Architectures d'apprentissage profond pour la reconnaissance d'actions humaines dans des séquences vidéo RGB-D monoculaires : application à la surveillance dans les transports publics

Huy-Hieu Pham

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Huy-Hieu PHAM

Le 19 septembre 2019

Architectures d'apprentissage profond pour la reconnaissance d'actions humaines dans des séquences vidéo RGB-D monoculaires. Application à la surveillance dans les transports publics.

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
IRIT : Institut de Recherche en Informatique de Toulouse

Thèse dirigée par
Denis KOUAMÉ et Louahdi KHOUDOUR

Jury

M. Stéphane CANU, Rapporteur
Mme Michèle GOUIFFÈS, Rapporteure
M. Yassine RUICHEK, Examinateur
M. Denis KOUAMÉ, Directeur de thèse
M. Louahdi KHOUDOUR, Co-directeur de thèse
M. Frédéric LERASLE, Président

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par : *l'Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier)*

Présentée et soutenue le *19 septembre 2019* par :

Huy Hieu PHAM

Architectures d'apprentissage profond pour la reconnaissance d'actions humaines dans des séquences vidéo RGB-D monoculaires. Application à la surveillance dans les transports publics.

Deep learning architectures for human action recognition from monocular RGB-D video sequences. Application to public transport monitoring.

JURY

Denis KOUAMÉ	Professeur Université Toulouse III – Paul Sabatier	Directeur de Thèse
Louahdi KHOUDOUR	Directeur de Recherche, Cerema	Co-Directeur de Thèse
Alain CROUZIL	Maître de Conférences, Université Toulouse III – Paul Sabatier	Encadrant
Stéphane CANU	Professeur, INSA de Rouen Normandie	Rapporteur
Michèle GOUIFFÈS	Maître de Conférences - HDR, Université Paris Sud	Rapporteure
Yassine RUICHEK	Professeur, Université de Technologie de Belfort-Montbéliard	Examinateur
Frédéric LERASLE	Professeur, Université Toulouse III – Paul Sabatier	Examinateur
Sergio A VELASTIN	Senior Research Scientist, Cortexica Vision Systems Ltd. Londres	Invité
Jean-Marc DAVIAU	Directeur Sécurité, Tisséo	Invité

École doctorale et spécialité :
MITT : Image, Information, Hypermédia
Unité de Recherche :
Institut de Recherche en Informatique de Toulouse (UMR 5505)
Directeur(s) de Thèse :
Denis KOUAMÉ et *Louahdi KHOUDOUR*
Rapporteurs :
Stéphane CANU et *Michèle GOUIFFÈS*

UNIVERSITÉ TOULOUSE III – PAUL SABATIER

Résumé

Doctorat

Architectures d'apprentissage profond pour la reconnaissance d'actions humaines dans des séquences vidéo RGB-D monoculaires. Application à la surveillance dans les transports publics.

par Huy Hieu PHAM

Cette thèse porte sur la reconnaissance d'actions humaines dans des séquences vidéo RGB-D monoculaires. La question principale est, à partir d'une vidéo ou d'une séquence d'images donnée, de savoir comment reconnaître des actions particulières qui se produisent. Cette tâche est importante et est un défi majeur à cause d'un certain nombre de verrous scientifiques induits par la variabilité des conditions d'acquisition, comme l'éclairage, la position, l'orientation et le champ de vue de la caméra, ainsi que par la variabilité de la réalisation des actions, notamment de leur vitesse d'exécution. Pour surmonter certaines de ces difficultés, dans un premier temps, nous examinons et évaluons les techniques les plus récentes pour la reconnaissance d'actions dans des vidéos. Nous proposons ensuite une nouvelle approche basée sur des réseaux de neurones profonds pour la reconnaissance d'actions humaines à partir de séquences de squelettes 3D. Deux questions clés ont été traitées. Tout d'abord, comment représenter la dynamique spatio-temporelle d'une séquence de squelettes pour exploiter efficacement la capacité d'apprentissage des représentations de haut niveau des réseaux de neurones convolutifs (CNNs ou ConvNets). Ensuite, comment concevoir une architecture de CNN capable d'apprendre des caractéristiques spatio-temporelles discriminantes à partir de la représentation proposée dans un objectif de classification. Pour cela, nous introduisons deux nouvelles représentations du mouvement 3D basées sur des squelettes, appelées SPMF (Skeleton Posture-Motion Feature) et Enhanced-SPMF, qui encodent les postures et les mouvements humains extraits des séquences de squelettes sous la forme d'images couleur RGB. Pour les tâches d'apprentissage et de classification, nous proposons différentes architectures de CNNs, qui sont basées sur les modèles Residual Network (ResNet), Inception-ResNet-v2, Densely Connected Convolutional Network (DenseNet) et Efficient Neural Architecture Search (ENAS), pour extraire des caractéristiques robustes de la représentation sous forme d'image que nous proposons et pour les classer. Les résultats expérimentaux sur des bases de données publiques (MSR Action3D, Kinect Activity Recognition Dataset, SBU Kinect Interaction, et NTU-RGB+D) montrent que notre approche surpasse les méthodes de l'état de l'art.

Nous proposons également une nouvelle technique pour l'estimation de postures humaines à partir d'une vidéo RGB. Pour cela, le modèle d'apprentissage profond appelé OpenPose est utilisé pour détecter les personnes et extraire leur posture en 2D. Un réseau de neurones profond est ensuite proposé pour apprendre la transformation permettant de reconstruire ces postures en trois dimensions. Les résultats expérimentaux sur la base de données Human3.6M montrent l'efficacité de la méthode proposée. Ces résultats ouvrent des perspectives pour une approche de la reconnaissance d'actions humaines à partir des séquences de squelettes 3D sans utiliser des capteurs de profondeur comme la Kinect.

Nous avons également constitué la base CEMEST, une nouvelle base de données RGB-D illustrant des comportements de passagers dans les transports publics. Elle

contient 203 vidéos de surveillance collectées dans une station du métro incluant des événements « normaux » et « anormaux ». Nous avons obtenu des résultats prometteurs sur cette base en utilisant des techniques d'augmentation de données et de transfert d'apprentissage. Notre approche permet de concevoir des applications basées sur des techniques de l'apprentissage profond pour renforcer la qualité des services de transport en commun.

Mots clés : reconnaissance d'actions humaines, réseaux de neurones convolutifs, recherche d'architecture neuronale, squelettes, capteur de profondeur.

UNIVERSITÉ TOULOUSE III – PAUL SABATIER

Abstract

Doctor of Philosophy

Deep learning architectures for human action recognition from monocular RGB-D video sequences. Application to public transport monitoring.

by Huy Hieu PHAM

This thesis is dealing with automatic recognition of human actions from monocular RGB-D video sequences. Our main goal is to recognize which human actions occur in unknown videos. This problem is a challenging task due to a number of obstacles caused by the variability of the acquisition conditions, including the lighting, the position, the orientation and the field of view of the camera, as well as the variability of actions which can be performed differently, notably in terms of speed. To tackle these problems, we first review and evaluate the most prominent state-of-the-art techniques to identify the current state of human action recognition in videos. We then propose a new approach for skeleton-based action recognition using Deep Neural Networks (DNNs). Two key questions have been addressed. First, how to efficiently represent the spatio-temporal patterns of skeletal data for fully exploiting the capacity in learning high-level representations of Deep Convolutional Neural Networks (D-CNNs). Second, how to design a powerful D-CNN architecture that is able to learn discriminative features from the proposed representation for classification task. As a result, we introduce two new 3D motion representations called SPMF (Skeleton Posture-Motion Feature) and Enhanced-SPMF that encode skeleton poses and their motions into color images. For learning and classification tasks, we design and train different D-CNN architectures based on the Residual Network (ResNet), Inception-ResNet-v2, Densely Connected Convolutional Network (DenseNet) and Efficient Neural Architecture Search (ENAS) to extract robust features from color–coded images and classify them. Experimental results on various public and challenging human action recognition datasets (MSR Action3D, Kinect Activity Recognition Dataset, SBU Kinect Interaction, and NTU-RGB+D) show that the proposed approach outperforms current state-of-the-art.

We also conducted research on the problem of 3D human pose estimation from monocular RGB video sequences and exploited the estimated 3D poses for recognition task. Specifically, a deep learning-based model called OpenPose is deployed to detect 2D human poses. A DNN is then proposed and trained for learning a 2D-to-3D mapping in order to map the detected 2D keypoints into 3D poses. Our experiments on the Human3.6M dataset verified the effectiveness of the proposed method. These obtained results allow opening a new research direction for human action recognition from 3D skeletal data, when the depth cameras are failing.

In addition, we collect and introduce in this thesis, CEMEST database, a new RGB-D dataset depicting passengers' behaviors in public transport. It consists of 203 untrimmed real-world surveillance videos of realistic "*normal*" and "*abnormal*" events. We achieve promising results on CEMEST with the support of data augmentation and transfer learning techniques. This enables the construction of real-world applications based on deep learning for enhancing public transportation management services.

Keywords: human action recognition, convolutional neural networks, neural architecture search, skeletal data, depth sensor.

