The second observation we can find that the T3D-I model improves much more than the T3D-II model (5.3% vs. 1.6%). This denotes optical flow contributes much more than motion saliency for action recognition, because motion velocity can provide more detailed motion information (*i.e.* motion velocity including motion speed and motion direction) of each pixel while motion saliency mainly provides salient moving pixel information. Lastly, we find that our proposed M3D model obtains the best performance compared with all the other models. The reason is that it can integrate all the appearance cue, direct motion cue and salient motion cue as a multi-cue input.

- As experimental results on the HMDB51 dataset illustrated, we can also find that our proposed M3D model obtains the best performance of classification accuracy (20.8%), followed by the T3D-I model (20.1%), T3D-II model (17.6%) and C3D model (20.8%) just like on the UCF101 dataset. It has been verified that the multi-cue input is more effective than the single-cue input once again and the motion information provided in the input is beneficial for 3D convolutions to improve action recognition. Further, we find the direct motion cue (*i.e.* motion velocity information) is more important than the salient motion cue (*i.e.* motion saliency information) for action recognition because the T3D-I model outperforms the T3D-II model (20.1% vs. 17.6%). Our proposed M3D model performs better than the T3D-I model (20.8% vs. 20.1%) because the M3D model provides the extra motion saliency cue in the input level which can suppress background motion noises.

Table 2.3: Classification accuracy of different numbers of layers of CNN with different input modalities on UCF101 and HMDB51 datasets.

Depth of layers of CNN	UCF101	HMDB51
Input modalities: video frame		
2-layer	39.3%	14.3%
4-layer	41.5%	15.0%
6-layer	42.6%	15.4%
Input modalities: video frame + optical flow frame		
2-layer	44.5%	16.2%
4-layer	45.3%	18.5%
6-layer	47.8%	19.2%
Input modalities: video frame + motion saliency frame		
2-layer	41.7%	14.8%
4-layer	43.5%	16.4%
6-layer	44.2%	17.2%
Input modalities: video frame + optical flow frame +motion saliency frame		
2-layer	45.5%	16.1%
4-layer	47.6%	18.6%
6-layer	48.4%	19.4%

2.4.3.2 Ablation experiments: different numbers of layers of CNN for M3D

To compare the effectiveness of different numbers of layers, we compare the classification accuracy of different numbers of layers of CNN with different input modalities on UCF101 and HMDB51 in Table 2.3. According to Table 2.3, for the input modalities of video frame, *i.e.* the single-cue input, we can find that the accuracy will improve as the depth of layers increases on both UCF101 and HMDB51 datasets. For the two-cue input (the input modalities of combining video frame with optical flow frame, or the input modalities of combining video frame with motion saliency frame) or the multi-cue input (the input modalities of combining video frame, optical flow frame and motion saliency frame), the same phenomenon appears, *i.e.* the accuracy will improve as the depth of layers increases. This verifies that the classification accuracy of our proposed M3D can be improved by increase the depth of layers. Thus, we apply our proposed M3D in a deeper CNN model, *i.e.* ResNet, and compare it with the state-of-the-art in Section 2.4.4.

2.4.3.3 Computation cost

We compare the computation costs of different models: C3D, T3D-I, T3D-II and our proposed M3D. Concretely, we train the models (C3D, T3D-I, T3D-II and M3D) for 50 epochs on the UCF101 dataset and use the average training time of an epoch as the

Table 2.4: The computation costs of different models (C3D, T3D-I, T3D-II and M3D) on the UCF101 dataset.

Models with different input modalities	Computation cost (s)
C3D (RGB)	1914
T3D-I (RGB + Optical Flow)	2044
3D-II (RGB + Motion Saliency)	2011
M3D (All Modalities)	**2054**

computation cost to be recorded. The computation costs of different models (C3D, T3D-I, T3D-II and M3D) are reported in Table 2.4. We can find that the time cost of the compared models is close, almost at the same order of magnitude. We speculate the reason is that the main difference among different models is the number of the input channels and others are the same. Thus the computation cost should be close for different compared models, which demonstrates our proposed M3D can improve the performance of action recognition with the almost same training computation cost compared with C3D. Note that for our proposed M3D, we need extra computation cost for computing the optical flow and motion saliency from the stacked RGB frames in the preprocessing stage.

In addition, we also demonstrate the training convergence of the loss function of the single-cue 3D model (C3D) and the multi-cue 3D model (our proposed M3D) on the UCF101 dataset in Figure 2.9. This shows that our proposed M3D model converges faster than the C3D model, which indicates the multi-cue input is more helpful for speeding up model training compared with the single-cue input.

2.4.4 Comparing the deep R-M3D model with the state-of-the-art

Based on the above experimental discussion, it has been verified the fact that 3D convolutions performed on a multi-cue input outperform 3D convolutions performed on a single-cue input. However, the classification accuracies are not satisfactory compared with current state-of-the-art models. One reason is that C3D model only contains about 8 layers, which is not deep enough. Another reason is that the scale of UCF101 and HMDB51 datasets only is about 10,000, which are not enough for training a deep neural network. Kinetics dataset [25] is a huge trimmed action dataset, which contains about 300,000 videos and 400 action categories. To explore a deep network on a larger dataset, we fine-tune the proposed R-M3D based on the pretrained Kinetics 3D ResNet on UCF101 and HMDB51 datasets.

We compare our proposed R-M3D ResNet model with other state-of-the-art models including one traditional hand-crafted model (*i.e.* improved dense trajectories (iDT) [41]) and six deep learning models (*i.e.* two-stream convolutional networks [34], two-stream

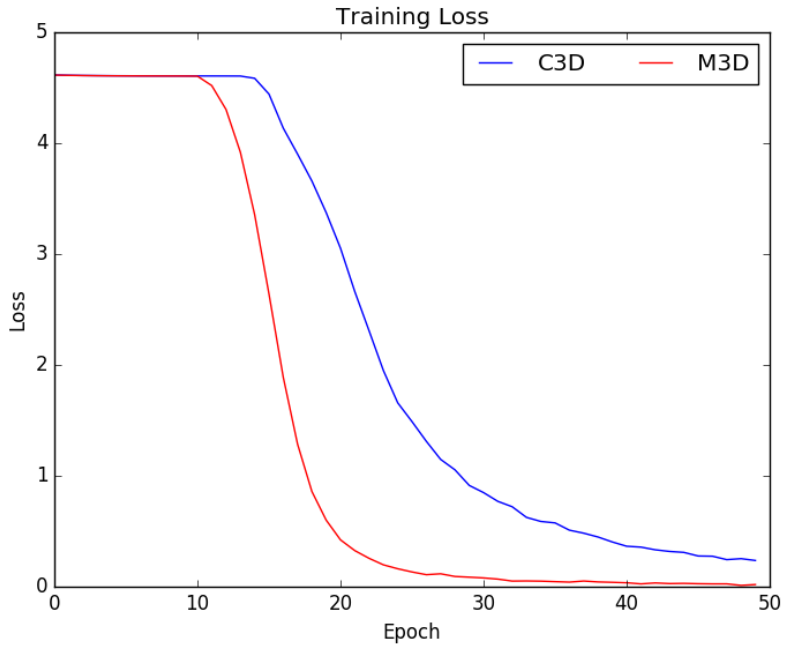

Figure 2.9: The training loss of C3D and M3D on the UCF101 dataset, the number of epochs is 50.

with LSTM [44], long-term recurrent convolutional networks (LRCN) [9], long-term temporal convolutions networks (LTCN) [39], Spatiotemporal Residual Networks (ST-ResNet) [10], Pseudo-3D Residual Networks (P3D ResNet) [33], 3D convolutional networks (C3D) [36])and 3D ResNet [16]). Note that 3D ResNet in [16] also is fine-tuned on UCF101 and HMDB51 datasets by using the Kinectics pretrained 3D ResNet. Experimental results on UCF101 and HMDB51 datasets are summarized in Table 2.5.

According to Table 2.5, we can find that the proposed fine-tuned R-M3D model outperforms all other compared state-of-the-art models (including iDT+FV, Two-stream networks, Two-stream + LSTM, LRCN, LTCN, C3D (3 nets), P3D ResNet and 3D ResNet) except ST-ResNet on UCF101 and HMDB51 datasets. Especially compared to 3D ResNet, the proposed R-M3D improves 3.9% and 4.4% on UCF101 and HMDB51 datasets, respectively. This demonstrates that 3D convolutions performed on a multi-cue input can improve the action recognition compared with 3D convolutions performed on a single-cue input. We can also find that fine-tuned R-M3D has a significant improvement compared with the M3D model on UCF101 and HMDB51 datasets. This indicates that both a deep network and a larger dataset are important and critical for 3D CNNs to improve action recognition. In addition, we find that the accuracy of our proposed R-M3D is lower 0.2% and 1.0% than ST-ResNet on both UCF101 and HMDB51 datasets, respectively. This reason may be attributed to ST-ResNet can make fully use of the complementary information between the two streams of the two-stream architecture.

Table 2.5: Top-1 classification accuracies of the proposed R-M3D model compared with the state-of-the-art models on UCF101 and HMDB51 datasets (Mean of all accuracies over 3 splits).

Models	UCF101	HMDB51
iDT+FV [41]	85.9%	57.2%
Two-stream networks [34]	86.9%	58.0%
Two-stream + LSTM [44]	88.6%	–
LRCN [9]	82.9%	–
LTCN [39]	91.7%	64.8%
C3D (3 nets) [36]	85.2%	–
P3D ResNet [33]	88.6%	–
ST-ResNet [10]	93.4%	66.4%
3D ResNet (fine-tuned) [16]	89.3%	61.0%
R-M3D (fine-tuned)	**93.2%**	**65.4%**

2.5 Summary

In this chapter, we propose a novel M3D model for action recognition. The M3D model directly performs 3D convolutions on a multi-cue input, *i.e.* stacked triple video frames including appearance information, direct motion information and salient motion information. Compared with the existing 3D CNN-based action recognition methods, the proposed novel triple video representation can integrate the appearance cue, direct motion cue and salient motion cue as input for 3D convolutions. Further, the salient motion cue is robust to background motion noises such as slight leaf jittering and water rippling, which has not been applied in action recognition before. We also develop R-M3D based on the deep 3D ResNet for action recognition. Experimental results verified the effectiveness of our proposed M3D model, and the proposed R-M3D model achieves competitive performance compared with the state-of-the-art.

References

[1] J Arunnehru, G Chamundeeswari, and S Prasanna Bharathi. Human action recognition using 3D convolutional neural networks with 3D motion cuboids in surveillance videos. *Procedia Computer Science*, 133:471–477, 2018.

[2] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

[3] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy

optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36. Springer, 2004.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Chenglizhao Chen, Shuai Li, Yongguang Wang, Hong Qin, and Aimin Hao. Video saliency detection via spatial-temporal fusion and low-rank coherency diffusion. *IEEE Transactions on Image Processing*, 26(7):3156–3170, 2017.

[6] Zhe Chen, Xin Wang, Zhen Sun, and Zhijian Wang. Motion saliency detection using a temporal fourier transform. *Optics & Laser Technology*, 80:1–15, 2016.

[7] Runmin Cong, Jianjun Lei, Huazhu Fu, Ming-Ming Cheng, Weisi Lin, and Qingming Huang. Review of visual saliency detection with comprehensive information. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[8] Xinyi Cui, Qingshan Liu, Shaoting Zhang, Fei Yang, and Dimitris N Metaxas. Temporal spectral residual for fast salient motion detection. *Neurocomputing*, 86:24–32, 2012.

[9] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016.

[11] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768–4777, 2017.

[12] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[13] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[14] Chenlei Guo, Qi Ma, and Liming Zhang. Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3154–3160, 2017.

[16] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[17] Jonathan Harel, Christof Koch, and Pietro Perona. Graph-based visual saliency. In *Advances in Neural Information Processing Systems*, pages 545–552, 2007.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for

image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[19] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[20] Xiaodi Hou and Liqing Zhang. Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. Ieee, 2007.

[21] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[22] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[23] Yuzhu Ji, Haijun Zhang, and QM Jonathan Wu. Salient object detection via multi-scale attention CNN. *Neurocomputing*, 322:130–140, 2018.

[24] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[25] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[27] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563, 2011.

[28] Zhenbing Liu, Zeya Li, Ruili Wang, Ming Zong, and Wanting Ji. Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. *Neural Computing and Applications*, 32(18):14593–14602, 2020.

[29] Zhenbing Liu, Zeya Li, Ming Zong, Wanting Ji, Ruili Wang, and Yan Tian. Spatiotemporal saliency based multi-stream networks for action recognition. In *Asian Conference on Pattern Recognition*, pages 74–84. Springer, 2019.

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[31] Eduardo M Pereira, Lucian Ciobanu, and Jaime S Cardoso. Cross-layer classification framework for automatic social behavioural analysis in surveillance scenario. *Neural Computing and Applications*, 28(9):2425–2444, 2017.

[32] Javier Sánchez Pérez, Enric Meinhardt-Llopis, and Gabriele Facciolo. Tv-l1 optical flow estimation. *Image Processing On Line*, pages 137–150, 2013.

[33] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *Proceedings of the IEEE International Conference*

on Computer Vision, pages 5533–5541, 2017.

[34] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[35] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[36] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[37] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

[38] Zhigang Tu, Hongyan Li, Dejun Zhang, Justin Dauwels, Baoxin Li, and Junsong Yuan. Action-stage emphasized spatio-temporal vlad for video action recognition. *IEEE Transactions on Image Processing*, 2019.

[39] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018.

[40] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, 2011.

[41] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[42] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[43] Yawen Xue, Xiaojie Guo, and Xiaochun Cao. Motion saliency detection using low-rank and sparse decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1485–1488, 2012.

[44] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[45] Shaoning Zeng, Jianping Gou, and Xiong Yang. Improving sparsity of coefficients for robust sparse and collaborative representation-based image classification. *Neural Computing and Applications*, 30(10):2965–2978, 2018.

[46] Haijun Zhang, Yuzhu Ji, Wang Huang, and Linlin Liu. Sitcom-star-based clothing retrieval for video advertising: a deep learning framework. *Neural computing and applications*, 31(11):7361–7380, 2019.

[47] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. Efficient

knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1774–1785, 2018.

[48] Yizhou Zhou, Xiaoyan Sun, Zheng-Jun Zha, and Wenjun Zeng. Mict: Mixed 3D/2D convolutional tube for human action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 449–458, 2018.

Chapter 3

Motion saliency based multi-stream multiplier ResNets for action recognition

In this chapter, we propose a Motion Saliency based multi-stream Multiplier ResNets (MSM-ResNets) for action recognition. The proposed MSM-ResNets model consists of three interactive streams: the appearance stream, motion stream and motion saliency stream. Similar to conventional two-stream CNNs models, the appearance stream and motion stream are responsible for capturing the appearance information and motion information, respectively, while the motion saliency stream is responsible for capturing the salient motion information. In particular, to effectively utilize the spatiotemporal interactive information between different streams, the proposed MSM-ResNets model establishes interactive connections between different streams instead of fusing three streams at the final output layer. Two kinds of different multiplicative connections are injected, the first one is to inject multiplicative connections from the motion stream to the appearance stream, while the second one is to inject multiplicative connections from the motion saliency stream to the motion stream. Experimental results verify the effectiveness of the proposed MSM-ResNets on two standard action recognition datasets: UCF101 and HMDB51.