Remerciements

Ce travail n'aurait pas été possible sans les apports et l'aide de nombreuses personnes dont mes superviseurs, mes amis, ma famille. Tout d'abord, et avant tout, je voudrais remercier Dr. Louahdi Khoudour, Directeur de Recherche au Cerema pour m'avoir offert l'occasion de travailler sur ce projet intéressant et pour sa direction tout au long de la période de trois ans de ma thèse. Un gros merci au Dr. Alain Crouzil, enseignant-chercheur à l'IRIT et Directeur du Département d'Informatique à l'Université Paul Sabatier et à Denis Kouamé, enseignant-chercheur à l'IRIT et professeur à l'Université Paul Sabatier, pour leur immense soutien et leurs conseils scientifiques. Un remerciement particulier au Professeur Sergio A Velastin à l'Université Carlos III de Madrid et au Dr. Pablo Zegers de l'Université des Andes au Chili, qui ont apporté leur soutien à mes activités de recherche et qui ont travaillé activement pour me fournir des outils pour poursuivre les objectifs visés.

Les trois années passées au Cerema et à l'Institut de Recherche en Informatique de Toulouse ont été une belle période de ma vie. Je tiens à remercier tous mes collègues : Christian Françoise, Jean-Paul Garrigos, Philippe Michou, Guillaume Saint Pierre, Emmanuel Delamarre, Anne Guerci, Céline Louise, Christophe Fautrat, Julien Philipot, André Nourisson, et Bérengère Mathieu. J'ai énormément appris d'eux sur la connaissance de la culture et de la langue française. Je voudrais aussi dire merci à Houssam Salmane, Chafik Bakey et Rachel Denot pour leur amitié et leur soutien pendant deux ans.

Je n'aurais pas réussi ce projet sans le soutien des amis proches en France : To Tat Dat, Nguyen Thi Thu Hang, Nguyen Thuy Nga, Nguyen Phuong, Nguyen, Hoang Phuong, Huynh Cong Bang et Prof. Nguyen Tien Zung. Je tiens à les remercier sincèrement. Par ailleurs, j'exprime toute ma reconnaissance à ma famille au Vietnam : Pham Van Si, Pham Duc Linh, Pham Duc An, Pham Trung Nghia, Pham Thi Huong, Pham Trong Hiep, Pham Hong Trang, Nguyen Dai Dang, Le Thi Thuy, Nguyen Quang Dong, Nguyen Hung Phat et beaucoup d'autres : je ne peux pas les énumérer tous. Je souhaite remercier spécialement ma femme, Nguyen Vu Thuong Huyen, pour son amour, sa patience, et son soutien. Merci à Minh Hien, qui m'a fourni une inspiration sans fin pour surmonter les défis dans ma vie.

Toulouse, France, le 24 juin 2019

Huy Hieu PHAM

Acknowledgements

This work would not have been possible without the advice and support of many people, from my supervisors, my friends and my family. First, and foremost, I would like to thank my advisors Dr. Louahdi Khoudour, Research Director at Cerema and Prof. Denis Kouamé at the IRIT - Paul Sabatier University for providing me the opportunity to work on this interesting project and for guiding me throughout three years of my PhD. My great thanks go to Dr. Alain Crouzil, teacher and researcher at the IRIT and Director of the Computer Science Department, Paul Sabatier University, for his immense support and his scientific advice. Special thanks to Prof. Sergio A Velastin, University Carlos III de Madrid and Dr. Pablo Zegers, Universidad de los Andes, who have been supportive of my career goals and who worked actively to provide me with the supervised academic time to pursue those goals.

Three years at the Cerema Institute and the Toulouse Institute of Computer Science (IRIT), are three beautiful years of my life. I would like to thank all my colleagues: Christian Françoise, Jean-Paul Garrigos, Philippe Michou, Guillaume Saint Pierre, Emmanuel Delamarre, Anne Guerci, Céline Louise, Christophe Fautrat, Julien Philipot, Andre Nourisson, and Bérengère Mathieu. I have learned from all of you about the knowledge of French language and culture. To Houssam Salmane, Bakey Chafik and Rachel Denot, I would like to thank you for your friendship and support that you've provided me over the past two years.

I would not have pursued this project without the support from all my close friends in France: To Tat Dat, Nguyen Thi Thu Hang, Nguyen Thuy Nga, Nguyen Phuong, Nguyen Hoang Phuong, Huynh Cong Bang et Prof. Nguyen Tien Zung. I would like to thank and acknowledge all of them. Above all, I would especially like to express my gratitude to my family in Vietnam: Pham Van Si, Pham Duc Linh, Pham Duc An, Pham Trung Nghia, Pham Thi Huong, Pham Trong Hiep, Pham Hong Trang, Nguyen Dai Dang, Le Thi Thuy, Nguyen Quang Dong, Nguyen Hung Phat and many other people I am not able to enumerate all. Finally, I wish to thank my loving and supportive wife, Nguyen Vu Thuong Huyen, for her patience and endless love. Thank you, Minh Hien, who provides unending inspiration for overcoming challenges in my life.

Toulouse, France, June 24, 2019

Huy Hieu PHAM

À mes parents . . .

Author's Publications

I hereby declare that the following publications have been produced during the course of this thesis:

Journal publications

[**J-3**] **Huy-Hieu Pham**, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Spatio-Temporal Image Representation of 3D Skeletal Movements for View-Invariant Action Recognition with Deep Convolutional Neural Networks*". Special Issue on Deep Learning-Based Image Sensors, Intelligent Sensors, Vol. 19 (8), 2019 **(Sensors 2019, Impact Factor: 3.014)** | .pdf

[**J-2**] **Huy-Hieu Pham**, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Learning to Recognize 3D Human Action from A New Skeleton-based Representation Using Deep Convolutional Neural Networks*". The IET Computer Vision Journal, Vol. 13 (319-328), 2019 **(IET 2018, Impact Factor: 1.132)** | .pdf

[**J-1**] **Huy-Hieu Pham**, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Exploiting Deep Residual Networks for Human Action Recognition from Skeletal Data*". The Computer Vision and Image Understanding Journal, Vol. 170 (51-66), 2018 **(CVIU 2018, Impact Factor: 2.776)** | .pdf

Peer-reviewed conference publications

[**C-3**] **Huy-Hieu Pham**, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*A Deep Learning Approach for Real-Time 3D Human Action Recognition from Skeletal Data*". The 16*th* International Conference on Image Analysis and Recognition, 27-30 September, 2019, Waterloo, Canada (**ICIAR 2019**) | .pdf

[**C-2**] **Huy-Hieu Pham**, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Skeleton to Color Map: A Novel Representation for 3D Action Recognition with Inception Residual Networks*". The 25*th* IEEE International Conference on Image Processing, 7-10 October, 2018, Athens, Greece (**ICIP 2018**) | .pdf

[**C-1**] **Huy-Hieu Pham**, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks*". The 8*th* International Conference on Pattern Recognition Systems, 11-13 July, 2017, Madrid, Spain (**ICPRS 2017**) | .pdf

Preprints

[**P-2**] **Huy-Hieu Pham**, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*A Unified Deep Framework for Joint 3D Pose Estimation and Action Recognition from a Single RGB Camera*" | .pdf

[**P-1**] **Huy-Hieu Pham**, Houssam Salmane, Louahdi Khoudour, Alain Crouzil, Pablo Zegers, Sergio A Velastin. "*Video-based human action recognition using deep learning: a review*" – Institutional Repository of University Carlos III (e-Archivo) | .pdf

Click here to open my updated Google Scholar.

Presentations

AUG. 2019 Conference talk, "*A Deep Learning Approach for Real-Time 3D Human Action Recognition from Skeletal Data*". The 16*th* International Conference on Image Analysis and Recognition (**ICIAR 2019**), Waterloo, Canada.

DEC. 2018 Invited speaker, "*Applied Machine Learning Days*" at the French Institute of Science and Technology for Transport, Development and Networks (IFSTTAR), Paris, France.

OCT. 2018 Conference talk, "*Skeleton to Color Map: A Novel Representation for 3D Action Recognition with Inception Residual Networks*". The 25th IEEE International Conference on Image Processing (**ICIP 2018**), Athens, Greece.

NOV. 2017 Invited speaker, "*An Introduction to Deep Learning for Image and Video Interpretation*" at the University Carlos III of Madrid (UC3M), Madrid, Spain.

JUL. 2017 Conference talk, "*Learning and Recognizing Human Action from Skeleton Movement with Deep Residual Neural Networks*". The 8th International Conference of Pattern Recognition Systems (**ICPRS 2017**), Madrid, Spain.

Contents

List of Figures

List of Tables

List of Abbreviations

ATN	**A**ttention **T**emporal **N**etworks
ADI	**A**verage **D**epth **I**mage
CNN	**C**onvolutional **N**eural **N**etwork
CRF	**C**onditional **R**andom **F**ield
cGAN	**C**onditional **G**enerative **A**dversarial **N**etwork
DCGAN	**D**eep **C**onvolutional **G**enerative **A**dversarial **N**etwork
DDI	**D**epth **D**ifference **I**mage
DBNs	**D**eep **B**elief **N**etworks
DDI	**D**epth **D**ifference **I**mage
DCGAN	**D**eep **C**onvolutional **G**enerative **A**dversarial Network
ENAS	**E**fficient **N**eural **A**rchitecture **S**earch
FFT	**F**ast **F**ourier **T**ransform
FTP	**F**ourier **T**emporal **P**yramid
GANs	**G**enerative **A**dversarial **N**etworks
HMM	**H**idden **M**arkov **M**odel
ICT	**I**nformation and **C**ommunication **T**echnologies
IBS	**I**nternet **B**roadcasting **S**ervices
KNN	**K**-**N**earest-**N**eighbor
KARD	**K**inect **A**ctivity **R**ecognition **D**ataset
LDA	**L**atent **D**irichlet **A**llocation
MLP	**M**ulti**L**ayer **P**erceptron
MHI	**M**otion **H**istory **I**mage
NAS	**N**eural **A**rchitecture **S**earch
OAR	**O**nline **A**ction **R**ecognition
RNN	**R**ecurrent **N**eural **N**etwork
RNN-LSTM	**R**ecurrent **N**eural **N**etwork with **L**ong **S**hort-**T**erm **M**emory units
ReLUs	**R**ectified **L**inear **U**nits
RBMs	**R**estricted **B**oltzmann **M**achines
SPMF	**S**keleton **P**ose-**M**otion **F**eature
SVM	**S**upport **V**ector **M**achine
SMS	**S**ocial **M**edia **S**ervices
SDA	**S**tacked **D**enoising **A**utoencoders
TCN	**T**emporal **C**onvolutional **N**etwork

Chapter 1

Introduction

Contents

Chapter overview: In this chapter, we introduce the topic of this PhD thesis, human action recognition from monocular RGB-D video sequences (Section 1.1). Then, we present the important role of the recognition of human actions in building various real-life applications (Section 1.2). In particular, we discuss the current trends, new challenges and interests related to this research topic (Section 1.3). After that, the problem statement and its scope as well as our main contributions are presented (Section 1.4 & Section 1.5). Finally, we close the chapter by the thesis structure (Section 1.6).

1.1 Human action recognition in videos

Human action recognition in videos plays a prominent role in many different intelligent video analysis systems. The main goal of video-based human action recognition is to automatically analyze ongoing video streams provided by unknown cameras to detect and determine which human actions occur in these videos. An action can be defined as a spatio-temporal sequence of human body movements that has starting and ending temporal points. According to computer vision field, given an input video that contains one or several actions, human action recognition attempts to label each action with its corresponding name. In other words, this is an automatic labelling process in which each action occurring in video is described by a suitable verb or noun.

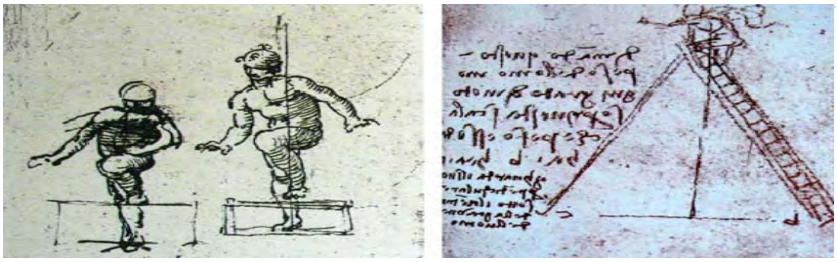


FIGURE 1.1: Early studies on human action recognition were motivated by human representations in arts. For example, this picture describes a man going upstairs or up a ladder that was drawn by Leonardo da Vinci (1452–1519) in the 15th century (Ivan, 2012).

Early studies on human action recognition were motivated by human representations in arts, biomechanics, motion perception (Ivan, 2012) and then expanded into so many modern applications in computer vision as nowadays (Ranasinghe, Al Machot, and Mayr, 2016). For the time being, human action recognition is one of the key techniques in building many intelligent systems involving video surveillance, human-machine interaction, self-driving cars, robot vision and so on. Although significant progress has been made over the past two decades, developing a fast and accurate action recognition framework is still a challenging task due to many obstacles such as viewpoint, occlusion or lighting conditions (Poppe, 2010). To deal with the challenges, traditional computer vision approaches consider an action recognition system as a hierarchical process, where the lower levels are on human detection and segmentation. The aim of these levels is to identify the regions of interest (ROIs) that correspond to static or moving humans in videos. The visual information of actions is extracted at the next level and represented by motion features or descriptors (Lowe, 2004; Dalal and Triggs, 2005; Laptev et al., 2008; Klaser, Marszałek, and Schmid, 2008). This high-level information is then used to train a classifier for recognizing actions. FIGURE 1.2 shows the pipeline of such a typical action recognition system.

FIGURE 1.2: Overview of a typical video-based human action recognition system. The regions of interest (ROIs) corresponding to human motions are first identified. Their spatial-temporal features or descriptors, *e.g.* SIFT (Lowe, 2004), HOG/HOF (Dalal and Triggs, 2005; Laptev et al., 2008), HOG-3D (Klaser, Marszałek, and Schmid, 2008), are then computed and fed into a classifier for recognizing actions.

Like many other topics in computer vision, human action recognition in videos is a fast growing field where new needs and challenges appear over a very short period of time. Some of them include data-independent solutions, real-time demand and the capacity of recognizing human actions in constrained and unconstrained videos. Meanwhile, the traditional approaches revealed some limitations that are difficult to overcome such as data-dependence and requirement of a lot of feature engineering. In this dissertation, we address the problem of recognizing human actions in RGB-D videos. We aim to fully exploit the capacity of state-of-the-art Deep Convolutional Neural Networks (D-CNNs) in learning high-level representation of human motions from RGB-D video sequences for action recognition task. Our goal is to propose new compact representations from RGB-D data and design high-performance deep learning models for 3D human action recognition. The proposed method should be able to automatically learn spatio-temporal motion features from training videos and recognize many different kinds of actions in unseen and realistic video settings with a high accuracy.

1.2 Motivation

The development and rapid expansion of information and communication technologies (ICT), especially Internet Broadcasting Services (IBS) and Social Media Services (SMS) has led to exponential boom of video data. Multimedia data, comprised of audio, video, and still images are now easily captured using so many different hardware platforms such as digital computers, notebooks, smartphones, and digital photo cameras. As a result, enormous amount

of data can be shared at unprecedented levels, in which many of the data are relevant to human actions. Therefore, there is a large demand for the computer vision community to recognize what humans do in videos in an automated fashion, playing an important role in our everyday life.

In recent years, human action recognition continues to be an increasingly active research in computer vision due to the interest in the development of many intelligent systems. The main goal of this area is to recognize what humans do in untrimmed videos. Many applications of video-based action recognition have been developed such as intelligent surveillance systems (Wei Niu et al., 2004; Valera and Velastin, 2005; Weiyao Lin et al., 2008), human-computer interfaces (Pickering, Burnham, and Richardson, 2007; Sonwalkar et al., 2015), health care (Zouba et al., 2009), and virtual reality (Maqueda et al., 2015). FIGURE 1.3 shows some specific applications in which the recognition of human actions plays a key role.

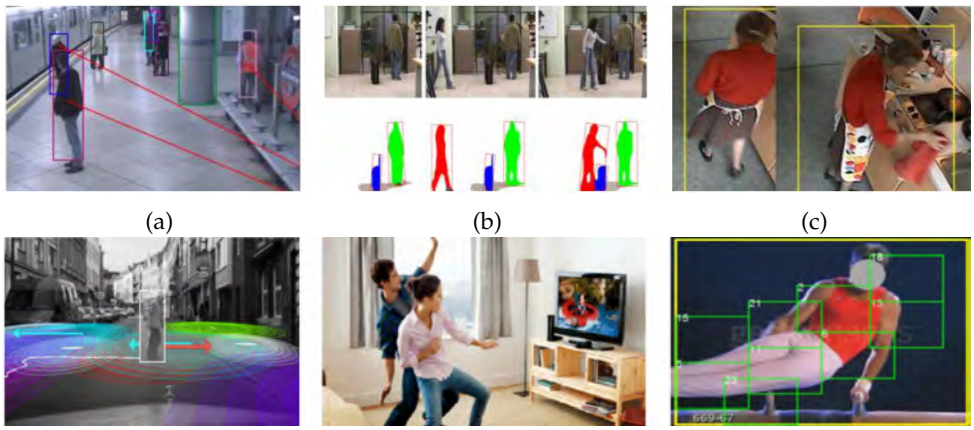


FIGURE 1.3: Some important applications of video-based human action analysis: **(a)** recognizing and tracking human actions in intelligent transport systems (Ryoo and Aggarwal, 2008); **(b)** stealing detection (Ryoo and Aggarwal, 2007); **(c)** remote monitoring service for elderly persons based on fall detection (Zouba et al., 2009); **(d)** pedestrian path prediction in self-driving cars (Kooij, Schneider, and Gavrila, 2014); **(e)** action recognition based on depth sensors in the entertainment industry (Zhang, 2012); **(f)** action localization and analysis in realistic sports videos (Tian, Rahul, and Shah, 2013).