The organization of this chapter is as follows. Section 3.1 introduces the motivation of the proposed approach. Section 3.2 introduces a motion saliency detection method and related work of two-stream CNNs based methods for action recognition. Then a novel motion saliency based multi-stream multiplier ResNets is proposed in Section 3.3. The architecture implementation of the proposed MSM-ResNets is presented in Section 3.4 and experimental results are analysed in Section 3.5. Finally, Section 3.6 concludes this chapter.

3.1 Introduction

Action recognition is a highly active research field in computer vision, which aims to recognize the action category from the videos [7, 9, 36]. However, the accuracy of current action recognition methods is still not on par with the human. The reason is that action recognition faces lots of challenges such as moving background clutters, camera viewpoint changes and illumination variation[3, 42–44].

Convolutional Neural Networks (CNNs)[33, 34, 38] have been widely applied to many image-based visual tasks such as image classification, object recognition and object segmentation [11, 19, 28, 30, 32]. For example, many famous CNNs architectures, such as AlexNet [20], VGGNet [30], Inception [32] and ResNets [16], have obtained significant successes in image classification field [27, 38, 45]. To utilize the successful CNNs architectures for action recognition, domain adaption is considered, *i.e.* transferring the knowledge of successful CNNs architectures in image-based visual tasks from the image domain to the video domain [25, 26, 47, 49]. Thus, various CNN-based models are proposed for action recognition [6, 15, 18], which can be roughly classified into three categories: two-stream CNN based models, 3D CNN based models and Recurrent Neural Network (RNN) based models [35, 37]. Among these models, two-stream CNN based models have achieved a superior performance in action recognition, which consists an appearance stream with RGB video frames as input and a motion stream with optical flow frames as input. Conventional two-stream CNN based models usually fuse the appearance information and the motion information at the final output layer in a late fusion manner, *i.e.* the spatiotemporal information is fused at the output layer. The appearance stream and the motion stream are separate and there are no interactions between them until the output layer. Thus, the conventional separate two-stream CNNs architecture cannot fuse the appearance cue and the motion cue over time [12–14, 22]. However, The interactions between the appearance stream and the motion stream can provide finer spatiotemporal fusion information, which can improve the accuracy of action recognition.

Aiming to establish interactive connections between the appearance stream and the motion stream instead of fusing them at the final output layer, Feichtenhofer *et al.* [8] proposed spatiotemporal multiplier networks based on two-stream CNNs architecture for action recognition, which can inject connections from the motion stream to the appearance stream by multiplicative interactions. They also have proven that bidirectional connections between the appearance stream and the motion stream can lead an inferior accuracy of action recognition. However, current most of two-stream CNNs based models treat each moving pixel equally and ignore paying attention on the salient actions, *i.e.* the salient moving pixels.

Inspired by the above discussions, the motivations of this chapter contain two folds. The first one is to capture the salient motion information, and the second one is to effectively utilize the spatiotemporal fusion information between different streams. In this chapter, we propose a Motion Saliency based multi-stream Multiplier ResNets (named MSM-ResNets for short) for action recognition. Compared with [8], our proposed MSM-ResNets model

consists of three interactive streams: the appearance stream, motion stream and motion saliency stream. The appearance stream is responsible for capturing the appearance information with RGB video frames as input. The motion stream is responsible for capturing the motion information with optical flow frames as input. The motion saliency stream is responsible for capturing the salient motion information with motion saliency frames as input. In particular, to utilize the complementary information between different streams over time instead of fusing the three streams in a late fusion manner to improve the accuracy of action recognition, the proposed MSM-ResNets model establishes multiplicative connections between different streams. Two kinds of different multiplicative connections are injected, the first one is to inject multiplicative connections to transmit the motion cue from the motion stream to the appearance stream, and the second one is to inject multiplicative connections to transmit the motion saliency cue from the motion saliency stream to the motion stream. Compared with [8], our proposed MSM-ResNets proposes a novel motion saliency stream for capturing the motion saliency information and injects multiplicative connections from the motion saliency stream to the motion stream, which can improve the robustness to the motion noise in the background (*e.g.*, slight leaf jittering or slight water rippling) [4, 24].

To sum up, the contributions of this chapter can be summarized as follows:

- A motion saliency stream is proposed for capturing the salient motion information.
- The proposed MSM-ResNets model establishes multiplicative connections between different streams (*i.e.* the appearance stream, the motion stream, and the motion saliency stream), which can capture the complementary information between different streams over time.
- The proposed MSM-ResNets model establishes multiplicative connections from the motion saliency stream to the motion stream to transmit the motion saliency cue from the motion saliency stream to the motion stream.

3.2 Background and related work

In this section, since our proposed MSM-ResNets is based on two-stream CNN architecture, we first introduce the principle of a temporal Fourier transform based motion saliency detection method, which is used for extracting motion saliency frames from consecutive RGB video frames in Section 3.2.1. Then we introduce related work of two-stream CNNs based methods for action recognition in Section 3.2.2.

3.2.1 Motion saliency detection

To capture the salient actions, *i.e.* the salient moving pixels, in the videos, a temporal Fourier transform based Motion Saliency detection method (named FMS for short) is introduced [4]. The core idea of the FMS method is that converting video frames from time domain to frequency domain by Fourier transform, and then identify the salient moving regions according the corresponding phase spectrum. The procedure of this method can be briefly summarized in Algorithm 1.

Algorithm1:	FMS
Input:	An action video
Output:	Motion saliency maps
Step 1:	Extracting consecutive RGB video frames from a given action video;
Step 2:	Extracting pixels in the same position (i, j) of consecutive RGB video frames and stacking them to form corresponding temporal sequences t_{ij};
Step 3:	Converting these temporal sequences t_{ij} from time domain to frequency domain by Fourier transform: $f_{ij} = \zeta(t_{ij})$, here ζ denotes Fourier transform;
Step 4:	Obtaining the corresponding phase spectrum of these temporal sequences $p_{ij} = \phi(f_{ij})$, here ϕ denotes the function of computing phase spectrum;
Step 5:	Converting these temporal sequences p_{ij} from frequency domain to time domain by inverse Fourier transform: $t'_{ij} = \zeta^{-1}(p_{ij})$, here ζ^{-1} denotes inverse Fourier transform;
Step 6:	Obtaining the salient moving pixels according to the result t'_{ij} and form the motion saliency maps;

The results of motion saliency maps obtained by the FMS motion saliency method from multiple consecutive video frames can be illustrated in Figure 3.1 as follows:

3.2.2 Two-stream based CNNs methods for action recognition

Two-stream CNNs based methods have been widely applied in action recognition field. Simonyan *et al.* [29] first proposed two-stream CNNs architecture for action recognition, which consists of the spatial stream and temporal stream. The spatial stream can extract the appearance features and the temporal stream can extract the motion features. Finally, the results of the spatial stream and the temporal stream are averaged. Tu *et al.*

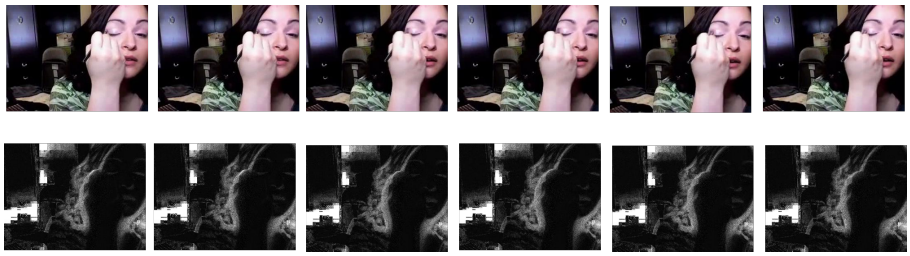

Figure 3.1: The top illustrates multiple consecutive video frames. The bottom illustrates the motion saliency map frames obtained by the FMS motion saliency method from multiple consecutive video frames.

[40] proposed a human-related region based multi-stream convolutional neural network for action recognition, which captures the human-related motion regions as an input for the additional human-related stream based on the conventional two-stream network. Liu *et al.* [23] proposed spatiotemporal saliency based multi-stream networks for action recognition, which adds the spatiotemporal saliency maps as an input based on conventional two-stream CNNs. Based on [23], Liu *et al.* [22] further proposed to combine the spatiotemporal saliency based multi-stream networks with attention-aware LSTM together for action recognition. However, there are not any interactions between the spatial stream and the temporal stream, which will lead to impair the accuracy of action recognition.

To establish interactions between the spatial stream and the motion stream for improving the accuracy of action recognition, Feichtenhofer *et al.* [10] proposed to register the spatial cue with the temporal cue by spatial fusion, which uses a special convolutional layer to fuse the spatial feature maps and the temporal feature maps at the same position. In [8], Feichtenhofer *et al.* proposed to inject residual connections from the appearance stream to the motion stream to establish the spatiotemporal interactions between the two streams. Tian *et al.* [36] proposed a multi-cue combination convolutional neural network for action recognition, which consists of three streams and uses three different features, *i.e.* the appearance features, the motion features, and the acoustic features, as input. Besides, dense connections are introduced between different streams.

In addition, Carreira *et al.* [2] proposed to expand the filters and pooling kernels into 3D based on the two-stream architecture, which can capture seamless spatiotemporal features from the videos. Varol *et al.* [41] proposed to use long-term temporal convolutions to capture the spatiotemporal features in a longer time period, which can improve the accuracy of action recognition. Dai *et al.* [5] proposed to apply attention mechanism on the outputs of different feature maps based on the two-stream model and combine the two-stream architecture with LSTM for capturing long-term cues in the video sequences.

However, current most of two-stream CNNs based models treat each moving pixel equally and ignore paying attention on the salient actions, *i.e.* the salient moving pixels, which will lead to a low accuracy when encountering various the motion noise in the background (*e.g.*, slight leaf jittering or slight water rippling) [4, 24].

3.3 Our proposed MSM-ResNets

In this section, we first introduce the baseline architecture of two-stream residual networks in Section 3.3.1. Based on the two-stream residual networks, then we will introduce the architecture of our proposed MSM-ResNets for action recognition in Section 3.3.2. Lastly, we introduce the process of multiplicative interactions of the proposed MSM-ResNets in Section 3.3.3.

3.3.1 Architecture of two-stream residual networks

Our proposed MSM-ResNets architecture is based on the two-stream residual networks, which consists of two separate CNN streams: the spatial stream is responsible for capturing the appearance features with RGB video frames as input; the temporal stream is responsible for capturing the motion features with L consecutive optical flow frames (usually $L = 20$ including 10 vertical optical flow frames and 10 horizontal optical flow frames) as input. Each stream performs its own action recognition classification task and the output of each stream is averaged on the softmax layer in a late fusion manner.

For each stream of the two-stream residual networks, ResNets [16] is adopted as the backbone network. ResNets can effectively address the degradation problem by adding skip connections to propagate information directly across multiple hidden layers. The residual block can be defined as follows:

$$\mathbf{x}_{l+1} = f(\mathbf{x}_l + \varphi(\mathbf{x}_l, W_l)) \tag{3.1}$$

where the residual block can be regarded as a layer, $\mathbf{x}_l$ and $\mathbf{x}_{l+1}$ denote the input and output of the lth layer, denotes the convolutional filter weights, $\varphi(\cdot)$ denotes a nonlinear mapping function, $f(\cdot)$ usually denotes the nonlinear function ReLU [17].

3.3.2 Architecture of MSM-ResNets

Based on two-stream residual networks, our proposed MSM-ResNets consists of three interactive streams: the appearance stream, motion stream and motion saliency stream. The appearance stream is responsible for capturing the appearance information with RGB video frames as input. The motion stream is responsible for capturing the motion information with optical flow frames as input. The motion saliency stream is responsible for capturing the salient motion information with motion saliency frames as input, which can also suppress the motion noise in the background (*e.g.*, slight leaf jittering or slight water rippling) [4, 24]. The motion saliency map frames can be obtained by the FMS motion saliency detection method [4] from multiple consecutive video frames. For each stream, ResNets is adopted as the backbone network.

In particular, to utilize the complementary information between different streams over time instead of fusing three streams on the final output layer, the proposed MSM-ResNets model establishes interactive connections between different streams. Two kinds of different multiplicative connections are injected, the first one is to inject multiplicative connections to transmit the motion cue from the motion stream to the appearance stream, and the second one is to inject multiplicative connections to transmit the motion saliency cue from the motion saliency stream to the motion stream. The structure of the proposed MSM-ResNets model with multiplicative connections from the motion stream to the appearance stream and from the motion saliency stream to the motion stream is illustrated in Figure 3.2.

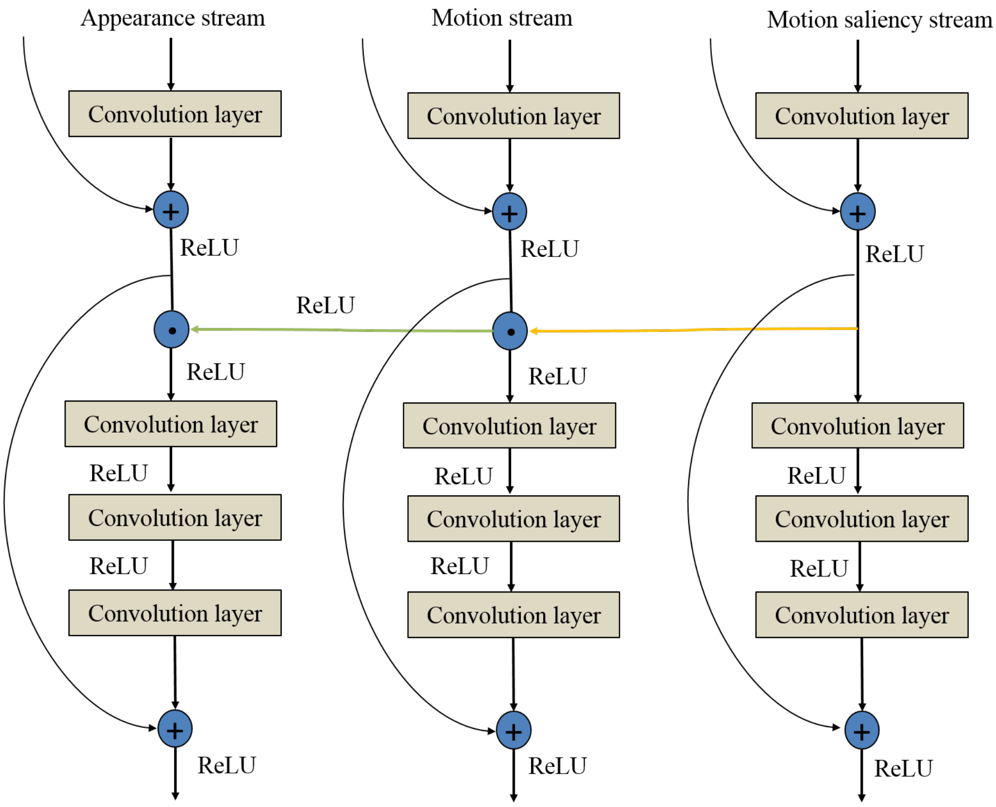

Figure 3.2: The structure of the proposed MSM-ResNets model with multiplicative connections from the motion stream to the appearance stream (green single arrow) and from the motion saliency stream to the motion stream (yellow single arrow).