Besides the specific applications mentioned above, there are many other new applications based on human action recognition techniques, which occur over a short period of time. As shown in the following FIGURE 1.4, we can now use a drone that integrates an action recognition algorithm to detect violent actions in crowds (Singh, Patil, and Omkar, 2018) or retail companies are now able to replace their traditional checkout systems by new autonomous systems. It is clear that there is a large demand in building real-time and accurate video-based human action recognition systems, which plays a huge impact for a safe and comfortable life.

(a) (b)

FIGURE 1.4: **(a)** Recognizing customer behaviors and their actions at an autonomous checkout system. The figure was taken from Standard Cognition (`https://standard.ai/`); **(b)** Detecting violent and abnormal crowd activity using drone (Singh, Patil, and Omkar, 2018).

1.3 Research challenges

A rapid increase in the number of researchers and techniques focusing on vision-based human action recognition has significantly advanced this field. However, this area is still challenging due to many obstacles such as large intra-class variations (see FIGURE 1.5), fuzzy boundaries between classes, viewpoint variations, occlusions, appearance changes, camera motion, cluttered background, lighting conditions, recording settings and so on (Poppe, 2010). Therefore, one of the main challenges with human action recognition in videos is to find a robust representation of actions, which is discriminative enough so that action recognizers or learning models can classify various different action classes.

FIGURE 1.5: The large intra-class variation and the variety in camera views are two enormous challenges in recognizing human actions. Sample frames are taken from the NTU-RGB+D dataset (Shahroudy et al., 2016).

In addition to the traditional challenges mentioned above, researchers and engineers also face new challenges. For instance, after the success of the action recognition systems on benchmarks produced "*in the lab*", more complex benchmarks have been released such as Hollywood-I (Laptev et al., 2008), Hollywood-II (Marszalek, Laptev, and Schmid, 2009), HMDB-51 (Kuehne et al., 2011), UCF-50 (Reddy and Shah, 2013), UCF-101 (Soomro, Zamir, and Shah, 2012), YouTube (Liu, Jiebo Luo, and Shah, 2009), Sports-1M (Karpathy et al., 2014),

ActivityNet (Heilbron et al., 2015), NTU-RGB+D (Shahroudy et al., 2016) and YouTube 8M (Abu-El-Haija et al., 2016). The complexity of large-scale datasets leads to the new problem of recognizing "*complex actions and behaviors in untrimmed videos*". We show in this thesis that experimental results on realistic human action datasets have so far given limited results specially when dealing with a large and varied range of actions. Additionally, how to build "*real-time action recognition systems*" is also a big question, in particular in the cases where these systems are built based on time-consuming models like machine learning and deep learning algorithms. The field of video-based human action recognition requires a combination of several disciplines including psychology and ontology (Rodríguez et al., 2014), and this is one of difficulties.

1.4 Problem statement and scope of study

This dissertation focuses on the problem of 3D human action recognition in realistic video material, *e.g.* surveillance videos. In our work, we consider "*an action*" as characterized by simple motion patterns, typically executed by a single person. An action can be defined as a spatio-temporal sequence of human body movements that consists of several action primitives ordered in time, starting and ending with temporal points. Meanwhile, "*an activity*" is more complex and involves coordinated actions among a small number of humans. Given a video that contains one or several actions, our goal is to predict action labels that occur in the video. This is also the main goal of the action recognition problem.

Over the last years, many techniques have been proposed for this task. In particular, deep learning based approaches have shown impressive performance and big potential in analyzing and recognizing human actions in videos. Many different deep architectures have been proposed for action recognition and advanced the state-of-the-art in this field. In this thesis, our first goal is to review and compare the existing deep learning-based methods for human action recognition in videos in order to identify which architectures and video representations are the best suitable. We then indicate the limitations of the existing techniques and propose new approaches for recognizing actions. More specifically, the following objectives are included in this thesis:

• We aim to identify the current state of deep learning-based approaches for human action recognition in videos, providing the most commonly used deep architectures for learning human motion features and show how they could be applied to address challenges in action recognition as well as discuss the advantages and limitations of each approach.

• We investigate and propose new 3D motion representations and deep learning frameworks for video-based human action recognition, both from RGB-D and RGB video sequences. The proposed approach should be able to recognize human actions from realistic videos and ensure a high accuracy.

• We aim to collect and introduce a real-world RGB-D dataset for evaluating our proposed action recognition and behavior analysis approach in order to improve security and safety in public transport.

1.5 Main contributions

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

• **First**, we introduce the problem of recognizing human actions in videos and provide an extensive review on deep learning-based action recognition methods. Over more than 250 related publications from top-tier conferences and journals, we identify the current state and the next questions of this field (Chapter 3).

• **Second**, we present new skeleton-based representations and deep learning frameworks for

3D action recognition from skeletal data provided by depth sensors. The proposed skeleton-based representations, which we refer to as SPMF and Enhanced-SPMF, are able to capture the spatio-temporal dynamics of skeleton movements and transforms them into a 2D structure as a single RGB image that suits the problem of learning representation with Deep Convolutional Neural Networks (D-CNNs). The proposed learning framework directly learns an end-to-end mapping between skeleton sequences and their action labels via the SPMF or Enhanced-SPMF. Experimental results on four highly competitive benchmark datasets demonstrate that the proposed method obtains a significantly improvement over the existing state-of-the-art approaches. In particular, our computational efficiency evaluations show that this method is able to achieve high-level of performance (Chapter 4).

• **Third**, this thesis introduces CEMEST database, our new RGB-D dataset depicting passenger behaviors in public transport. It consists of 203 untrimmed real-world surveillance videos of realistic « *normal* » and « *anomalous* » events. We achieve promising results in real-world conditions of this dataset thanks to the support of data augmentation and transfer learning techniques. This enables the construction of real-world applications based on deep learning for enhancing monitoring and security in public transport (Chapter 4).

• **Finally**, we propose and introduce a unified deep learning framework for 3D pose estimation and action recognition from RGB images. This framework uses a 2D skeleton detector called OpenPose to produce 2D human poses from RGB images. Then, this framework is integrated a deep neural network in order to learn a "*2D-to-3D mapping*" between 2D poses and 3D poses. The obtained 3D human poses are then used for the recognition task (Chapter 5). We show that the proposed deep learning framework is able to solve both two tasks in an effective manner.

1.6 Structure of the thesis

The thesis is structured as follows: Chapter 2 is an introduction to deep learning. We present background knowledge around machine learning and deep learning as well as the most important deep learning models. Chapter 3 provides a review of various state-of-the-art deep learning-based techniques for human action recognition in RGB-D videos. A detailed description of our proposed approaches for skeleton-based action recognition using depth sensors is given in Chapter 4. Chapter 5 describes the proposed deep learning approach for 3D skeleton reconstruction and action recognition from RGB cameras. Finally, Chapter 6 summarizes and discusses the key findings of this thesis. We then outline the limitations of our approaches and end this thesis by providing some promising directions for future work.

Chapter 2

Overview of Deep Learning

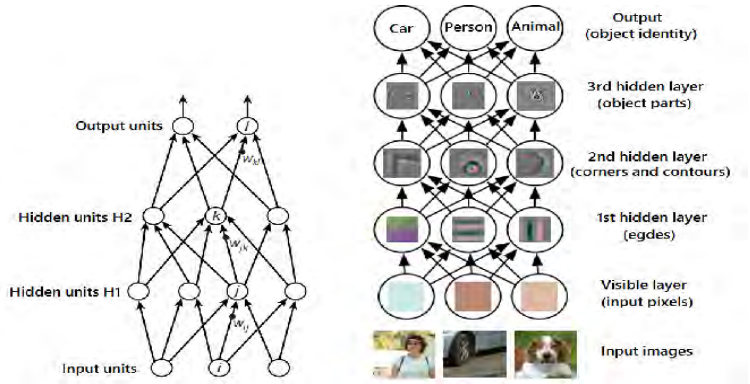
Contents

Chapter overview: In this chapter, we present the basic concepts of deep learning and review some key deep learning algorithms such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTMs), Deep Belief Networks (DBNs), Stacked Denoising Autoencoders (SDAs), and Generative Adversarial Networks (GANs). The key idea and mathematical model behind each algorithm will be introduced. This helps us to identify a suitable deep learning model for addressing the problem of human action recognition in videos.

2.1 Deep Learning: A summary

For the sake of completeness, we present in this chapter an overview of deep learning algorithms – the key technique that will be used in this thesis for addressing the problem of human action recognition in videos. Before that, we briefly summarize the concept of machine learning. Machine learning is the branch of algorithms that allows computers to automatically learn from data. They can be used for identifying objects in images, detecting spam emails, understanding text, finding genes associated with a particular disease and numerous other real-life applications. The primary goal of machine learning is to develop general-purpose algorithms, which are able to make accurate predictions in many different tasks. Mathematically, machine learning algorithms try to match the density function that produced the data. For example in classification problems, we try to identify a set of categories $\mathcal{C}$ from a space of all possible examples $\mathcal{X}$. Given any set of labeled examples $(\mathbf{x}_1, \mathbf{c}_1), ..., (\mathbf{x}_m, \mathbf{c}_m)$, where $\mathbf{x}_i \in \mathcal{X}$ and $\mathbf{c}_i \in \mathcal{C}$, the goal of machine leaning is to find a mapping function $\mathcal{F}(\cdot)$ that satisfies $\mathbf{c}_i = \mathcal{F}(\mathbf{x}_i)$ for all i. Machine learning methods are typically classified into four categories including supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning.

Deep learning is a class of techniques in machine learning. It became a major breakthrough in computer vision after the AlexNet (Krizhevsky, Sutskever, and Hinton, 2012a) achieved a record performance on ImageNet (Rahmani and Mian, 2016). Generally speaking, deep learning methods are machine learning methods, used to model high-level abstractions in data through the use of artificial neural networks, which are composed of multiple



(a) (b)

FIGURE 2.1: **(a)** Illustration of a multilayer network model (LeCun, Bengio, and Hinton, 2015). This model allows a computer to automatically determine the representations needed for prediction tasks. The first layer, namely *"visible layer"*, contains natural data in their raw form. Starting from the visible layer, a series of hidden layers is built through extracting increasingly abstract features from lower levels. More abstract concepts are learned from the lower levels. The highest layer contains useful information for predicting the content of input data. **(b)** An example of a deep learning model for classification task (Zeiler and Fergus, 2014; Goodfellow, Bengio, and Courville, 2016). Given some pictures, the first layer includes an array of pixel values. The first hidden layer represents the presence of edges. The second hidden layer identifies corners and contours from edges provided by the first layer. By connecting corners and contours, the third layer can determine specific objects.

nonlinear transformations. FIGURE 2.1 illustrates a multilayer network and the construction process of the higher layers from the first layer. Various deep learning architectures have been proposed over the years (see TABLE 2.1). Many of them have been shown to produce state-of-the-art performances on many visual recognition tasks, not least within human action recognition. In the next sections, we describe the most important deep learning architectures for video-based human action recognition including Convolutional Neural Networks (CNNs – Fukushima, 1980; Rumelhart, Hinton, and Williams, 1986; LeCun et al., 1989a; Krizhevsky, Sutskever, and Hinton, 2012a), Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTMs – Hochreiter and Schmidhuber, 1997), Deep Belief Networks (DBNs – Hinton, Osindero, and Teh, 2006), Stacked Denoising Autoencoders (SDAs – Vincent et al., 2008), and Generative Adversarial Networks (GANs – Goodfellow et al., 2014).

2.2 Convolutional Neural Networks (CNNs)

After the success of the AlexNet (Krizhevsky, Sutskever, and Hinton, 2012a) in 2012, Convolutional Neural Networks (CNNs) became one of the most important deep learning models and play a dominant role for solving visual recognition tasks. The CNN models are able to learn visual representations on the raw data without any hand-crafted feature extraction. The idea of CNNs was firstly presented in 1980 by Fukushima (Fukushima, 1980), in which CNNs are inspired by the structure of the visual nervous system (Hubel and Wiesel, 1962). Different CNN architectures continued to be proposed and developed, *e.g.* by Rumelhart, Hinton, and Williams, 1986, LeCun et al., 1989a and Krizhevsky, Sutskever, and Hinton, 2012a. Three key ideas behind a CNN architecture include *"local connections"*, *"shared weights"*, and *"pooling"* – which are described below.

Local connections: In regular neural networks, each hidden layer consists of a set of neurons where each neuron is fully connected to all neurons in the previous layer. This model

TABLE 2.1: Some popular deep learning architectures for visual recognition tasks, including human action recognition.

Architecture	Main articles
CNNs	Fukushima, 1980;
	Rumelhart, Hinton, and Williams, 1986;
	LeCun et al., 1989a;
	Krizhevsky, Sutskever, and Hinton, 2012a;
	Szegedy et al., 2015a;
	Simonyan and Zisserman, 2014b;
	Kaiming et al., 2016.
RNN-LSTMs	Hochreiter and Schmidhuber, 1997.
DBNs	Hinton, Osindero, and Teh, 2006;
	Salakhutdinov and Hinton, 2009.
Sparse Coding	Olshausen and Field, 1996;
	Lee et al., 2006.
SDAs	Vincent et al., 2008.
GANs	Goodfellow et al., 2014.

does not work efficiently when the input vector has a high dimension. To make this more efficient, one possibility is to reduce the number of connections between the first hidden layer to the input or each hidden layers to each other. Given an image as an input vector, every input pixel is not connected to every neuron in the first hidden layer. Instead, the neurons in the first hidden layer are connected to localized regions of the input image. This sub-region is called the *"local receptive field"*. For each local receptive field, we can identify a neuron in the first hidden layer.

Shared weights: For standard neural networks such as multilayer perceptrons (MLP – Ruck, Rogers, and Kabrisky, 1990), all neurons of the first layer are computed by the dot product function of input vector $\mathbf{x}$ and its weights $\mathbf{w}$, where many different $\mathbf{w}_i$ values are used. A technique called *"weight sharing"* is used to reduce the number of parameters $\mathbf{w}_i$ in a CNN. Specifically, some of parameters are constrained to be equal to each other. Mathematically, the weight sharing technique can be performed by using a convolution operator, in which the filters are applied to many local receptive fields in the input image. A *"feature map"* is generated by sliding a filter over the input matrix and computing the dot product.

Pooling: *"Pooling"* is a non-linear down-sampling process. Its main goal is to reduce the dimensionality of the input features while retaining the most important information in feature maps. This process allows to reduce the computational cost. At the same time it provides invariance to small transformations. Pooling is performed by using a pooling function to replace the output of the network at a certain location with summary statistics of the nearby outputs.

These concepts above can now be put together to form a complete CNN architecture that consists of a series of stages, as shown in FIGURE 2.2. In a CNN, the convolution layer plays the role of a local feature extractor while the pooling layer merges semantically similar features into one and reduce their dimensions. The last layer is a fully connected layer working as a classifier. Rectified Linear Units (ReLUs – Nair and Hinton, 2010) are commonly used as activation functions to train CNNs and Dropout (Srivastava et al., 2014) is used to prevent overfitting. Further details on the development of the CNNs can be found for example on state-of-the-art CNN architectures such as GoogLeNet (Szegedy et al., 2015a), VGG-Net (Simonyan and Zisserman, 2014b), ResNet (Kaiming et al., 2016), Inception-v3 (Szegedy et al., 2016), etc.

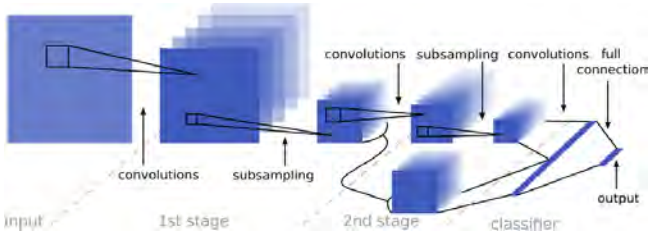


FIGURE 2.2: Illustration of the typical block diagram of a CNN (Sermanet and LeCun, 2011). The input image is fed through two stage of convolutions and subsampling for learning rich features from raw data. A linear classifier is used for classification task.

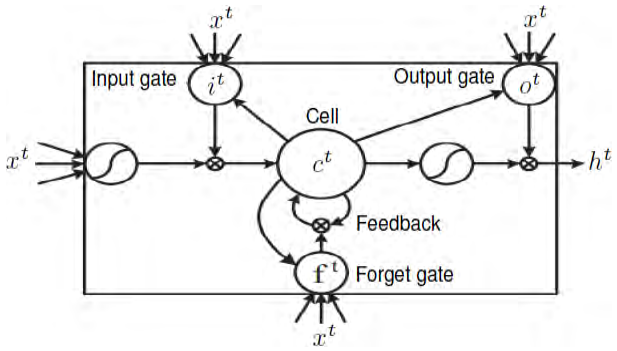
2.3 Recurrent Neural Networks with Long Short-Term Memory units (RNN-LSTM)

Recurrent Neural Networks (RNNs) are commonly used to model the temporal dynamics of human actions in videos because this architecture allows to store and access the long range contextual information of a temporal sequence. The main difference between RNNs and MLPs (Ruck, Rogers, and Kabrisky, 1990) is the presence of cyclical connections in the RNNs. With these connections, an RNN can learn to map from the entire history of previous inputs to each output (Graves, 2008). However, they are very difficult to train due to the vanishing gradient problem (Bengio, Simard, and Frasconi, 1994). To this end, Long Short-Term Memory units (LSTM – Hochreiter and Schmidhuber, 1997) have been proposed. FIGURE 2.3 describes the LSTM structure and its information flow. RNNs are not only able to make use of previous context in data sequences but also to exploit future context as well. Bidirectional RNN-LSTMs (Schuster and Paliwal, 1997) have been proposed to do this by processing and storing both past and future context of data with two separate hidden layers. All the information is then fed to the same output layer. By replacing the nonlinear units in the Bidirectional RNNs architecture by LSTM cells, we can obtain RNN-LSTMs as shown in FIGURE 2.4. In the next chapter, we will go into more detail on how RNN-LSTMs can be applied to model spatial and temporal dynamics of human actions.