3.3.3 Process of multiplicative interactions: Forward propagation and backpropagation

The process of multiplicative interactions among the appearance stream, the motion stream and the motion saliency stream in a residual block is illustrated in Figure 3.3.

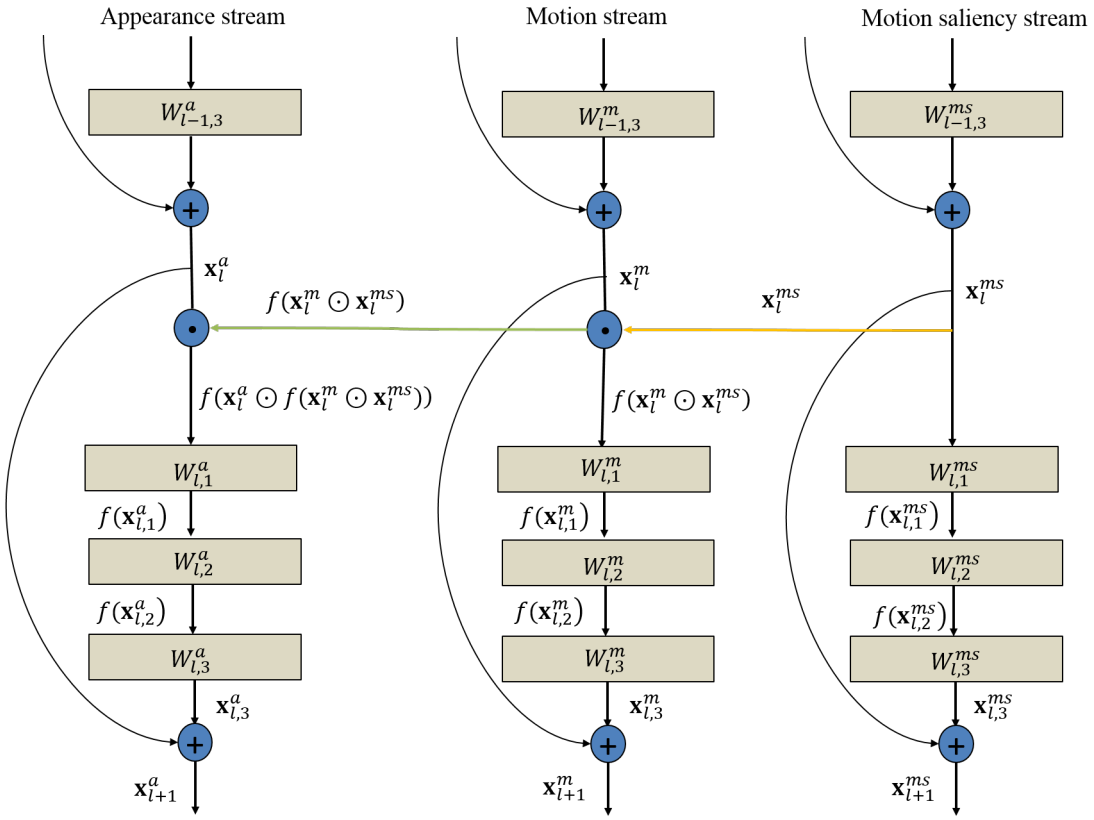

Figure 3.3: The process of multiplicative interactions in the residual building block of the proposed MSM-ResNets. The green single arrow denotes the multiplicative connections from the motion stream to the appearance stream, the yellow single arrow denotes the multiplicative connections from the motion saliency stream to the motion stream.

For the forward propagation of the residual building block of the proposed MSM-ResNets, given the lth layer input $\mathbf{x}^a_l$ of the appearance stream, the lth layer input $\mathbf{x}^m_l$ of the motion stream and the lth layer input $\mathbf{x}^{ms}_l$ of the motion saliency stream, we can obtain the corresponding forward outputs of the lth layer of the appearance stream, the lth layer of the motion stream and the lth layer of the motion saliency stream:

$$\mathbf{x}^a_{l+1} = f(\mathbf{x}^a_l + \varphi(f(\mathbf{x}^a_l \odot f(\mathbf{x}^m_l \odot \mathbf{x}^{ms}_l)), W^a_l)) \tag{3.2}$$

$$\mathbf{x}^m_{l+1} = f(\mathbf{x}^m_l + \varphi(f(\mathbf{x}^m_l \odot \mathbf{x}^{ms}_l), W^m_l)) \tag{3.3}$$

$$\mathbf{x}^{ms}_{l+1} = f(\mathbf{x}^{ms}_l + \varphi(f(\mathbf{x}^{ms}_l, W^{ms}_l)) \tag{3.4}$$

where $\odot$ denotes the elementwise multiplication, $\mathbf{x}^a_{l+1}$ denotes the output of the lth layer of the appearance stream (*i.e.* the input of the $(l+1)th$ layer of the appearance stream), $\mathbf{x}^m_{l+1}$

denotes the output of the lth layer of the motion stream (*i.e.* the input of the $(l+1)th$ layer of the motion stream) and $\mathbf{x}_{l+1}^{ms}$ denotes the output of the lth layer of the motion saliency stream (*i.e.* the input of the $(l+1)th$ layer of the motion saliency stream).

For the forward propagation, we can find that these following observations according to Equations (3.2)-(3.4): the output of the motion saliency stream is modulated by the motion saliency signal $\mathbf{x}_l^{ms}$; the output of the motion stream is modulated by the motion signal $\mathbf{x}_l^m$ and the motion saliency signal $\mathbf{x}_l^{ms}$; the output of the appearance stream is modulated by the appearance signal $\mathbf{x}_l^a$, the motion signal $\mathbf{x}_l^m$ and the motion saliency signal $\mathbf{x}_l^{ms}$.

For the backpropagation of the residual building block of the proposed MSM-ResNets, the loss function denoted as $loss$, according to the chain rule of derivation and the backpropagation [8, 20], we can derive the corresponding gradient of the appearance stream, the corresponding gradient of the motion stream, and the corresponding gradient of the motion saliency stream.

In detail, the corresponding gradient of the appearance stream in a residual block can be derived as:

$$\begin{aligned}
\frac{\partial loss}{\partial \mathbf{x}_l^a} &= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \mathbf{x}_l^a} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial f(\mathbf{x}_l^a + \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a))}{\partial \mathbf{x}_l^a} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a} f^{'}(\cdot)\frac{\partial(\mathbf{x}_l^a + \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a))}{\varphi \mathbf{x}_l^a} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a} f^{'}(\cdot)(1 + \frac{\varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a))}{\varphi \mathbf{x}_l^a})
\end{aligned} \tag{3.5}$$

where $f^{'}(\cdot)$ denotes the derivative of the nonlinear function $f(\cdot)$.

The corresponding gradient of the motion stream in a residual block can be derived as:

$$\begin{aligned}
\frac{\partial loss}{\partial \mathbf{x}_l^m} &= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^m}\frac{\partial \mathbf{x}_{l+1}^m}{\partial \mathbf{x}_l^m} \\
&+ \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}\frac{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}{\partial f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^m}\frac{\partial f(\mathbf{x}_l^m + \varphi(f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^m))}{\partial \mathbf{x}_l^m} \\
&+ \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}\frac{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}{\partial f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^m} f'(\cdot)\frac{\partial (\mathbf{x}_l^m + \varphi(f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^m))}{\partial \mathbf{x}_l^m} \\
&+ \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}\frac{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}{\partial f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})} \\
&= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^m} f'(\cdot)(1 + \frac{\partial \varphi(f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^m)}{\partial \mathbf{x}_l^m}) \\
&+ \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}\frac{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}{\partial f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})}
\end{aligned} \tag{3.6}$$

where the first item in the right of Equation (3.6) denotes the gradient from the motion stream and the second item in the right of Equation (3.6) denotes the gradient from the appearance stream.

The corresponding gradient of the motion saliency stream in a residual block can be derived as:

$$\begin{aligned}
\frac{\partial loss}{\partial \mathbf{x}_l^{ms}} &= \frac{\partial loss}{\partial \mathbf{x}_{l+1}^{ms}}\frac{\partial \mathbf{x}_{l+1}^{ms}}{\partial \mathbf{x}_l^{ms}} + \frac{\partial loss}{\partial \mathbf{x}_{l+1}^m}\frac{\partial \mathbf{x}_{l+1}^m}{\partial \varphi(f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^m)}\frac{\partial \varphi(f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^m)}{\partial \mathbf{x}_l^{ms}} \\
&+ \frac{\partial loss}{\partial \mathbf{x}_{l+1}^a}\frac{\partial \mathbf{x}_{l+1}^a}{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}\frac{\partial \varphi(f(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})), W_l^a)}{\partial f(\mathbf{x}_l^m \odot \mathbf{x}_l^{ms})}
\end{aligned} \tag{3.7}$$

where the first item in the right of Equation (3.7) denotes the gradient from the motion saliency stream, the second item in the right of Equation (3.7) denotes the gradient from the motion stream and the third item in the right of Equation (3.7) denotes the gradient from the appearance stream.

For the backpropagation, we can find that these following opposite observations compared with the forward propagation according to Equations (3.5) - (3.7): the gradient flowing of the appearance stream in a residual block is modulated by the appearance signal $\mathbf{x}_l^a$; the gradient flowing of the motion stream in a residual block is modulated by the motion saliency signal $\mathbf{x}_l^{ms}$ except the motion signal $\mathbf{x}_l^m$; the gradient flowing of the motion stream in a residual block is modulated by the appearance signal $\mathbf{x}_l^a$ and the motion signal $\mathbf{x}_l^m$ except the motion saliency signal $\mathbf{x}_l^{ms}$.

Table 3.1: The 34-layer residual network architecture used in our proposed MSM-ResNets. $\odot^{m\to a}$ denotes the injected multiplicative interaction from the motion stream to the appearance stream. $\odot^{ms\to m}$ denotes the injected multiplicative interaction from the motion saliency stream to the motion stream. The convolution operation on each convolutional layer can be described as $[W \times H, C]$. W denotes the width of the convolution kernel, H denotes the height of the convolution kernel and C denotes the number of feature maps.

Conv1	Pool1	Conv2_x	Conv3_x
$[7 \times 7, 64]$ stride size: 2	kernel size: 3×3 max pool stride size: 2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix}$ $\odot^{m\to a}, \odot^{ms\to ms}$ $\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix}$ $\odot^{m\to a}, \odot^{ms\to ms}$ $\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 3$
Conv4_x	Conv5_x	Pool5	FC
$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix}$ $\odot^{m\to a}, \odot^{ms\to ms}$ $\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix}$ $\odot^{m\to a}, \odot^{ms\to ms}$ $\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	7×7 Average pool	Softmax

3.4 Architecture implementation

In the architecture implementation of the proposed MSM-ResNets, ResNets is adopted as the backbone network for our proposed MSM-ResNets. In detail, we adopt 34-layer ResNets as the backbone network, which is suitable for the scale of the UCF101 [31] and HMDB51 [21] datasets. To utilize the successful CNNs in image classification for action recognition, the pretrained ResNets on ImageNet dataset [19] is adopted to initialize the backbone network ResNets of our proposed MSM-ResNets. Table 3.1 demonstrates the residual block architecture used in our proposed MSM-ResNets.

We illustrate the framework of the proposed MSM-ResNets in Figure 3.4. The proposed MSM-ResNets consists of three interactive streams: an RGB video frame is fed into the appearance stream, optical flow frames are fed into the motion stream and a motion saliency frame is fed into the motion saliency stream. The outputs of three streams are average on the softmax layer and the averaged result is adopted as the final prediction of the proposed MSM-ResNets.

3.5 Experiments

In this section, we present the experimental details. First, we introduce experimental preliminary in Section 3.5.1. Then we conduct the ablation contrast experiments to verify

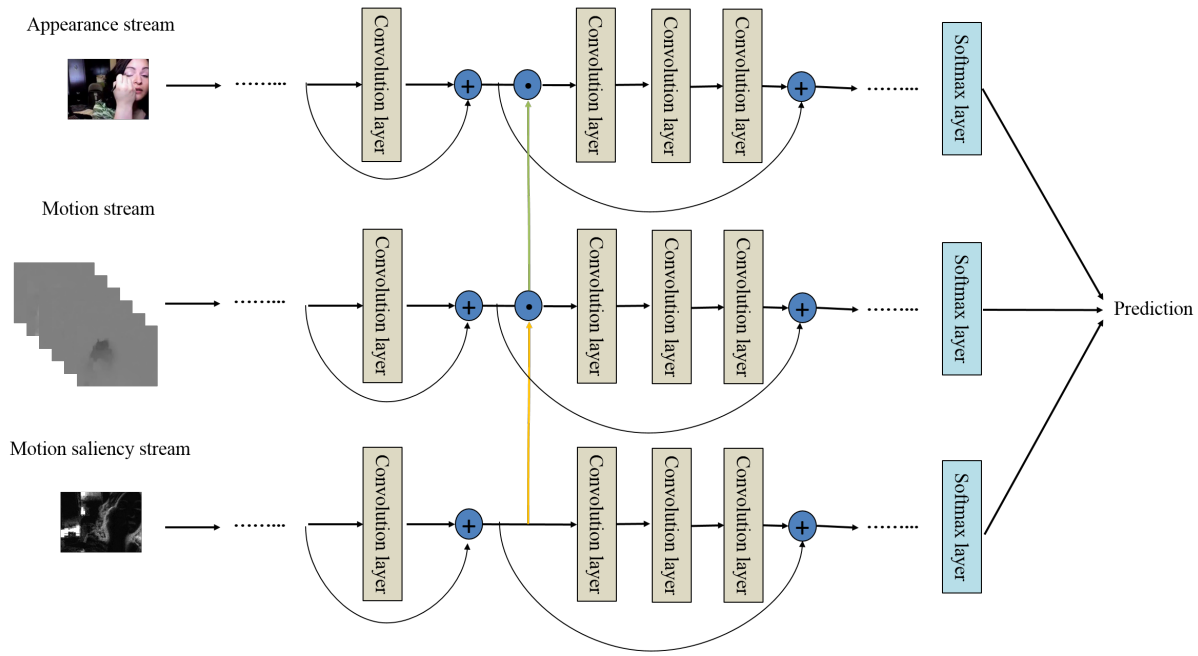

Figure 3.4: The framework of the proposed MSM-ResNets, which contains the appearance stream, motion stream and motion saliency stream. An average fusion manner is adopted on the softmax layer for the final prediction.

the effectiveness of the proposed MSM-ResNets in Section 3.5.2. Finally, we compare our proposed MSM-ResNets with the state-of-the-art in Section 3.5.3.

3.5.1 Experimental preliminary

In this section, we first introduce experimental datasets for action recognition in Section 3.5.1.1. Then we introduce the training procedure in Section 3.5.1.2.

3.5.1.1 Experimental datasets

We evaluate our proposed MSM-ResNets with all other comparison methods on two standard action recognition datasets: UCF101 and HMDB51. The UCF101 dataset is a realistic action recognition dataset, which directly collects 13320 action videos from Youtube media and contains 101 action categories. Three train/test splits are provided for the UCF101 dataset, which can split the whole UCF101 dataset into the training dataset and test dataset. The training dataset contains about 9.4k action videos and test dataset contains about 3.7k action videos. HMDB51 dataset mostly collects action videos from movies and YouTube, which collects about 6.8k action videos and contains 51 action categories. Similar to the UCF101 dataset, HMDB51 dataset also provides three train/test

splits for splitting the training dataset and test dataset. All the experiments are performed according to the provided three different train/test splits, and we adopt the average classification accuracy over three splits as the final accuracy result for action recognition on the UCF101 and HMDB51 datasets [48].

3.5.1.2 Training procedure

We use Python to implement the proposed MSM-ResNets based on deep learning framework Pytorch. All the experiments are conducted in the Ubuntu 16.04 Operation System environment. We train our model on four Nivida GTX 1080Ti GPUs. The initial value of learning rate is set 0.01. We use mini-batch stochastic gradient descent (SGD) with momentum value 0.9 to train the proposed MSM-ResNets and the batch size is set 64. All the inputs (including RGB video frames, optical flow frames and motion saliency frames) are resized into 224×224. To improve the generalization ability of the proposed MSM-ResNets, we perform data augmentation according to [46]. We randomly crop the input sample from five different positions: top left corner, top right corner, bottom left corner, bottom right corner and center. In addition, we also randomly horizontally flip the input sample with 50% probability. Besides, we use the pretrained ResNets on ImageNet dataset [19] to initialize the backbone network and backpropagation is adopted to training the network.

For the appearance stream, the input of the appearance stream is a single RGB video frame. For the motion stream, we use an effective and efficient optical flow estimation method [1] to extract the corresponding optical flow frames from multiple consecutive RGB video frames. The input of motion stream is 2L consecutive optical flow frames, which usually includes L vertical optical flow frames and L horizontal optical flow frames (the value of L usually is set as 10). Similar to the appearance stream, the input of the motion saliency stream is a single motion saliency frame, which is extracted from the RGB video frames by the motion saliency detection method used in [4]. According to Figure 4, we train each stream with SGD and obtain the corresponding softmax score. Then an average fusion is applied on all the softmax scores of the appearance stream, motion stream and motion saliency stream. Backpropagation is applied to adjust and update the proposed MSM-ResNets during the training process.

We also illustrate the loss computing process of the proposed MSM-ResNets in Figure 3.5. In detail, for the appearance stream, we can obtain the corresponding softmax score. Then we can compute the prediction and the ground truth to obtain the corresponding loss of the appearance stream, which can be denoted as $loss_a$. For the motion stream and the motion saliency stream, we can also obtain the corresponding loss of the motion stream and the motion saliency stream, which can be denoted as $loss_m$ and $loss_{ms}$. We sum up the losses of these streams and obtain the final total loss of the proposed MSM-ResNets, which can be denoted as $loss_{sum}$.

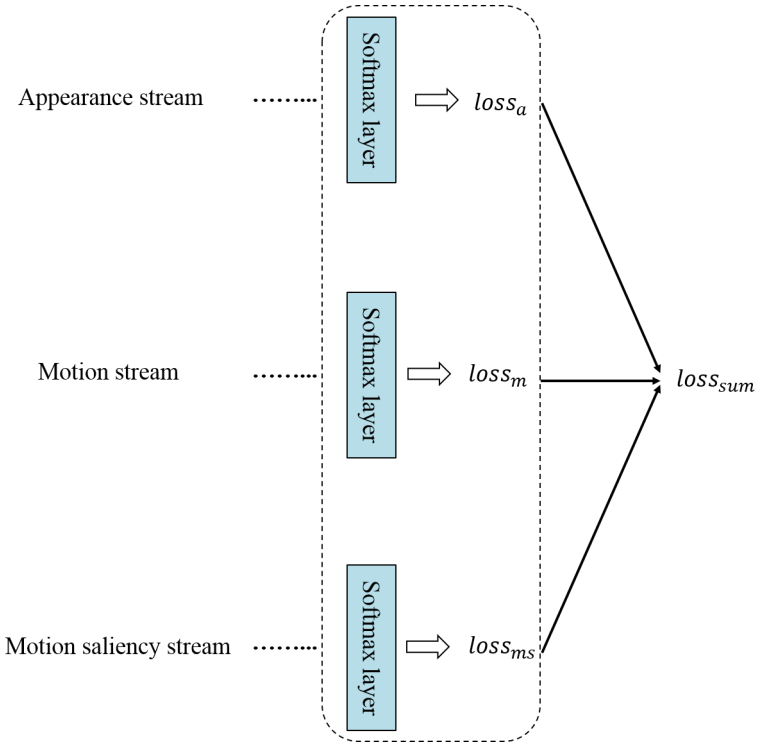

Figure 3.5: The loss computing process of the proposed MSM-ResNets. $loss_a$ denotes the loss of the appearance stream, $loss_m$ denotes the loss of the motion stream, $loss_{ms}$ denotes the loss of the motion saliency stream, and $loss_{sum}$ denotes the total loss of all streams.

3.5.2 Analysis of ablation experiments

To verify the effectiveness of the proposed MSM-ResNets, we set up a set of ablation experiments. For a fair comparison, all the comparison methods (including two-stream CNNs, spatiotemporal multiplier networks and the proposed MSM-ResNets) adopt 34-layer residual network as the backbone network. The first comparison method is the two-stream convolutional neural networks (two-stream CNNs) [29], which is the baseline network architecture for two-stream CNNs based methods. The second comparison method is the spatiotemporal multiplier networks in [8], which injects multiplicative interactions from the motion stream to the appearance stream based on two-stream CNNs. The third comparison method is our proposed MSM-ResNets, which is based on the spatiotemporal multiplier networks [8]. In addition, we average the results of the softmax layers of different streams as the final prediction. Table 3.2 illustrates the ablation experimental results of the comparison methods including the baseline two-stream CNNs, the spatiotemporal multiplier networks and the proposed MSM-ResNets on the standard UCF101 and HMDB51 datasets.

According to Table 3.2, we can find that the proposed MSM-ResNets obtains the best performance compared with the baseline two-stream CNNs and the spatiotemporal mul-

Table 3.2: Classification accuracy of the comparison methods: two-stream CNNs, spatiotemporal multiplier networks and the proposed MSM-ResNets on the UCF101 and HMDB51 datasets.

Methods (34-layer)	UCF101	HMDB51
Two-stream CNNs	87.2%	58.9%
Spatiotemporal multiplier networks	91.3%	63.8%
MSM-ResNets	**93.5%**	**66.7%**

tiplier networks on both the UCF101 and HMDB51 datasets. In detail, compared with the spatiotemporal multiplier networks, the proposed MSM-ResNets improves 2.2% and 2.9% on UCF101 and HMDB51 datasets, respectively. The reason is that the proposed MSM-ResNets can capture the salient action information and suppress the motion noise in the background (*e.g.*, slight leaf jittering or slight water rippling) [4, 24] by the novel developed motion saliency stream, which can transmit motion saliency cue from the motion saliency stream to the motion stream by injecting multiplicative connections. This verifies the effectiveness of the proposed MSM-ResNets because we keep the same settings (including the same depth, the same backbone network and the same fusion manner on the softmax layer) for the spatiotemporal multiplier networks and the proposed MSM-ResNets. Compared with conventional two-stream CNNs, the proposed MSM-ResNets significantly improves 6.3% and 7.8% on the UCF101 and HMDB51 datasets, respectively. This verifies the superior performance of the proposed MSM-ResNets again.

We also illustrate the training loss of the two-stream CNNs, spatiotemporal multiplier networks and the proposed MSM-ResNets over 100 epochs on the UCF101 dataset in Figure 3.6. We can find that the proposed MSM-ResNets obtains the fastest convergence rate, then follows the spatiotemporal multiplier networks and two-stream CNNs, which shows that the proposed MSM-ResNets converges faster than the spatiotemporal multiplier networks and two-stream CNNs.

3.5.3 Comparison with the state-of-the-art

Except the above ablation studies, we also compare the proposed MSM-ResNets with the state-of-the-art methods including conventional two-stream CNNs method [29], spatiotemporal multiplier networks (named SMN for short) [8], convolutional two-stream network (named CTN for short) [10], 3D convolutional networks (named C3D for short) [39], Long-term temporal convolution network (named LTCN for short) [41], spatiotemporal saliency based multi-stream networks (named STS-network for short) [23] and spatiotemporal saliency based multi-stream networks with attention-aware LSTM (named STS-ALSTM for short) [22]. The comparison results of the proposed MSM-ResNets with the state-of-the-art methods on the UCF101 and HMDB51 datasets are reported in Table 3.3.

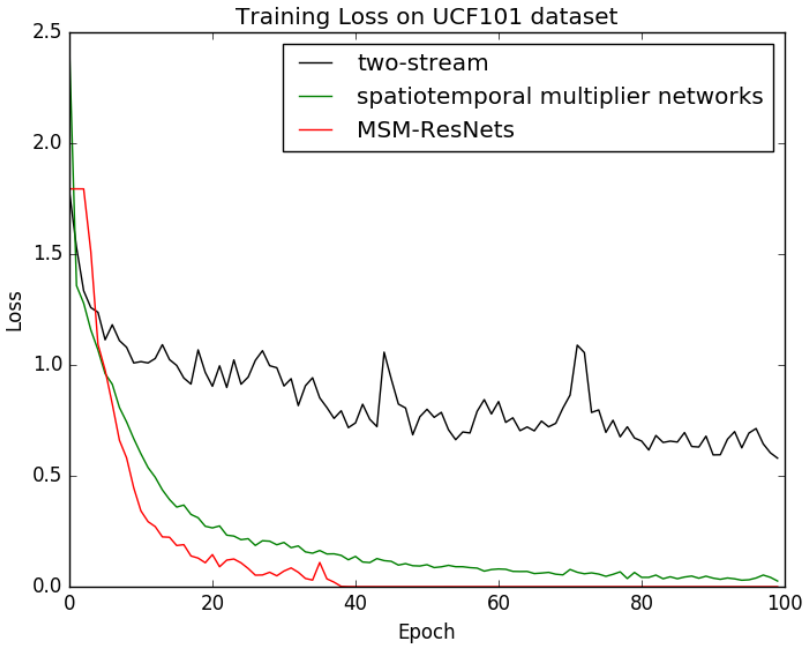

Figure 3.6: Training loss of two-stream CNNs, spatiotemporal multiplier networks and the proposed MSM-ResNets on the UCF101 dataset.

According to Table 3.3, we can find that the proposed MSM-ResNets achieves the best performance compared with other comparison methods including two-stream CNNs, SMN (34-layer), CTN, C3D, LTCN, STS-network and STS-ALSTM on both the UCF101 and HMDB51 datasets. This shows that our MSM-ResNets can effectively improve the accuracy of action recognition. In detail, for the UCF101 dataset, the proposed MSM-ResNets improves 6.6% (vs two-stream CNNs), 0.7% (vs SMN (34-layer)), 1.0% (vs CTN), 8.3% (vs C3D), 1.8% (vs LTCN), 3.4% (vs STS-network) and 0.8% (vs STS-ALSTM), respectively. For the HMDB51 dataset, the proposed MSM-ResNets improves 8.7% (vs two-stream CNNs), 1.5% (vs SMN (34-layer)), 1.3% (vs CTN), 1.9% (vs LTCN), 4.3% (vs STS-network) and 2.3% (vs STS-ALSTM), respectively. Firstly, we can find that, compared with the classic two-stream CNNs, our MSM-ResNets significantly improves 6.6% and 8.7% on the UCF101 and HMDB51 datasets, respectively. This shows that the proposed MSM-ResNets can significantly improve the accuracy of action recognition. The reason is that the motion saliency stream can provide the salient motion information and can transmit motion saliency cue from the motion saliency stream to the motion stream, which can help improve the accuracy of action recognition. Secondly, we can find that our MSM-ResNets improves 0.7% and 1.5% compared with the SMN (34-layer) model on the UCF101 and HMDB51 datasets, respectively. This verified that the developed novel motion saliency stream of the proposed MSM-ResNets and the corresponding multiplicative injections from the motion saliency stream to the motion stream can improve the accuracy of action recognition. Thirdly, compared with the STS-network, the proposed MSM-ResNets improves 3.4% and

Table 3.3: The comparison results of the proposed MSM-ResNets with the state-of-the-art methods on the UCF101 and HMDB51 datasets.

Methods	UCF101	HMDB51
Two-stream CNNs [29]	86.9%	58.0%
SMN (34-layer) [8]	92.8%	65.2%
CTN [10]	92.5%	65.4%
C3D [39]	85.2%	-
LTCN [41]	91.7%	64.8%
STS-network [23]	90.1%	62.4%
STS-ALSTM [22]	92.7%	64.4%
MSM-ResNets	**93.5%**	**66.7%**

4.3% on the UCF101 and HMDB51 datasets, respectively. This shows the interactions between different streams can help improve the accuracy of action recognition. Because the STS-network only adds a spatiotemporal saliency stream based on the conventional two-stream network, and there are not any interactions between different streams (*i.e.* the appearance stream, the motion stream and the spatiotemporal saliency stream) of the STS-network.

Although the proposed MSM-ResNets outperforms other compared state-of-the-art models, the time cost of the MSM-ResNets is expensive at the data pre-processing stage. The compared C3D model directly feeds the RGB frames into the 3D convolutional neural network and there are not any data pre-processing at the input stage. Compared with the C3D model, the proposed MSM-ResNets needs computing optical flow frames and motion saliency maps from the consecutive RGB frames, which are time-consuming.

3.6 Summary

In this chapter, we propose a Motion Saliency based multi-stream Multiplier ResNets (MSM-ResNets) for action recognition. The proposed MSM-ResNets model consists of three interactive streams: the appearance stream, motion stream and motion saliency stream. There are two kinds of different multiplicative connections between these streams: from the motion stream to the appearance stream and from the motion saliency stream to the motion stream. The proposed MSM-ResNets can capture the salient motion information and suppress the motion noise in the background by the novel developed motion saliency stream. Experimental results verify the effectiveness of the proposed MSM-ResNets and achieves competitive performance with other state-of-the-art methods on both the UCF101 and HMDB51datasets.

References

[1] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision*, pages 25–36. Springer, 2004.

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[3] Zhe Chen, Ruili Wang, Zhen Zhang, Huibin Wang, and Lizhong Xu. Background–foreground interaction for moving object detection in dynamic scenes. *Information Sciences*, 483:65–81, 2019.

[4] Zhe Chen, Xin Wang, Zhen Sun, and Zhijian Wang. Motion saliency detection using a temporal fourier transform. *Optics & Laser Technology*, 80:1–15, 2016.

[5] Cheng Dai, Xingang Liu, and Jinfeng Lai. Human action recognition using two-stream attention based LSTM networks. *Applied Soft Computing*, 86:105820, 2020.

[6] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[8] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768–4777, 2017.

[9] Christoph Feichtenhofer, Axel Pinz, Richard P Wildes, and Andrew Zisserman. What have we learned from deep representations for action recognition? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7844–7853, 2018.

[10] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[11] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[12] Yuanhao Gong. Mean curvature is a good regularization for image processing. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(8):2205–2214, 2018.

[13] Yuanhao Gong and Orcun Goksel. Weighted mean curvature. *Signal Processing*, 164:329–339, 2019.

[14] Yuanhao Gong and Ivo F Sbalzarini. Curvature filters efficiently reduce certain variational energies. *IEEE Transactions on Image Processing*, 26(4):1786–1798, 2017.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016.

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[21] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[22] Zhenbing Liu, Zeya Li, Ruili Wang, Ming Zong, and Wanting Ji. Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. *Neural Computing and Applications*, 32(18):14593–14602, 2020.