2.4 Deep Belief Networks (DBNs)

Deep Belief Networks (DBNs – Hinton, Osindero, and Teh, 2006) have been used successfully for many recognition tasks such as handwritten digits recognition (Hinton, 2002), object recognition (Nair and Hinton, 2009), modeling human motion (Taylor, Hinton, and Roweis, 2007), etc. The DBNs are probabilistic generative models that are constructed by stacking several Restricted Boltzmann Machines (RBMs – Hinton, Sejnowski, and Ackley, 1984). The RBMs are shallow networks containing two layers: one layer of *"visible"* units that represents the input data and one layer of *"hidden"* units that learns to represent features. As shown in FIGURE 2.5a, in an RBM architecture, all visible units of the visible layer are connected to all the hidden units of the hidden layer and there are no connections between two units of the same layer. The standard type of RBM has binary-valued hidden and visible units, meaning each unit can only be in one of two states, *"0"* or *"1"*. The probability of setting a unit to *"1"* is a sigmoid function of its bias, weights, and the state of other units. Given a binary RBM with m visible units $\mathcal{V} = \{v_i\}, i \in (1, ..., m)$ and n hidden units $\mathcal{H} = \{h_j\}, j \in (1, ..., n)$, where v_i and h_j are the binary states of visible unit i and hidden unit j or $(v_i, h_j) \in (0, 1)^{m+n}$. The joint probability distribution for visible and hidden units (Hinton, 2010) is defined as

$$P(v_i, h_j) = \frac{1}{Z} e^{-E(v_i, h_j)}, \tag{2.1}$$



$$i^t = \sigma(W_{xi}x^t + W_{hi}h^{t-1} + W_{ci}c^{t-1} + b_i)$$
$$f^t = \sigma(W_{xf}x^t + W_{hf}h^{t-1} + W_{cf}c^{t-1} + b_f)$$
$$c^t = f^t c^{t-1} + i^t tanh(W_{xc}x^t + W_{hc}h^{t-1} + b_c)$$
$$o^t = \sigma(W_{xo}x^t + W_{ho}h^{t-1} + W_{co}c^t + b_o)$$
$$h^t = o^t tanh(c^t)$$

FIGURE 2.3: Diagram of an LSTM unit (Graves, 2008). A typical LSTM unit contains an input gate i^t, a forget gate f^t, an output gate o^t, an output state h^t and a memory cell state c^t. The information flow is described by the above equations where σ is the sigmoid activation; x^t is the input to the network at time t; the matrices W are the connection weights between units.

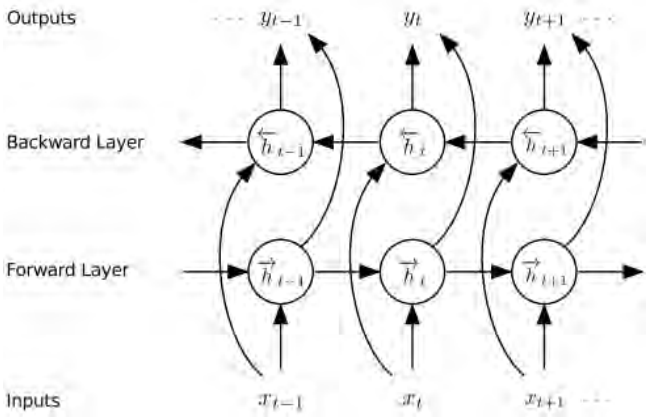


FIGURE 2.4: Architecture of an RNN-LSTM (Graves, Mohamed, and Hinton, 2013). The circular nodes represent LSTM units. Given an input sequence $\mathbf{x} = (x_1, ..., x_T)$, the network computes the forward hidden sequence $\overrightarrow{h}_t$ and the backward hidden sequence $\overleftarrow{h}_t$. The output vector sequence $\mathbf{y} = (y_1, ..., y_T)$ is then computed by $y_t = W_{\overleftarrow{h}y}\overleftarrow{h}_t + W_{\overrightarrow{h}y}\overrightarrow{h}_t + b_y$, where $W_{\overleftarrow{h}y}$ is the input-hidden weight matrix and the b_y terms denote bias vectors.

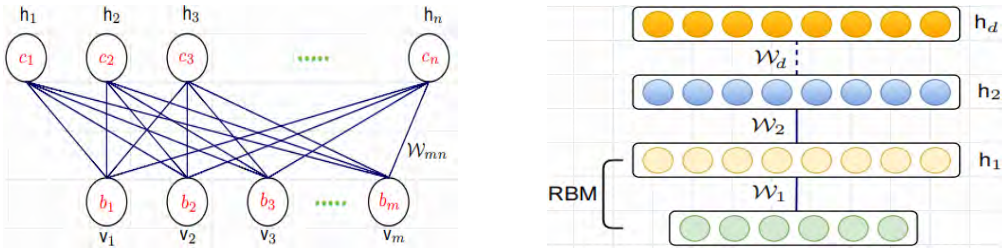


FIGURE 2.5: **(a)** An RBM with m visible units and n hidden units. **(b)** Overview of a DBN composed of d RBMs. Here, the symbols W_1,W_2,...,W_d denote the weight matrices between the connections.

where Z is the partition function computed by summing over possible pairs of (v_i, h_j),

$$Z = \sum_{v_i, h_j} e^{-E(v_i, h_j)}. \tag{2.2}$$

$E(v_i, h_j)$ is the energy function given by

$$E(v_i, h_j) = -\sum_{i=1}^{m} a_i v_i - \sum_{j=1}^{n} b_j h_j - \sum_{i,j} v_i h_j w_{i,j}. \tag{2.3}$$

In equation (2.3), a_i and b_j are biases, $w_{i,j}$ is the weight between v_i and h_j units. In a binary RBM model, there are no direct connections between visible units nor between hidden units. Therefore, given the input data $\mathbf{v}$ through the visible units, the binary state of each unit h_j is 1 with probability

$$p(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_i v_i w_{i,j}), \tag{2.4}$$

where $\sigma(x)$ is the sigmoid function, $\sigma(x) = \dfrac{1}{1 + e^{-x}}$. Given a hidden vector $\mathbf{h}$, we can also reconstruct the states of a visible unit by

$$p(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_j h_j w_{i,j}). \tag{2.5}$$

During the training phase, the weights $w_{i,j}$ and biases a_i, b_j can be updated by solving the following optimization problems

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{i,j}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}, \tag{2.6}$$

$$\frac{\partial \log p(\mathbf{v})}{\partial a_i} = \langle v_i \rangle_{data} - \langle v_i \rangle_{model}, \tag{2.7}$$

$$\frac{\partial \log p(\mathbf{v})}{\partial b_j} = \langle h_j \rangle_{data} - \langle h_j \rangle_{model}. \tag{2.8}$$

where $\langle \cdot \rangle$ denotes an average over the sampled states. The conditional distribution $p(h_j|\mathbf{v})$ in equation (2.4) shows that the hidden layer can be constructed by updating the state of units h_j when given a data vector $\mathbf{v}$. In practice, since all units in the hidden layer are conditionally independent given the visible layer, the state of each unit can be computed by using block Gibbs sampling (Hinton, Osindero, and Teh, 2006). This technique allows to update the state of all the units in parallel. A DBN could be viewed as a stack of several RBMs.

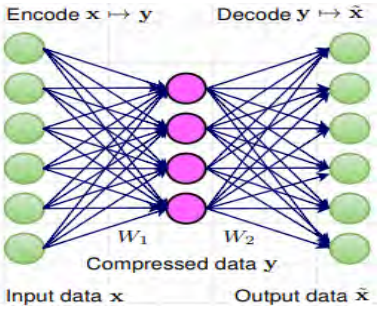


FIGURE 2.6: The typical structure of an autoencoder.

Therefore, training a DBN is performed through training each of its RBM. The work of Hinton, Osindero, and Teh, 2006 provided an efficient procedure to train DBNs. During training, the current hidden layer is regarded as a visible layer for the next hidden layer and training a DBN starts from the lowest RBM. This procedure is repeated layer-to-layer until reaching the highest RBM and known as the *"greedy layer-wise training strategy"*. In general, each component of the DBNs or an RBM acts as a feature extractor on inputs. It extracts *"low level"* features at the bottom hidden layer as well as more *"abstract"* features at the higher hidden layers. For classification tasks, the DBN model could be extended by adding a soft-max layer on the top of its architecture.