[23] Zhenbing Liu, Zeya Li, Ming Zong, Wanting Ji, Ruili Wang, and Yan Tian. Spatiotemporal saliency based multi-stream networks for action recognition. In *Asian Conference on Pattern Recognition*, pages 74–84. Springer, 2019.

[24] Léo Maczyta, Patrick Bouthemy, and Olivier Le Meur. CNN-based temporal detection of motion saliency in videos. *Pattern Recognition Letters*, 128:298–305, 2019.

[25] Pourya Shamsolmoali, Xiaofang Li, and Ruili Wang. Single image resolution enhancement by efficient dilated densely connected residual network. *Signal Processing: Image Communication*, 79:13–23, 2019.

[26] Pourya Shamsolmoali, Masoumeh Zareapoor, Ruili Wang, Deepak Kumar Jain, and Jie Yang. G-ganisr: Gradual generative adversarial network for image super resolution. *Neurocomputing*, 366:140–153, 2019.

[27] Pourya Shamsolmoali, Masoumeh Zareapoor, Ruili Wang, Huiyu Zhou, and Jie Yang. A novel deep structure u-net for sea-land segmentation in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(9):3219–3232, 2019.

[28] Pourya Shamsolmoali, Masoumeh Zareapoor, Huiyu Zhou, Ruili Wang, and Jie Yang. Road segmentation for remote sensing images using adversarial spatial pyramid networks. *arXiv preprint arXiv:2008.04021*, 2020.

[29] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[31] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[32] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[33] Yan Tian, Tao Chen, Guohua Cheng, Shihao Yu, Xi Li, Jianyuan Li, and Bailin Yang. Global context assisted structure-aware vehicle retrieval. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[34] Yan Tian, Guohua Cheng, Judith Gelernter, Shihao Yu, Chao Song, and Bailin Yang. Joint temporal context exploitation and active learning for video segmentation. *Pattern Recognition*, 100:107158, 2020.

[35] Yan Tian, Judith Gelernter, Xun Wang, Jianyuan Li, and Yizhou Yu. Traffic sign detection using a multi-scale recurrent attention network. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4466–4475, 2019.

[36] Yan Tian, Xun Wang, Jiachen Wu, Ruili Wang, and Bailin Yang. Multi-scale hierarchical residual network for dense captioning. *Journal of Artificial Intelligence Research*, 64:181–196, 2019.

[37] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. LSTM-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

[38] Yan Tian, Yujie Zhang, Di Zhou, Guohua Cheng, Wei-Gang Chen, and Ruili Wang. Triple attention network for video segmentation. *Neurocomputing*, 2020.

[39] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[40] Zhigang Tu, Wei Xie, Qianqing Qin, Ronald Poppe, Remco C Veltkamp, Baoxin Li, and Junsong Yuan. Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition*, 79:32–43, 2018.

[41] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2017.

[42] Dianhui Wang and Caihao Cui. Stochastic configuration networks ensemble with heterogeneous features for large-scale data analytics. *Information Sciences*, 417:55–71, 2017.

[43] Dianhui Wang and Ming Li. Stochastic configuration networks: Fundamentals and algorithms. *IEEE Transactions on Cybernetics*, 47(10):3466–3479, 2017.

[44] Dianhui Wang and Ming Li. Deep stochastic configuration networks with universal approximation property. In *International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2018.

[45] Huihui Wang, Yang Gao, Yinghuan Shi, and Ruili Wang. Group-based alternating direction method of multipliers for distributed linear classification. *IEEE Transactions on Cybernetics*, 47(11):3568–3582, 2016.

[46] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and

Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[47] Jianfei Yin, Ruili Wang, Shunda Ju, Yizhe Bai, and Joshua Zhexue Huang. An asymptotic statistical learning algorithm for prediction of key trading events. *IEEE Intelligent Systems*, 35(2):25–35, 2020.

[48] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1774–1785, 2017.

[49] Hao Zheng, Ruili Wang, Wanting Ji, Ming Zong, Wai Keung Wong, Zhihui Lai, and Hexin Lv. Discriminative deep multi-task learning for facial expression recognition. *Information Sciences*, 2020.

Chapter 4

Spatial and temporal saliency based four-stream network with multi-task learning for action recognition

A novel spatial and temporal saliency based four-stream network with multi-task learning is proposed for action recognition. The proposed model comprises two parts: (i) The first part is a spatial and temporal saliency based four-stream network, which comprises four streams: an appearance stream, a motion stream, a novel spatial saliency stream, and a novel temporal saliency stream. The novel spatial saliency stream is used to acquire spatial saliency information and the novel temporal saliency stream is used to acquire temporal saliency information. (ii) The second part is a multi-task learning based long short-term memory network (LSTM), which adopts the feature representations obtained by obtained convolutional neural networks (CNN) as input. The multi-task learning based LSTM can share the complementary knowledge between different streams and capture long-term dependency relationships of the consecutive frames over temporal evolution. We conduct experiments on UCF101, HMDB51 and Kinetics datasets to verify the effectiveness of our network, and demonstrate that our network has better performance than the state-of-the-art methods in recognizing actions.

The organization of this chapter is as follows. In Section 4.1, we introduce the motivation of the proposed approach. The related work of two-stream based models and Recurrent Neural Network (RNN) based models for action recognition is introduced in Section 4.2. Then we present the proposed model for action recognition in Section 4.3. The implementation details of the proposed model are provided in Section 4.4. In Section 4.5, we discuss our experimental results. Finally, a conclusion is drawn in Section 4.6.

4.1 Introduction

Convolutional Neural Networks (CNNs) have been successfully used to solve many image processing tasks such as image recognition, object detection and object segmentation [12, 15, 16]. CNN can learn deep hierarchical features through multiple progressive convolutional layers, which are different from shallow hand-crafted features and more suitable for various visual recognition tasks. Inspired by the successful use of CNNs in image processing, various CNN-based methods were developed for video processing such as video action recognition [4, 26].

Action recognition aims to recognize and categorize actions in videos. Different from static images, videos have both spatial component and temporal component. The temporal component contains motion information, which is important for recognizing actions. Two-stream models are a classic CNN-based model for action recognition [26], which comprises two streams: (i) a temporal stream for acquiring the motion information, and (ii) a spatial stream for acquiring the appearance information. Although various two-stream models [9, 11, 26] were developed in action recognition, there are two issues to be considered in two-stream models:

(i) *Videos usually contain useless backgrounds such as clutters and unrelated background objects, which may affect and mislead the recognition of desirable actions.* For this issue, Liu *et al.* [21] considered capturing the spatiotemporal foreground object information to improve recognition accuracy. Wang *et al.* [33] considered extracting the salient features as an input instead of using an entire video frame. Tu *et al.* [30] developed a multi-stream CNN architecture, which extracted human-related regions from video frames and the corresponding optical flow frames as complementary input. However, most of the existing saliency based methods mainly focus on extracting the salient object information while ignoring the salient motion information, which is important for improving the accuracy of action recognition.

(ii) *The fusion of spatial information and temporal information in two-stream models is important.* For this issue, Feichtenhofer *et al.* [11] considered fusing spatial information with temporal information on a specified convolution layer instead of the final softmax layer. Ming *et al.* [38] considered injecting multiplicative connections between different stream to transmit different spatiotemporal cues. These methods lack the ability of modelling long-term dependency relationships of consecutive frames. Liu *et al.* [21] considered using attention-aware LSTMs to capture the long-term dependency relationships of different streams (*i.e.* the appearance stream, motion stream and spatiotemporal saliency stream). An average fusion of the predictions of different LSTMs is adopted. However, this method only simply averages the predictions of different stream as the fusion result, which cannot take fully advantage of the complementary knowledge among different streams.

According to the above analysis, this chapter proposes a novel spatial and temporal saliency based four-stream network with multi-task learning for action recognition. Our proposed

model comprises two parts: (i) The first part is the spatial and temporal saliency based four-stream network, which consists of four streams: an appearance stream, a motion stream, a novel spatial saliency stream, and a novel temporal saliency stream. The novel spatial saliency stream is used for acquiring the salient object cue while ignoring the background. The novel temporal saliency stream is used for acquiring the salient motion cue over long temporal periods and suppressing motion noises. (ii) The second part is a multi-task learning based LSTM, which adopts the feature representations obtained by convolutional neural networks (CNN) as input. The multi-task learning based LSTM can share the complementary knowledge between different streams and model/capture the long-term dependency relationship of the consecutive frames over temporal evolution.

The proposed model contains Multi-cue (*i.e.* the spatial cue, temporal cue, spatial saliency cue and the temporal saliency cue), Multi-stream and Multi-task learning. Thus, we name the proposed model as the 3M model. Different from the conventional CNN features-based LSTM methods for action recognition [35], our proposed 3M model obtains the CNN features of each stream using the proposed four-stream network and utilizes an LSTM to share different inputs for the entire network training.

The key contributions of this chapter can be categorized as follows:

- We propose a novel spatial saliency stream to capture the salient object information from videos for recognizing actions.
- We propose a temporal saliency stream to capture the salient motion information from videos for recognizing actions.
- Our 3M model can model/capture different long-term dependency relationships of different cues (*i.e.* the spatial cue, temporal cue, spatial saliency cue and the temporal saliency cue) by LSTMs. Further, it can fuse different cues as a whole by multi-task learning.

4.2 Related work

4.2.1 Two-stream based model for action recognition

The first two-stream based model for action recognition was introduced by Simonyan and Zisserman [26], which comprised a temporal stream and a spatial stream. The spatial stream used the raw video frames as input fed into a CNN to capture appearance information. The temporal stream uses the extracted optical flow frames as an input fed into a CNN to capture the motion information. Then an average of the outputs of both the spatial and temporal stream was adopted as the final prediction. However, this model only considered fusing the spatiotemporal information in the output stage while ignored the spatiotemporal information fusions over temporal evolutions.

To explore the spatiotemporal information fusion, Karpathy *et al.* [18] proposed three different fusion methods, named early fusion, late fusion and slow fusion for action recognition. These fusion methods mainly inferred the motion information implicitly from consecutive multiple video frames because they only adopted the RGB frames as input. Feichtenhofer *et al.* [11] fused spatial feature information with temporal feature information on a specified convolution layer rather than on the final softmax layer. In [9], a spatiotemporal residual network was proposed by injecting residual connections between two streams. Feichtenhofer *et al.* [10] considered learning long-term temporal relationships by building multiplicative connections between different streams. Shi *et al.* [24] developed a data-driven two-stream graph convolutional network, which could utilize the flexibility of graph convolutional network to improve the performance.

To directly capture the spatiotemporal information from videos, a two-stream 3D CNN was proposed in [4]. Carreira and Zisserman [4] extended a 2D two-stream ConvNets into a 3D two-stream ConvNets for action recognition, which could seamlessly learn spatiotemporal representation features of video actions. To reduce the expensive computation cost of the two-steam 3D convolutional neural networks proposed in [4], Xie *et al.* [34] kept the 3D spatiotemporal convolutions on the low layers or the top layers of the two-stream 3D convolutional neural networks, and used 2D convolutions to replace the 3D convolutions in the remaining 3D convolution layers. In [8], a SlowFast network was developed for action recognition, which consisted of a slow pathway and a fast pathway. The slow pathway operated at a low frame rate is responsible for learning the spatial features and the fast pathway operated at a high frame rate is responsible for learning the temporal features.

However, it is difficult for two-stream CNN based models to directly capture the long-term dependency relationship of consecutive frames over the whole temporal evolution.

4.2.2 RNN based model for action recognition

Recurrent Neural Network (RNN) is a natural architecture for processing sequential data, which has been successfully applied in lots of sequential tasks such as language modelling, machine translation and video description [3, 31]. For action recognition, RNN based models usually comprise two steps: (i) extracting features from video frames; (ii) feeding the extracted features into RNNs for classification. For the feature extractor, CNNs are becoming more and more popular compared with conventional shallow methods [23, 32] in action recognition. For the classifier, LSTM is usually adopted for solving the long-term dependency problem existing in the standard RNNs.

Various RNN based models for action recognition have been developed. Baccouche *et al.* [1] utilized the visual Bag of words (visual BoW) method for feature extraction and adopted an LSTM as a classifier for human action recognition. Unlike the previous shallow feature extraction methods, Ng *et al.* [35] proposed to use CNNs as the feature extractors to extract the appearance features from the consecutive video frames and the motion features

from the stacked optical flow frames. Then the extracted appearance features and motion features were fed into an individual LSTM for predicting the action category, respectively. An average strategy was adopted for fusing the outputs of both the LSTMs. Donahue *et al.* [7] developed a long-term recurrent convolutional network for visual recognition, which combined CNN and LSTM as a whole to jointly train them simultaneously. In [2], Baccouche *et al.* extended the 2D ConvNets into 3D ConvNets to learn the spatiotemporal features, and then an RNN was applied for modelling the temporal evolution of learned spatiotemporal features. Si *et al.* [25] developed a graph convolutional LSTM to extract the discriminative spatial features and temporal features from videos. Dai *et al.* [6] proposed a two-stream attention-based LSTM for action recognition, which can pay attention to the discriminative features to improve the performance. Besides, Ma *et al.* [22] proposed a temporal segment LSTM to capture the correlations by concatenating spatial features and temporal features into feature matrices in the temporal dimension.

However, previous RNN based models ignored utilizing the complementary knowledge between different streams. Different from previous two-stream CNN based models and RNN based models, our 3M model can capture motion saliency information and take advantage of the complementary knowledge between different streams by multi-task learning.

4.3 Our proposed 3M model

The proposed spatial and temporal saliency based four-stream network with multi-task learning (named the 3M model for short) for action recognition consists of two parts: the first part is the Spatial and Temporal Saliency based Four-stream network (named the STSF model for short), and the second part is the multi-task learning based LSTM. In this section, we first give the preliminary about the used spatial saliency detection method [37], the temporal saliency detection method [5] and the principle of LSTM in Section 4.3.1. Then the first part of our proposed 3M model, *i.e.* the proposed STSF model, will be presented in Section 4.3.2. Finally, the proposed 3M model will be presented in Section 4.3.3.

4.3.1 Preliminary

4.3.1.1 Spatial saliency detection

In this section, spatial saliency detection denotes the saliency detection performed in the spatial dimension, *i.e.* the traditional saliency detection. The spatial saliency detection methods aim to detect the salient objects that are different from the surrounding environment, which is useful for many visual tasks and applications such as object detection, image segmentation and action recognition [21]. This chapter utilizes a fast and high-efficient

saliency detection method [37] for the saliency detection, which is robust to pixel-level fluctuation and runs at about 80 frames per second (FPS). We first use the saliency detection method in [37] to generate the *spatial saliency maps* from the stacked video frames. Figure 4.1 illustrates the generated *spatial saliency maps* from the stacked video frames. Then we develop a novel spatial saliency stream with the obtained spatial saliency maps as an input, which is a part of the proposed STSF model.

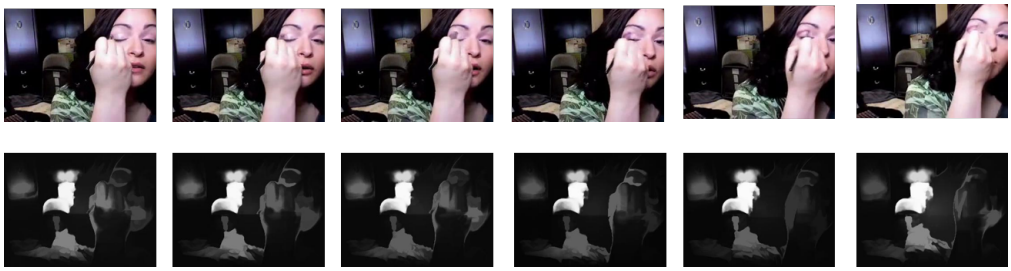

Figure 4.1: The generated *spatial saliency maps* from stacked video frames by a spatial saliency detection method [37].

4.3.1.2 Temporal saliency detection

In this section, temporal saliency detection denotes the saliency detection performed in the temporal dimension, *i.e.* the motion saliency detection. Motion saliency detection methods are capable of capturing the salient moving objects in a video while ignoring video backgrounds [17], which have been successfully applied in various vision-based tasks [13]. This chapter utilizes a fast and efficient motion saliency detection method [5] to identify salient moving pixels by the obtained phase spectrum in the frequency domain. This method can capture the salient moving objects over a long-term period. We first use this method to generate the *temporal/motion saliency maps* from the stacked video frames. Figure 4.2 illustrates the *temporal/motion saliency maps* from the stacked video frames. Then we develop a novel temporal/motion saliency stream with the temporal saliency maps as an input, which is a part of the proposed STSF model.

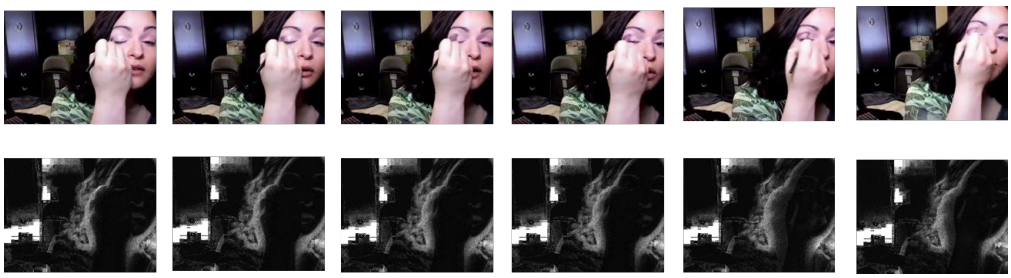

Figure 4.2: The generated *temporal/motion saliency maps* from stacked video frames by a temporal Fourier transform based motion saliency detection method [5].

4.3.1.3 LSTM classifier

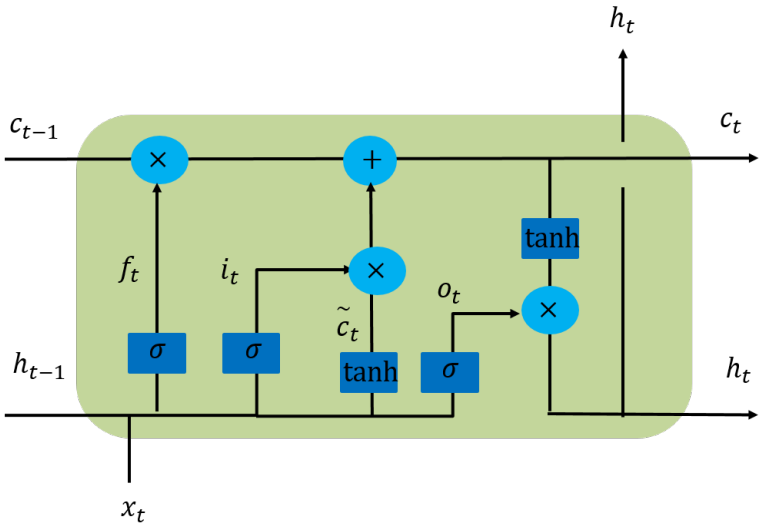

Figure 4.3: The basic unit of LSTM.

The basic unit of LSTM can be illustrated in Figure 4.3. The core of LSTM is to maintain and update a cell state c_t at each time step t, which consists of three gates: forget gate f_t, input gate i_t and output gate o_t. f_t is responsible for forgetting the old useless information; i_t is responsible for updating the new information; o_t is responsible for what information will be output. The definitions of the three gates are shown as follows:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f), \tag{4.1}$$

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i), \tag{4.2}$$

$$\overline{c}_t = tanh(W_{\overline{c}x}x_t + W_{\overline{c}h}h_{t-1} + b_{\overline{c}}), \tag{4.3}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \overline{c}_t, \tag{4.4}$$

$$o_t = \sigma(W_{ox}x_t + W_{oh}h_{t-1} + b_o), \tag{4.5}$$

$$h_t = o_t \odot tanh(c_t), \tag{4.6}$$

where x_t denotes the input at time step t; $\overline{c}_t$ is the candidate cell state at time step t; h_t represents the hidden state at time step t; W denotes the hidden state at time step t; b represents the weight matrix, σ is the bias; $tanh$ denotes the sigmoid function; $tanh$ denotes the hyperbolic tangent function; $\odot$ denotes the element-wise product.

4.3.2 The first part of our proposed 3M model: the STSF model

The architecture of the STSF model is illustrated in Figure 4.4. In the pre-processing stage, the consecutive video frames are extracted from the video data. Then the corresponding optical flow frames, saliency maps and motion saliency maps can be obtained from consecutive video frames. Note that the optical flow frames in our experiments are obtained by the OpenCV implementation based the TVL1 optical flow method [36], which is an efficient optical flow method. The saliency maps and motion saliency maps are generated by the saliency detection method [37] and the motion saliency detection method [5] introduced in Section 4.3.1.

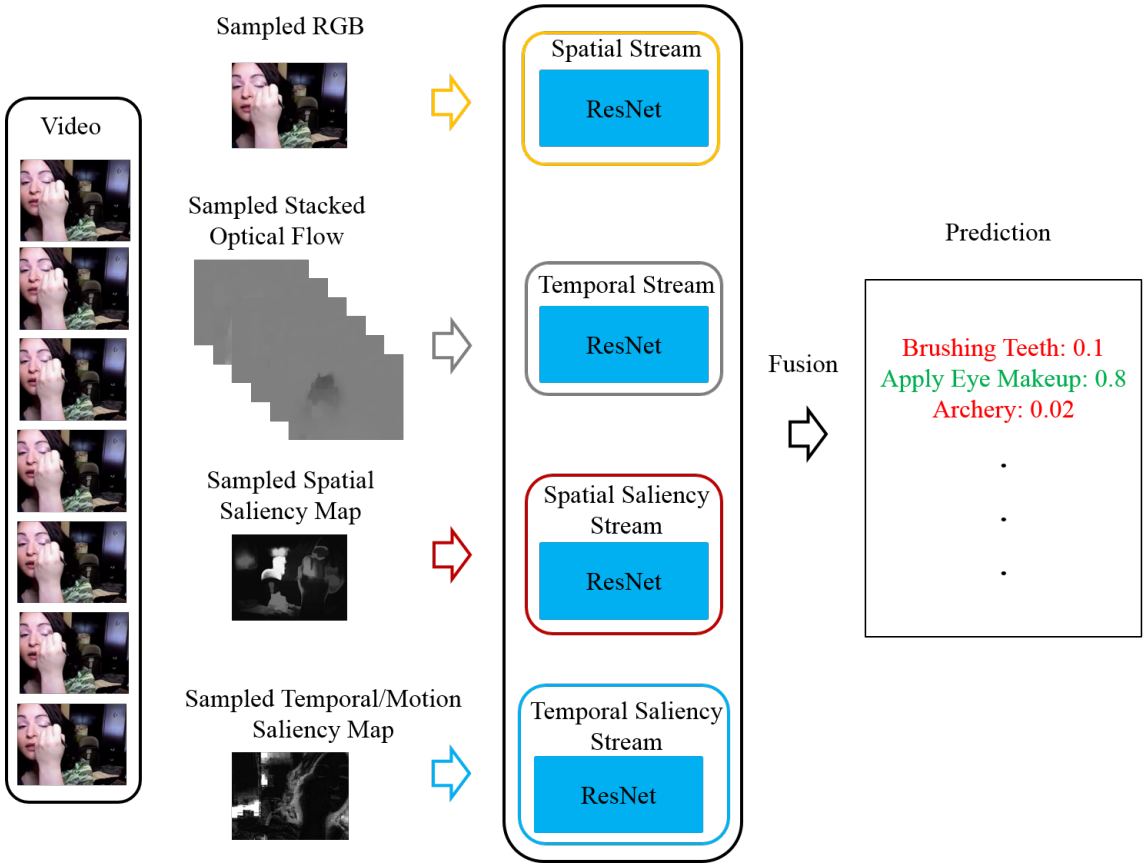

Figure 4.4: The architecture of the first part of our 3M model: the STSF model.

The network architecture of the proposed STSF model contains four streams: the spatial stream, temporal stream, spatial saliency stream and temporal saliency stream. All streams adopt ResNet as the backbone network, which is a widely used excellent network in action recognition [4, 9].

The spatial stream: The spatial stream captures the global spatial information from videos by using the sampled RGB video frames as the input. For this stream, we first sample

multiple RGB video frames at certain intervals and input them one by one into the network. Then we average the losses of the spatial stream of all the selected sampled RGB video frames as the final loss of the spatial stream. The prediction output of the spatial stream is denoted as $y^{spatial}$.

The temporal stream: The temporal stream captures the global motion information of each pixel by using the sampled stacked optical flow frames as the input. For this stream, we first sample $2L$ stacked optical flow frames including L stacked optical flow frames in the vertical direction and L stacked optical flow in the horizontal direction. In our experiments, the value of L is set to 10. Then we feed the sampled stacked optical flow frames into the temporal stream to obtain the output of the temporal stream denoted as $y^{temporal}$.

The spatial saliency stream: The spatial saliency stream captures the salient object information from videos by using the sampled saliency maps as the input. Similar to the settings in the spatial stream, we sample multiple spatial saliency maps and average the corresponding losses. The final output of the spatial saliency stream can be denoted as $y^{spatialsaliency}$.

The temporal saliency stream: The temporal saliency stream captures the salient motion information from videos over long periods by using the sampled motion saliency maps as the input. Similar to the settings in the spatial stream and the spatial saliency stream, for the temporal saliency stream, we sample multiple temporal saliency maps and average the corresponding losses. The final output of the temporal saliency stream can be denoted as $y^{temporalsaliency}$.

We train each stream separately and the cross-entropy loss is adopted as the loss function for each stream in the STSF model. Taking the spatial stream as an example, the loss function of the spatial stream can be defined as follows:

$$Loss = -\Sigma_i y_i log(y_i^{spatial}), \tag{4.7}$$

where $Loss$ denotes the cross-entropy loss function; y_i represents the true class value of class i, and $y_i^{spatial}$ denotes the final prediction output of the spatial stream.

An average fusion strategy is adopted for fusing the outputs of the four streams in the STSF model. Thus, the final output of the STSF model can be defined as follows:

$$\widehat{y} = \frac{1}{4}(y^{spatial} + y^{temporal} + y^{spatialsaliency} + y^{temporalsaliency}), \tag{4.8}$$

where $\widehat{y}$ denotes the final prediction output of the STSF model.

4.3.3 Our proposed 3M model

Taking the advantages of CNN and LSTM, we propose a novel spatial and temporal saliency based four-stream network with multi-task learning (*i.e.* the 3M model) for action recogni-

tion. The framework of the 3M model is illustrated in Figure 4.5. The 3M model consists of two parts: a four-stream CNN feature extractor and a multi-task learning based LSTM. In Section 4.3.3.1, we first introduce the proposed four-stream CNN feature extractor and then we introduce the multi-task learning based LSTM for action recognition in Section 4.3.3.2.

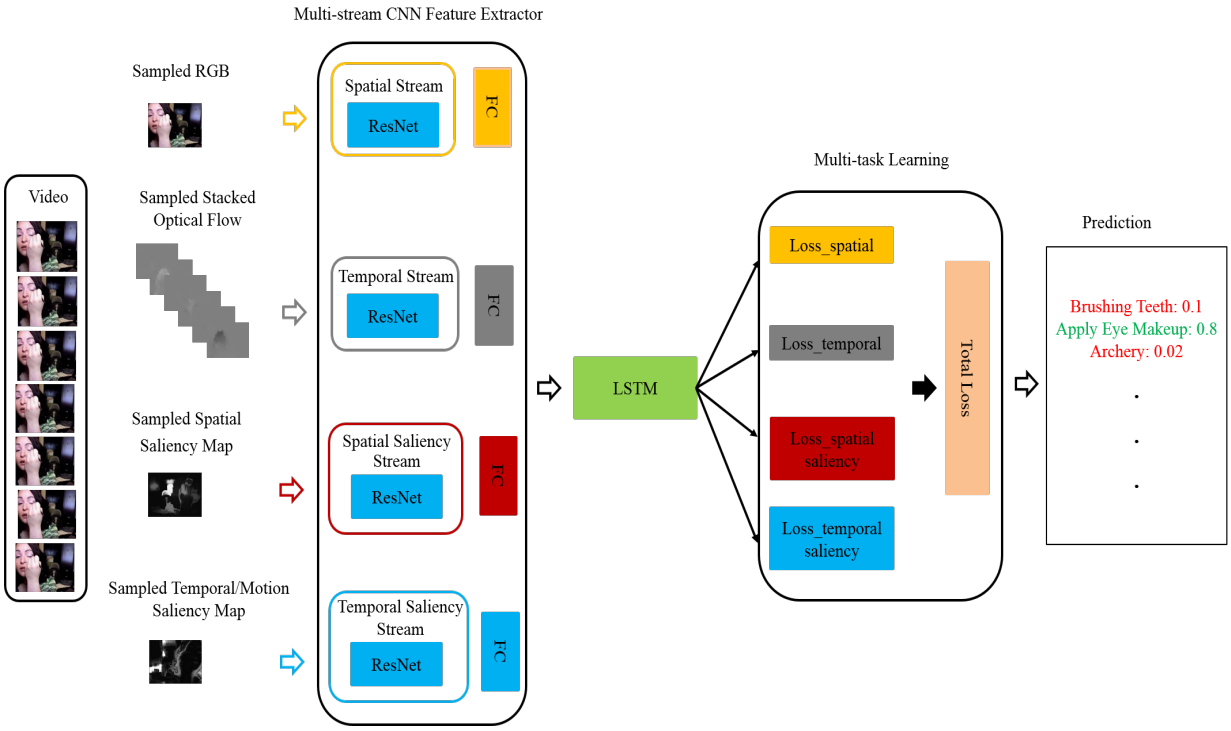

Figure 4.5: The framework of the proposed 3M model.

4.3.3.1 Four-stream CNN feature extractor

The front end of the proposed 3M model is the four-stream CNN feature extractor. We use a well-trained STSF model as the CNN feature extractor. In our proposed 3M model, ResNet is adopted as the backbone CNN. ResNet contains only one fully-connected layer, *i.e.* the softmax layer. To better obtain the CNN feature representations of ResNet, an extra fully-connected layer is added before the softmax layer. The new added fully connected layer can be regarded as the feature representation layer of CNN. The details of the fully-connected layer can refer to Section 4.4.