2.5 Stacked Denoising Autoencoders (SDAs)

Stacked Denoising Autoencoder (SDA) is another important deep learning architecture. It was first introduced in 2008 by Vincent et al., 2008. The idea of an autoencoder is shortly described as follows: given a set of data points $\mathbf{x} = \{x_1, x_2, ..., x_m\}$, mapping $\mathbf{x}$ to another set of data points $\mathbf{y} = \{y_1, y_2, ..., y_n\}$, where $n < m$. From the compressed set $\mathbf{y}$, we reconstruct a set of $\tilde{\mathbf{x}}$, which approximates the original data $\mathbf{x}$. The mapping $\mathbf{x} \mapsto \mathbf{y}$ is called *"encoding"* and the mapping $\mathbf{y} \mapsto \tilde{\mathbf{x}}$ is called *"decoding"*. Formally, the processes of encoding and decoding are performed as follows

$$\mathbf{y} = W_1 x_i + b_1, \tag{2.9}$$

$$\tilde{\mathbf{x}} = W_2 y_i + b_2, \tag{2.10}$$

where $W_1 \in \mathbb{R}^{m \times m}, W_2 \in \mathbb{R}^{n \times n}$. FIGURE 2.6 illustrates the network architecture of a typical autoencoder. To reconstruct $\tilde{\mathbf{x}}$ and approximate the original data $\mathbf{x}$, we minimize the difference between $\mathbf{x}$ and $\tilde{\mathbf{x}}$ by minimizing the function

$$J(W_1, b_1, W_2, b_2) = \sum_{i=1}^{m} (\tilde{x}_i - x_i)^2. \tag{2.11}$$

From equations (2.10) and (2.11), a SDA can be trained by optimizing the following loss function

$$J(W_1, b_1, W_2, b_2) = \sum_{i=1}^{m} \left((W_1 W_2 x_i - 1)x_i + b_1 W_2 + b_2\right)^2. \tag{2.12}$$

The SDAs are constructed by stacking several autoencoders together to create a deep architecture. The weights are fine-tuned with a back-propagation algorithm. The unsupervised pre-training of each autoencoder is performed in a greedy layer-by-layer manner. Once these SDAs were learned, its output will then be used as the input representations of a supervised learning algorithm for recognition tasks.

2.6 Generative Adversarial Networks (GANs)

In recent years, Generative Adversarial Networks (GANs – Goodfellow et al., 2014) have gained a lot of popularity in the field of computer vision. GAN-based approaches have been used and shown great results in image synthesis (Reed et al., 2016), image super-resolution (Ledig et al., 2017), image-to-image translation (Isola et al., 2017) and so on. In this section, we briefly review the mathematical model behind a GAN framework and its training procedure.

A GAN model consists of two components (see FIGURE 2.7[1]): a generator G and a discriminator D. Given an input noise vector z, which is sampled from a normal distribution $p_z(z)$, the generator G is trained to generate an image x in order to ensure that x is indistinguishable from a real data distribution $p_{\text{data}}(x)$. While training G, we maximize the probability so that x belongs to the given distribution $p_{\text{data}}(x)$. The generated image x is fed into the discriminator D alongside a stream of images taken from the real distribution. In other words, D is trained to estimate the probability of a given sample coming from the real distribution. To this end, we need to make sure that the decisions of the discriminator D over real data are accurate by maximizing $\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]$. Meanwhile, given a fake sample $G(z), z \sim p_z(z)$, the discriminator is expected to output a probability, $D(G(z))$, close to zero by maximizing $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$. On the other hand, the generator is trained to increase the chances of D producing a high probability for a fake example, thus to minimize $\mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]$. When combining both aspects together, D and G are playing a minimax game, in which we should optimize the following loss function $\mathcal{L}(D, G)$

$$\begin{aligned}\min_G \max_D \mathcal{L}(D, G) = \min_G \max_D (&\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] \\ &+ \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))]).\end{aligned} \tag{2.13}$$

In practice, both components G and D are two neural networks. The loss function $\mathcal{L}(D, G)$ from equation (2.13) can be optimized using gradient-based methods since both G and D are differentiable with respect to their inputs and parameters. In 2016, Radford, Metz, and Chintala, 2015 introduced a set of architectures called Deep Convolutional GANs (DCGANs) in order to train GANs in a better way. This study showed that GANs can learn good representations of images for supervised learning and generative modeling. In Chapter 3, we will examine the potentials of GANs in analyzing actions in videos.

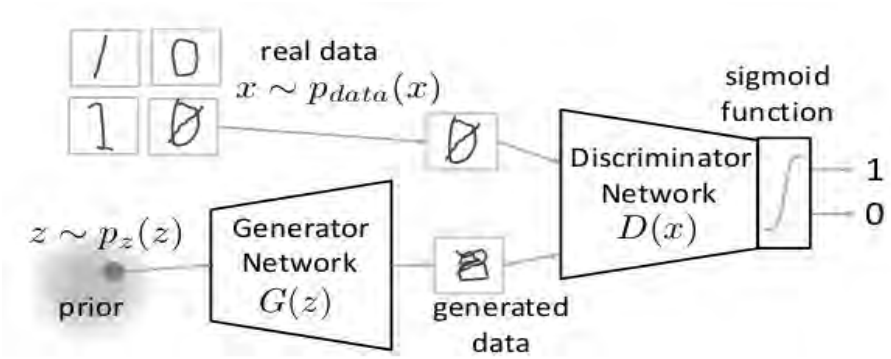


FIGURE 2.7: Training process of a Generative Adversarial Network (GAN).

[1]The figure was taken from https://www.analyticsvidhya.com/blog/2017/06/introductory-generative-adversarial-networks-gans/.

2.7 Conclusion

We have presented in this chapter the basic concepts of deep learning and review some key deep learning algorithms such as Convolutional Neural Networks (CNNs), Recurrent Neural Networks with Long Short-Term Memory units (RNN-LSTMs), Deep Belief Networks (DBNs), Stacked Denoising Autoencoders (SDAs) and Generative Adversarial Networks (GANs) as well as the mathematical concepts behind them. At this stage, the different deep networks are just presented without any comparison between them. The main goal in the context of this PhD work is to proceed to human action recognition and that is why the next chapter is dedicated to the presentation and analysis of deep learning approaches for video-based human action recognition.

Chapter 3

Deep Learning Architectures for Human Action Recognition in Videos: State-of-the-Art

Contents

Chapter overview: Video-based human action recognition is an important yet challenging task in computer vision. The ability to accurately detect and predict actions in unknown videos enables the construction of many important applications such as smart surveillance, human-machine interface, robotics and so on. In recent years, deep learning-based approaches have shown impressive performance and big potential in analyzing and recognizing human actions in videos. Many different deep architectures have been proposed for action recognition and advanced the state-of-the-art in this field. This chapter provides a detailed and comprehensive overview of the current state of deep learning-based human action recognition from RGB-D video sequences. Specifically, we describe the most commonly

used deep architectures for learning human motion features and show how they could be applied to address challenges in action recognition as well as discuss the advantages and limitations of each approach. In particular, through quantitative analyses on three large-scale benchmark datasets including HMDB-51 (Kuehne et al., 2011), UCF-101 (Soomro, Zamir, and Shah, 2012) and NTU-RGB+D (Shahroudy et al., 2016), we identify state-of-the-art deep architectures that have been successfully applied for human action recognition and then provides current trends and open problems for future works. Many public action recognition datasets are also introduced and analyzed, providing the latest achievements and challenges of the field.

3.1 Related reviews and public datasets

3.1.1 Previous reviews

We first consider related earlier reviews in video-based human action recognition. Looking at the major conferences and journals in computer vision and image processing, several earlier surveys have been published (Aggarwal and Cai, 1999; Moeslund and Granum, 2001; Wang, Hu, and Tan, 2003; Moeslund, Hilton, and Krüger, 2006; Turaga et al., 2008). For instance, Aggarwal and Cai, 1999 reviewed methods for human motion analysis, focusing on three major areas: motion analysis, tracking a moving human from a single view or multiple cameras and recognizing human actions from image sequences. Moeslund and Granum, 2001 reviewed approaches on human motion capture. They considered a general structure for motion analysis systems as a hierarchical process with four steps: initialization, tracking, pose estimation, and recognition and then reviewed the papers based on this taxonomy. Wang, Hu, and Tan, 2003 presented an overview on human motion analysis, in which motion analysis was illustrated as a three-level process including human detection, human tracking, and behavior understanding. Moeslund, Hilton, and Krüger, 2006 described the work in human capture and analysis, centered on initialization of human motion, tracking, pose estimation, and recognition. Turaga et al., 2008 reviewed the major approaches for recognizing human actions and activities. They considered *"actions"* as characterized by simple motion patterns, typically executed by a single person. Meanwhile, *"activities"* are more complex and involve coordinated actions among a small number of humans.

Many reviews on human action recognition approaches have been made since 2010 (*e.g.* Poppe, 2010; Weinland, Ronfard, and Boyer, 2011; Popoola and Wang, 2012; Ke et al., 2013; Aggarwal and Xia, 2014; Guo and Lai, 2014). For instance, Poppe, 2010 focused on image representation and action classification methods. A similar survey by Weinland, Ronfard, and Boyer, 2011 also concentrated on approaches for action representation and classification. Popoola and Wang, 2012 presented a survey focusing on contextual abnormal human behavior detection for surveillance applications. Ke et al., 2013 reviewed human action recognition methods for both static and moving cameras, covering many problems such as feature extraction, representation techniques, action detection and classification. Aggarwal and Xia, 2014 introduced a review of human action recognition based on 3D data, especially using RGB and depth information acquired by 3D sensors. Meanwhile Guo and Lai, 2014 provided a review of existing approaches on still image-based action recognition.