The workflow of the proposed four-stream CNN feature extractor can be described as follows: Given a video, we first extract the consecutive video frames from the video. Then the corresponding optical flow frames, saliency maps and motion saliency maps are extracted from the consecutive video frames. Followed by the video frames, optical flow frames, saliency maps and motion saliency maps fed into the corresponding spatial stream,

temporal stream, spatial saliency stream and temporal saliency stream. The corresponding obtained fully-connected layer representations of different streams are denoted as $FC_{spatial}$, $FC_{temporal}$, $FC_{spatialsaliency}$ and $FC_{temporalsaliency}$, respectively. We use $FC_{spatial}$, $FC_{temporal}$, $FC_{spatialsaliency}$ and $FC_{temporalsaliency}$ as the final four-stream CNN feature representations, which are fed into the multi-task learning based LSTM classifier.

4.3.3.2 Multi-task learning based LSTM

The back-end of the proposed 3M model is a multi-task learning based LSTM classifier. Given a video, we first use the four-stream CNN feature extractor to extract the four-stream CNN feature vectors $FC_{spatial}$, $FC_{temporal}$, $FC_{spatialsaliency}$ and $FC_{temporalsaliency}$. Then we input the extracted CNN feature vectors into the corresponding LSTM classifier for action recognition. A native method is that we can input vectors into the corresponding LSTMs for training. An average fusion of the outputs of all the LSTMs. However, training each LSTM individually will ignore utilizing the relationship and complementary knowledge between the obtained four feature representations. Multi-task learning can combine different inputs to train the entire network at the same time and share knowledge between multiple tasks, which can be utilized to improve accuracy.

In our case, we modify an LSTM architecture so that it has four softmax classification layers. The four softmax classification layers are corresponding to four different inputs, *i.e.* the four vectors. Each softmax classification layer has its own loss function (including $Loss_{spatial}$, $Loss_{temporal}$, $Loss_{spatialsaliency}$ and $Loss_{temporalsaliency}$), which operates on its own respective input ($Loss_{spatial}$, $Loss_{temporal}$, $Loss_{spatialsaliency}$ and $Loss_{temporalsaliency}$). The total loss is computed as the sum of each task's loss, which can be defined as follows:

$$Loss_{total} = Loss_{spatial} + Loss_{temporal} + Loss_{spatialsaliency} + Loss_{temporalsaliency}, \tag{4.9}$$

where the total loss $Loss_{total}$ is adopted as the final loss for evaluating the margin between the prediction result and the actual result.

4.4 Implementation details

We first introduce the modified ResNet-34 architecture in Section 4.4.1, which is adopted as the backbone network of each stream. Then we present the implementation of multi-task learning based LSTM in Section 4.4.2. Finally, we present the training and recognition procedure in Section 4.4.3.

4.4.1 The architecture of the modified ResNet

ResNet is the backbone network in our proposed 3M model [16, 28]. In the implementation, ResNet-34 (containing 34 layers) is chosen for the STSF model while we modify ResNet-34 by adding a new fully-connected layer before the softmax layer for the 3M model. The modified ResNet-34 is also adopted as the backbone network for the STSF model. The new added fully-connected layer is regarded as the feature representation layer of CNN, which will be extracted by the four-stream CNN feature extractor of the 3M model. Thus, the modified ResNet-34 consists of 35 layers. The architecture of this modified ResNet-34 is simply expressed as (Conv1, MaxPooling, Conv2_3, Conv3_4, Conv4_6, Conv5_3, AveragePooling, new added FC and Softmax layer), where the x in Conv{2, 3, 4, 5}_x represents the multiple of the residual block [16]. The size of each input frame is resized into 224×224. The architecture of the modified ResNet-34 can be illustrated in Table 4.1.

4.4.2 The implementation of multi-task learning based LSTM

In the implementation of the 3M model, a multi-task learning based LSTM is adopted as the back-end classifier. We use a deep multi-task learning based LSTM architecture, which consists of five stacked hidden LSTM layers and four softmax classification layers. Each LSTM layer contains 512 memory cells. The input of the multi-task learning based LSTM is the extracted CNN features $FC_{spatial}$, $FC_{temporal}$, $FC_{spatialsaliency}$ and $FC_{temporalsaliency}$. The output of multi-task learning based LSTM is the average of the final predictions of four softmax classification layers.

4.4.3 Training and recognition procedure

The experimental environment is Ubuntu 18.04. Four NVIDIA Quadro RTX 6000 GPUs (24GB $\times$ 4) are used for performing experiments. We implement the proposed 3M model by Python and Pytorch. Mini-batch stochastic gradient descent is chosen to train our 3M model. The size of mini-batch is set to 256. The value of the learning rate is initially set to 0.01. An action video can be decompressed into hundreds or thousands of frames at a frame rate of 24 FPS (frames per second). Thus many decompressed adjacent frames are very close and redundant.

For action recognition, a common way is that randomly extracting video frames at a certain interval from the decompressed video frames to form a sample clip. In our experiments, for the spatial stream, spatial saliency stream and temporal saliency stream, the number of sampled frames is set to 16 and the input of each stream is a frame (size 224×224). For the temporal stream, we sample $2L$ frames (including L frames in the vertical direction and L frames in the horizontal direction) each time and sample 10 times to obtain 10 sample clips. The input of the temporal stream is $2L$ frames (size $224 \times 224 \times 2L$). For each

Table 4.1: The architecture of the modified ResNet-34. The residual building block is described in brackets. The input size is 224×224.

Layer	Layer information	Output size
Conv1	$[7 \times 7, 64]$, kernel: 7×7, feature map number: 64, stride: 2×2, padding: 3×3	112×112, feature map: 112×112
Maxpool	kernel: 3×3, stride: 2×2, padding: 1×1	56×56
Conv2_x	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$ for all convolutional layers: stride: 1×1 padding: 1×1	56×56
Conv3_x	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$ for the first convolutional layers: stride: 2×2 padding: 1×1, for other convolutional layers: stride: 1×1 padding: 1×1	28×28
Conv4_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$ for the first convolutional layers: stride: 2×2 padding: 1×1, for other convolutional layers: stride: 1×1 padding: 1×1	7×7
Conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$ for the first convolutional layers: stride: 2×2 padding: 1×1, for other convolutional layers: stride: 1×1 padding: 1×1	7×7
Averagepool	kernel: 7×7, stride: 1×1 padding: 0×0	1×1
New added FC	512×1024	-
Softmax	$1024 \times numberofclasses$	-

video, we do such sampling operations 10 times to form 10 sample clips. For each stream, we average the output predictions of 10 sample clips as the final prediction of the stream. Then we average the predictions as the final accuracy of action recognition.

For the proposed 3M model, the STSF model is adopted as the four-stream CNN feature extractor. We mainly train the multi-task learning based LSTM classifier with four softmax layers. Like the training strategy in the STSF model, for each sample clip, we can obtain four predictions of the four softmax layers. Since we sample 10 times for a video, thus we will obtain 40 (4×10) predictions through the multi-task learning based LSTM. Then we average them and take the average result as the final video-level prediction of the 3M model.

4.5 Experimental analysis

In this section, we present the experimental details. Firstly, we introduce experimental datasets (UCF101, HMDB51, and Kinetics) in Section 4.5.1. To verify the effectiveness of our STSF model, a set of ablation experiments are performed on UCF101 and HMDB51 in Section 4.5.2. Finally, in Section 4.5.3, we compare the performance of our 3M model with several state-of-the-art methods and discusses our experimental results.

4.5.1 Experimental datasets

In our experiments, three popular action recognition datasets UCF101 [27], HMDB51 [20] and Kinetics [19] are used to test the effectiveness of our 3M model.

The UCF101 dataset contains about 13000 action videos and each video contains about 180 frames on average. The number of action classes of the UCF101 dataset is 101. Three different splits are provided by the dataset organisers for splitting the UCF101 dataset. Each UCF101 split can divide the UCF101 dataset into about 9500 training videos and 3500 test videos. The evaluation rule for the UCF101 dataset is the average accuracy across three different splits. The HMDB51 dataset contains about 6800 action videos. The number of action classes of the HMDB51 dataset is 51. The HMDB51 dataset organisers also provide three different splits to divide the HMDB51 dataset into training data and test data. Each HMDB51 split can divide the HMDB1 dataset into about 3700 training videos and 3100 test videos. The evaluation rule for the HMDB51 dataset is the same as the UCF101 dataset. Kinetics is a large video dataset for action recognition, which contains about $300k$ videos. The training set contains more than $240k$ videos, validation set contains $20k$ videos and the test set contains $40k$ videos. All the videos are collected from the YouTube videos and each video is trimmed into one video clip. The evaluation rule for the Kinetics dataset also is the classification accuracy of action recognition.

Table 4.2: The comparison of the two-stream model, SST model, TST model and the proposed STSF model on the UCF101 and HMDB51 datasets.

Different streams	UCF101	HMDB51
Two-stream [26]	86.9%	58.0%
SST	90.0%	61.7%
TST	91.2%	62.3%
STSF	**92.2%**	**64.0%**

4.5.2 Ablation experiments on UCF101 and HMDB51

In this section, a set of ablation experiments are performed on UCF101 and HMDB51 to verify the effectiveness of our STSF model. The comparison models can be listed as follows:

- The benchmark two-stream model (named two-stream for short) [26] is the first comparison model.
- The second comparison model is the Spatial Saliency based Three-stream model (named SST for short), which comprises the spatial stream, temporal stream, and the spatial saliency stream. This model is used to verify the effectiveness of the proposed novel spatial saliency stream.
- The third comparison model is the Temporal Saliency based Three-stream model (named TST for short), which comprises the spatial stream, temporal stream, and the temporal saliency stream. This model is used to verify the effectiveness of the proposed novel temporal saliency stream.
- The last comparison model is our proposed STSF model, which comprises the spatial stream, temporal stream, spatial saliency stream, and the temporal saliency stream.

We illustrate the comparison of the recognition accuracy of the two-stream model, SST model, TST model and the proposed STSF model on UCF101 and HMDB51 in Table 4.2.

Table 4.2 shows that the proposed STSF model outperforms all other models with different stream (two-stream model, SST model and TST model) on both UCF101 and HMDB51, which proves that both the novel spatial saliency stream and the novel temporal saliency stream can improve accuracy. Specifically, the TST model improves 2.3% and 3.3% than the two-stream model in terms of classification accuracy on UCF101 and HMDB51, respectively, which proves the effectiveness of the proposed novel temporal saliency stream. This phenomenon is also verified in the comparison of the proposed STSF model with the SST model, which shows that the proposed STSF model improves 2.2% and 2.3% than the SST model on UCF101 and HMDB51, respectively. Further, the SST model improves 2.1% and 1.3% than the two-stream model on UCF101 and HMDB51, respectively. This proves the effectiveness of the proposed novel spatial saliency stream. This is also verified

by the fact that our proposed STSF model improves 1.0% and 1.7% than the TST model on UCF101 and HMDB51, respectively. Moreover, the proposed STSF model improves 5.7%, 2.3% and 1.4% on average than the two-stream model, the SST model and the TST model on UCF101 and HMDB51, respectively. This also can verify the effectiveness of the proposed STSF model.

In addition, Figure 4.6 shows the training loss of different models (two-stream model, SST model, TST model and the proposed STSF model) on UCF101. Figure 4.6 shows that the proposed STSF model achieves the fastest training convergence speed, then follows the SST model and TST model, the conventional two-stream model has the slowest training convergence speed. This phenomenon demonstrates that the proposed STSF model can help speed up the training convergence speed.

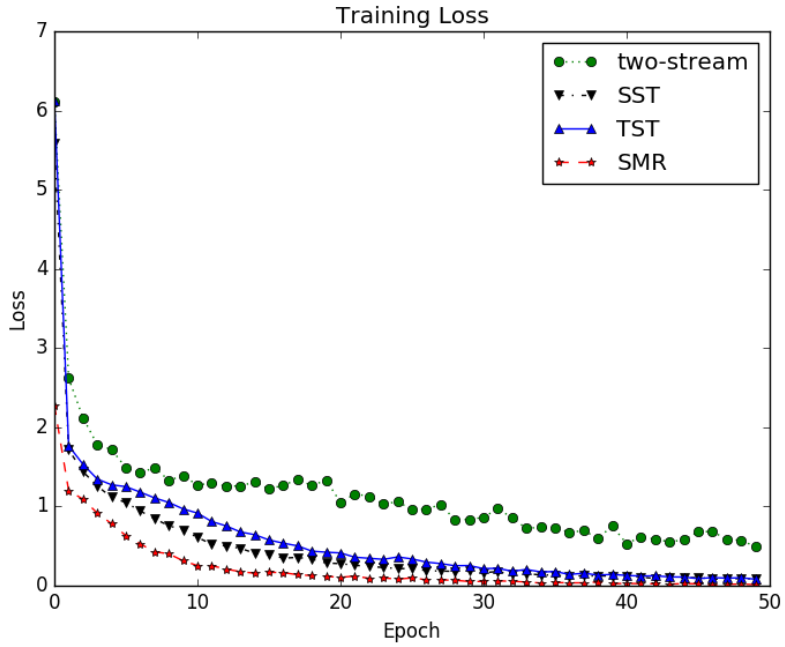

Figure 4.6: The training loss of different models on UCF101.

4.5.3 Compared with other state-of-the-art models

In this section, we compare our proposed STSF model and 3M model with several state-of-the-art methods on the UCF101, HMDB51, and Kinetics datasets to verify the effectiveness of our STSF model and 3M model. We use the pre-trained weight parameters on the Kinetics dataset to initialize our models, and then fine-tune them on the UCF101 and HMDB datasets. The referenced models (mainly including two-stream based models and 3D convolution based models) are listed as follows:

Table 4.3: The accuracies of different models on UCF101, HMDB51 and Kinetics.

Comparison methods	UCF101	HMDB51	Kinetics
Two-stream CNN [26]	86.9%	58.0%	61.0%
Two-stream with LSTM [35]	88.6%	–	–
STS [21]	90.1%	62.4%	–
STS-ALSTM [21]	92.7%	64.4%	–
ST-ResNet [9]	93.4%	66.4%	–
CTN [11]	92.5%	65.4%	–
C3D [29]	85.2%	–	56.1%
3D ResNet-34 [14]	89.6%	63.5%	58.0%
STSF	**93.2%**	**65.7%**	**65.5%**
3M	**94.7%**	**67.2%**	**68.7%**

- Two-stream CNN [26]: It is the first classic two-stream CNN model for action recognition.
- Two-stream with LSTM [35]: It utilized CNNs to generate appearance and motion features from RGB frames and optical flow frames, and then an LSTM is applied to model/capture the long-term dependency in videos.
- SpatioTemporal Saliency based multi-stream networks (STS) [21]: It utilized a spatiotemporal saliency stream to acquire the spatiotemporal object cue from videos to improve the performance.
- SpatioTemporal Saliency based multi-stream networks with attention-aware LSTM (STS-ALSTM) [21]: It utilized attention-aware LSTMs to capture the long-term dependency relationships of different streams (*i.e.* the appearance stream, motion stream and spatiotemporal saliency stream). An average fusion of the predictions of different LSTMs is adopted.
- Spatiotemporal Residual Networks (ST-ResNet) [9]: It injected residual connections between the appearance and motion stream to capture the spatiotemporal features from videos.
- Convolutional Two-stream Network (CTN) [11]: It fused the appearance cue with the motion cue at a convolution layer instead of at the softmax layer.
- 3D convolutional networks (C3D) [29]: It utilized a 3D filter kernel to perform 3D convolution operations along both spatial and temporal dimensions.
- 3D ResNet-34 [14]: It utilized a 34-layer 3D residual network to extract spatiotemporal features from videos.

The comparison results of different models for action recognition on UCF101, HMDB51 and Kinetics are reported in Table 4.3. It shows that our proposed 3M model achieves the best classification accuracy compared with all referenced models on all the datasets (UCF101, HMDB51 and Kinetics). This shows that the advantage of combining four-stream CNN feature extractor and multi-task learning based LSTM can significantly improve the classification accuracy. Compared with the STSF model, The 3M model improves 2.5%, 3.2% and 3.2% on UCF101, HMDB51, and Kinetics, respectively. This demonstrates that the multi-task learning based LSTM can further improve the accuracy based on the STSF model. In addition, we can find the proposed STSF model outperforms the two-stream CNN model, two-stream with LSTM model, STS model, STS-ALSTM model, CTN model, C3D model and the 3D ResNet-34 model on both UCF101 and HMDB51. While the accuracy of the STSF model is inferior to the ST-ResNet model on both UCF101 and HMDB51. The reason can be attributed to the efficient spatiotemporal information fusion, which can help learn more efficient spatiotemporal features for action recognition.

4.6 Summary

We propose a spatial and temporal saliency based four-stream network with multi-task learning for action recognition. Our results show that both the spatial and temporal salient information extracted from videos can benefit the enhancement of the accuracy of action recognition. The temporal long-term dependency relationships of different CNN feature inputs extracted from different stacked frames (*i.e.* video frames, optical flow frames, spatial saliency frames and temporal saliency frames) can be captured by LSTMs and shared by the final loss function, which can take full advantage of CNNs and LSTMs.

References

[1] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Action classification in soccer videos with long short-term memory recurrent neural networks. In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010.

[2] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011.

[3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Zhe Chen, Xin Wang, Zhen Sun, and Zhijian Wang. Motion saliency detection using a temporal fourier transform. *Optics & Laser Technology*, 80:1–15, 2016.

[6] Cheng Dai, Xingang Liu, and Jinfeng Lai. Human action recognition using two-stream attention based LSTM networks. *Applied Soft Computing*, 86:105820, 2020.

[7] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[8] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019.

[9] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016.

[10] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768–4777, 2017.

[11] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[12] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[13] Fang Guo, Wenguan Wang, Ziyi Shen, Jianbing Shen, Ling Shao, and Dacheng Tao. Motion-aware rapid video saliency detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[14] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 3154–3160, 2017.

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[17] Wanting Ji, Ruili Wang, and Junbo Ma. Dictionary-based active learning for sound event classification. *Multimedia Tools and Applications*, 78(3):3831–3842, 2019.

[18] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[20] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011.

[21] Zhenbing Liu, Zeya Li, Ruili Wang, Ming Zong, and Wanting Ji. Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. *Neural Computing and Applications*, 32(18):14593–14602, 2020.

[22] Chih-Yao Ma, Min-Hung Chen, Zsolt Kira, and Ghassan AlRegib. TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Processing: Image Communication*, 71:76–87, 2019.

[23] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local SVM approach. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 3, pages 32–36. IEEE, 2004.

[24] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[25] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1227–1236, 2019.

[26] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[27] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[28] Yan Tian, Xun Wang, Jiachen Wu, Ruili Wang, and Bailin Yang. Multi-scale hierarchical residual network for dense captioning. *Journal of Artificial Intelligence Research*, 64:181–196, 2019.

[29] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[30] Zhigang Tu, Wei Xie, Qianqing Qin, Ronald Poppe, Remco C Veltkamp, Baoxin Li, and Junsong Yuan. Multi-stream CNN: Learning representations based on human-related regions for action recognition. *Pattern Recognition*, 79:32–43, 2018.

[31] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2015.

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3169–3176. IEEE, 2011.

[33] Xuanhan Wang, Lianli Gao, Jingkuan Song, and Hengtao Shen. Beyond frame-level CNN: saliency-aware 3-D CNN with LSTM for video action recognition. *IEEE Signal Processing Letters*, 24(4):510–514, 2016.

[34] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision*, pages 305–321, 2018.

[35] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[36] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime TV-L1 optical flow. In *Joint Pattern Recognition Symposium*, pages 214–223. Springer, 2007.

[37] Jianming Zhang, Stan Sclaroff, Zhe Lin, Xiaohui Shen, Brian Price, and Radomir Mech. Minimum barrier salient object detection at 80 fps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1404–1412, 2015.

[38] Ming Zong, Ruili Wang, Xiubo Chen, Zhe Chen, and Yuanhao Gong. Motion saliency based multi-stream multiplier resnets for action recognition. *Image and Vision Computing*, (https://doi.org/10.1016/j.imavis.2021.104108), 2021.

Chapter 5

Summary

This chapter gives a conclusion of this thesis. We first gives a comparison of experimental results of all methods in Section 5.1. Then the contributions of this thesis are summarized in 5.2. Finally, the limitations and future work of this thesis are discussed in Section 5.3.

5.1 Comparison of experimental results of all methods

In this section, we compare all the methods presented in previous Chapters together. Table 5.1 illustrates the experimental results of all compared methods on UCF101 and HMDB51. *Note that Kinetics is not included. The reason is that Chapters 2 and 3 are my work in the early stage of my PhD study. The deep learning server used at that time was not powerful enough to process the Kinetics data set since the Kinetics data set is a large-scale video action data set containing about 300,000 videos. We thus conducted experiments on the general-scale benchmark data sets UCF101 and HMDB51. Chapter 4 is the work in the later stage of my PhD study. Since more powerful deep learning servers have been obtained that can be used to process the Kinetics data set, thus we conducted our experiments on both the general-scale benchmark data sets UCF101 and HMDB51, and the large-scale data set Kinetics.*

We can find that all our proposed methods (*i.e.* R-M3D, MSM-ResNets and 3M) achieve the state-of-the-art performance compared with other methods. Especially, the 3M model achieves the best performance compared with the R-M3D model and the MSM-ResNets model. The reason can be attributed to that this model obtains richer input information (including four types of input information, *i.e.* RGB information, optical flow information, temporal salient motion information and spatial salient information) compared with the

Table 5.1: The experimental results of all compared methods on UCF101 and HMDB51.

Methods	UCF101	HMDB51
iDT+FV [21]	85.9%	57.2%
Two-stream networks [16]	86.9%	58.0%
Two-stream + LSTM [22]	88.6%	–
LRCN [5]	82.9%	–
LTCN [20]	91.7%	64.8%
C3D (3 nets) [19]	85.2%	–
P3D ResNet [14]	88.6%	–
ST-ResNet [6]	93.4%	66.4%
3D ResNet (fine-tuned) [10]	89.3%	61.0%
SMN (34-layer) [7]	92.8%	65.2%
CTN [8]	92.5%	65.4%
STS-network [13]	90.1%	62.4%
STS-ALSTM [12]	92.7%	64.4%
3D ResNet-34 [9]	89.6%	63.5%
R-M3D (fine-tuned) (Chapter 2)	**93.2%**	**65.4%**
MSM-ResNets (Chapter 3)	**93.5%**	**66.7%**
3M (Chapter 4)	**94.7%**	**67.2%**

R-M3D model and the MSM-ResNets model (including three types of input information, *i.e.* RGB information, optical flow information and temporal salient motion information).

5.2 Summary of contributions

The main contributions of this thesis can be summarized as follows:

In Chapter 2, we explored the multi-cue based spatiotemporal features, and presented a multi-cue 3D convolutional neural network (M3D) for action recognition. It was demonstrated that directly performing 3D convolutions on multiple cues achieves higher accuracy than a single cue, *i.e.* the multi-cue based 3D convolution is effective. In addition, the deeper depth of the M3D network, the higher the accuracy of action recognition.

In Chapter 3, we explored the salient motion information for action recognition, and presented a motion saliency based multi-stream multiplier ResNets (MSM-ResNets). It was verified that the motion saliency information is beneficial for improving the accuracy of action recognition, and the way of injecting multiplicative connections from the motion saliency stream to the motion stream is effective for improving the accuracy.

In Chapter 4, we explored the salient spatiotemporal information over time evolution,

and presented a novel spatial and temporal saliency based four-stream network (3M) with multi-task learning for action recognition. Experimental results showed that both the spatial saliency information and motion saliency information is helpful for improving the classification accuracy. In addition, the complementary knowledge between different streams has been verified beneficial for the classification accuracy.

5.3 Limitations and future work

Lastly, we discuss the limitations and future work. For the limitations, the first one is that the proposed models cost more time to pre-process the data for obtaining the motion saliency maps. The second one is that the proposed models mainly focus on utilizing the RGB information, optical flow information, and motion saliency information. Some other useful information has not been considered such as the geometrical information between different samples.

For the future, the following directions are worth exploring and researching:

- **The choice of the motion cue input is important**. In our M3D and R-M3D models, we use two motion detection techniques (*i.e.* optical flow and motion saliency detection) to capture different motion information. Maybe there are other methods to provide more suitable motion information than them. For 3D convolutions performed on a multi-cue input, we develop the triple video representation. If more suitable multi-cue representation methods are developed [18], we may get even better results.
- **The effective fusion method between different information streams is important**. In our MSM-ResNets model, we consider injecting multiplicative connections between different streams (*i.e.* the appearance stream, the motion stream and the motion saliency stream). In our 3M model, we use different LSTMs to capture the temporal long-term dependency relationships of different streams, the fusion of different LSTMs is performed at the softmax layer. It is interesting to design more effective interactive fusion methods between different streams and explore the interactive relationships between different LSTMs [4].
- **The geometrical structure information is desirable to explore**. Deep learning has been successfully applied in video action recognition. However, most current works ignore utilizing the geometrical structure information existing among various video samples to improve the accuracy of action recognition. Manifold learning is a classic method for revealing and learning the intrinsic geometric structure of data samples, which assumes that the processed data is sampled from a high-dimensional Euclidean space, but intrinsically lying on a potential low-dimensional manifold [2, 15, 17]. Thus various manifold learning based subspace learning methods are designed for maintaining certain manifold structure information between different data samples

during space transformation [1, 11]. It would be interesting to integrate manifold learning into CNN architecture for action recognition [3].

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. *Advances in Neural Information Processing Systems*, 14:585–591, 2001.

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(Nov):2399–2434, 2006.

[3] Xin Chen, Jian Weng, Wei Lu, Jiaming Xu, and Jiasi Weng. Deep manifold learning combined with convolutional neural networks for action recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):3938–3952, 2017.

[4] Cheng Dai, Xingang Liu, and Jinfeng Lai. Human action recognition using two-stream attention based lstm networks. *Applied Soft Computing*, 86:105820, 2020.

[5] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[6] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Advances in Neural Information Processing Systems*, pages 3468–3476, 2016.

[7] Christoph Feichtenhofer, Axel Pinz, and Richard P Wildes. Spatiotemporal multiplier networks for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4768–4777, 2017.

[8] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3154–3160, 2017.

[10] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[11] Xiaofei He and Partha Niyogi. Locality preserving projections. *Advances in Neural Information Processing Systems*, 16:153–160, 2003.

[12] Zhenbing Liu, Zeya Li, Ruili Wang, Ming Zong, and Wanting Ji. Spatiotemporal saliency-based multi-stream networks with attention-aware LSTM for action recognition. *Neural Computing and Applications*, 32(18):14593–14602, 2020.

[13] Zhenbing Liu, Zeya Li, Ming Zong, Wanting Ji, Ruili Wang, and Yan Tian. Spa-

tiotemporal saliency based multi-stream networks for action recognition. In *Asian Conference on Pattern Recognition*, pages 74–84. Springer, 2019.

[14] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3D residual networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.

[15] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[16] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[17] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[18] Yan Tian, Yifan Cao, Jiachen Wu, Wei Hu, Chao Song, and Tao Yang. Multi-cue combination network for action-based video classification. *IET Computer Vision*, 13(6):542–548, 2019.

[19] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4489–4497, 2015.

[20] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018.

[21] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[22] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

Appendix A

Statement of Contribution

I confirm that the "Statement of Contribution to Doctoral Thesis Containing Publications (DRC16)", have been completed for each published article within the thesis, and are bound into the thesis and included in the electronic copy.

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

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:	MING ZONG
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 2

Please select one of the following three options:

(●) The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

Ming Zong, Ruili Wang, Zhe Chen, Maoli Wang, Xun Wang, and Johan Potgieter. "Multi-cue based 3D residual network for action recognition." Neural Computing and Applications. doi: https://doi.org/10.1007/s00521-020-05313-8, 2020.

( ) The manuscript is currently under review for publication – please indicate:

- The name of the journal:

- The percentage of the manuscript/published work that was contributed by the candidate:

- Describe the contribution that the candidate has made to the manuscript/published work:

( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Ming Zong 数字口名者：Ming Zong DN：cn=Ming Zong, o=Massey University, ou=School of Natural and Computational Sciences, email=m.zong@massey.ac.nz, c=NZ 日期：2021.02.02 11:26:56 +13'00'
Date:	02-2 月-2021
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2021.02.02 12:37:31 +13'00'
Date:	2-2 月-2021

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
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

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:	MING ZONG
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 3

Please select one of the following three options:

(●) The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

Ming Zong, Ruili Wang, Xiubo Chen, Zhe Chen and Yuanhao Gong. "Motion saliency based multi-stream multiplier ResNets for action recognition" Image and Vision Computing. doi: https://doi.org/10.1016/j.imavis.2021.104108, 2021.

( ) The manuscript is currently under review for publication – please indicate:

- The name of the journal:

- The percentage of the manuscript/published work that was contributed by the candidate:

- Describe the contribution that the candidate has made to the manuscript/published work:

( ) It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Ming Zong 数字口名者：Ming Zong DN：cn=Ming Zong, o=Massey University, ou=School of Natural and Computational Sciences, email=m.zong@massey.ac.nz, c=NZ 日期：2021.02.02 11:27:40 +13'00'
Date:	02-2 月-2021
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2021.02.02 12:37:59 +13'00'
Date:	2-2 月-2021

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
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

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:	MING ZONG
Name/title of Primary Supervisor:	Professor Ruili Wang
In which chapter is the manuscript /published work:	Chapter 4

Please select one of the following three options:

○ The manuscript/published work is published or in press

- Please provide the full reference of the Research Output:

◉ The manuscript is currently under review for publication – please indicate:

- The name of the journal:

IEEE Transactions on Circuits and Systems for Video Technology

- The percentage of the manuscript/published work that was contributed by the candidate: 75.00

- Describe the contribution that the candidate has made to the manuscript/published work:

For this manuscript, the contribution that the candidate has made include conceptualization, investigation, methodology, validation and draft writing.

○ It is intended that the manuscript will be published, but it has not yet been submitted to a journal

Candidate's Signature:	Ming Zong 数字口名者：Ming Zong DN：cn=Ming Zong, o=Massey University, ou=School of Natural and Computational Sciences, email=m.zong@massey.ac.nz, c=NZ 日期：2021.02.02 22:48:36 +13'00'
Date:	02-2 月-2021
Primary Supervisor's Signature:	Prof Ruili Wang Digitally signed by Prof Ruili Wang DN: cn=Prof Ruili Wang, c=NZ, o=Massey University , ou=School of Natural and Computational Sciences , email=ruili.wang@massey.ac.nz Date: 2021.02.02 23:31:17 +13'00'
Date:	2-Feb-2021

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.