Recently, Cheng et al., 2015 reviewed approaches on human action recognition in which all methodologies are classified into two categories: single-layered approaches and hierarchical approaches. Vrigkas, Nikou, and Kakadiaris, 2015 categorized human action recognition methods into two main categories including *"unimodal"* and *"multimodal"*. The authors then reviewed action classification methods for each of these two categories. The work of Subetha and Chitrakala, 2016 mainly focused on human action recognition and human-object interaction methods. Presti and La Cascia, 2016 provided a review of human action recognition based on 3D skeletons, summarizing the main technologies, including both hardware and software for solving the problem of action classification inferred from skeletal data. Recently, another review by Kang and Wildes, 2016 summarized various action recognition and detection algorithms, focused on encoding and classifying motion features. TABLE 3.1 summarizes previous reviews of this field.

To the best of our knowledge, there is no comprehensive review on deep learning-based human action recognition[1]. We believe that such a review is very beneficial for the computer vision community and is what motivates us to realize this work. Different from previous works, we focus on reviewing and analyzing deep learning approaches for human action recognition in videos. Not only to provide a comparative analysis about the current state of deep learning based action recognition approaches, but also to point out new challenges and trends in this field. Our review will also add to the latest reviews on human action recognition in the literature.

TABLE 3.1: Summary of previous reviews on video-based human action recognition and their key points. Ordered by year of publication, earliest to latest.

Authors & Year	Main topics
Aggarwal and Cai, 1999	Human motion analysis, tracking.
Moeslund and Granum, 2001	Motion initialization, tracking, pose estimation, and recognition.
Wang, Hu, and Tan, 2003	Human detection, tracking, action understanding.
Moeslund, Hilton, and Krüger, 2006	Human motion capture, action and behavior analysis.
Turaga et al., 2008	Human behavior recognition.
Poppe, 2010	Feature extraction and classification of human actions.
Weinland, Ronfard, and Boyer, 2011	Full-body action segmentation and recognition.
Popoola and Wang, 2012	Human motion analysis and behavior recognition.
Ke et al., 2013	Action recognition from static and moving cameras.
Aggarwal and Xia, 2014	Human action recognition from 3D data.
Guo and Lai, 2014	Human action recognition from still image.
Cheng et al., 2015	Single-layered and hierarchical approaches for action recognition.
Vrigkas, Nikou, and Kakadiaris, 2015	Human action classification.
Subetha and Chitrakala, 2016	Recognition of action and human-object interactions.
Presti and La Cascia, 2016	Action classification based on 3D skeletal data.
Kang and Wildes, 2016	Human action detection and recognition.

3.1.2 Benchmark datasets for human action recognition in videos

With the increase in the study of human action recognition methods, many benchmark datasets have been recorded and published (*e.g.* Schuldt, Laptev, and Caputo, 2004; Gorelick et al., 2007; Weinland, Ronfard, and Boyer, 2006; Schuldt, Laptev, and Caputo, 2004; Marszalek, Laptev, and Schmid, 2009; Liu, Jiebo Luo, and Shah, 2009; Singh, Velastin, and Ragheb, 2010; Michael and Jake, 2009; Li, Zhang, and Liu, 2010; Wang et al., 2012; Niebles, Chen, and Fei-Fei, 2010; Oh et al., 2011; Kuehne et al., 2011; Sung et al., 2011; Koppula, Gupta, and Saxena, 2013; Yun et al., 2012a; Wolf et al., 2014; Reddy and Shah, 2013; Soomro, Zamir, and Shah, 2012; Wang et al., 2014; Rahmani et al., 2016; Karpathy et al., 2014; Jiang et al., 2014; Gorban et al., 2015; Heilbron et al., 2015; Abu-El-Haija et al., 2016; Shahroudy et al., 2016; Sigurdsson et al., 2016). Much progress in human action recognition has been demonstrated on these standard benchmark datasets. They allow researchers to develop, evaluate and compare new approaches for the problem of human action recognition in videos. In this section, we summarize the most important benchmark datasets, from the early datasets that contain simple actions and acquired under controlled environments, *e.g.* KTH (Schuldt, Laptev, and Caputo, 2004), Weizmann (Gorelick et al., 2007) or IXMAS (Weinland, Ronfard, and Boyer, 2006), to recent benchmark datasets with millions of video samples providing complex actions and human behaviors from the real world scenarios, *e.g.* Sports-1M (Karpathy et al., 2014) and NTU-RGB+D (Shahroudy et al., 2016). TABLE 3.2 shows the datasets and their main descriptions. We divided these benchmarks into four categories, including

[1]The study was conducted at the end of year 2016. New approaches in 2017 and later were not considered. However, the latest approaches on deep learning-based action recognition were mentioned and analyzed in technical sections of this thesis.

single action (category I), human-human interaction, human-object interaction and behavior (category II), surveillance (category III) and sport videos and other types (category IV).

The complexity of each dataset depends much on its recorded setting and acquisition process. For example, early benchmarks such as KTH (Schuldt, Laptev, and Caputo, 2004), Weizmann (Gorelick et al., 2007) or IXMAS (Weinland, Ronfard, and Boyer, 2006) were made under controlled environments for idealized human actions. Specifically, all of them are composed of simple and unrealistic actions with homogeneous background. Many methods have already achieved excellent recognition rates on these benchmarks, *e.g.* 100% on the Weizman (Gorelick et al., 2007) by Ikizler and Duygulu, 2007 or Brahnam and Nanni, 2009. In other words, we can say that the simple datasets have already been solved.

After the success of the action recognition systems on benchmarks produced "*in the lab*", more complex benchmarks have been released, for instance, MSR Action3D (Li, Zhang, and Liu, 2010), UT-Interaction (Michael and Jake, 2009), Daily Activity3D (Wang et al., 2012), CAD-60 (Jaeyong Sung et al., 2012), CAD-120 (Koppula, Gupta, and Saxena, 2013), VIRAT 2.0 (Oh et al., 2011), SBU-Kinect Interaction (Yun et al., 2012a). These datasets aim to provide challenging videos of human action under unconstrained environments with complex background and illumination condition. However, they are still not "*real*" actions. Real and large-scale benchmark datasets play a key role in exploiting machine learning algorithms, in particular deep learning techniques. To solve this problem, many researchers have extracted realistic situations from movies or sport videos on social networks, *e.g.* YouTube, to make new realistic benchmark datasets. See for example: Hollywood-1 (Schuldt, Laptev, and Caputo, 2004), Hollywood-2 (Marszalek, Laptev, and Schmid, 2009), HMDB-51 (Kuehne et al., 2011), UCF-50 (Reddy and Shah, 2013), UCF-101 (Soomro, Zamir, and Shah, 2012), YouTube Liu, Jiebo Luo, and Shah, 2009, Sports-1M (Karpathy et al., 2014), ActivityNet (Heilbron et al., 2015), YouTube 8M (Abu-El-Haija et al., 2016). To build these benchmarks, a general approach is to collect videos from "*in-the-wild*" sources with a large amount of samples and action classes. It can easily be noticed that several datasets are designed for improving the learning performance of deep learning models due to their very large-scales. For example, there are around one million YouTube videos belonging to a taxonomy of 487 classes in Sports-1M (Karpathy et al., 2014) and ActivityNet (Heilbron et al., 2015) provides more than 200 activity classes with 10,024 training videos. The NTU-RGB+D dataset (Shahroudy et al., 2016) contains more than 56 thousand video samples, 4 million frames with 60 different action classes and performed by 40 different subjects. These large-scale datasets are an important premise for the development of deep learning-based approaches because they require a large number of training data and most of the major advances of human action recognition have come with the creation and the publication of such large-scale datasets. FIGURE 3.1 and FIGURE 3.2 provide readers an evolution of action recognition benchmarks in terms of complexity of action and data size, respectively.

The complexity of large-scale datasets also leads to new research and problems. Among them, the current and most important problem in action recognition that needs to be solved by the computer vision community is the problem of *"recognizing complex actions and behaviors in untrimmed videos"*. In fact, experimental results on realistic human action datasets have so far given limited results specially when dealing with a large and varied range of actions. More details about the recognition performance of deep learning based approaches on several large-scale action recognition datasets will be discussed in Section 3.3 of this chapter.

3.2 Deep learning approaches for video-based human action recognition

In this section, we discuss the main challenges in exploiting deep neural networks for human action recognition in videos. We then review different deep learning architectures for action recognition and show their pros and cons. More specifically, we review approaches based on

TABLE 3.2: Some popular benchmark datasets for video-based human action recognition (ordered by year of publication).

Dataset	Author & Year	# Action	Type of action
KTH (I)	Schuldt, Laptev, and Caputo, 2004	6	Walking, jogging, running, boxing, hand waving, etc.
Weizman (I)	Gorelick et al., 2007	10	Walking, running, jumping, etc.
IXMAS (I)	Weinland, Ronfard, and Boyer, 2006	13	Check watch, cross arms, wave, punch, kick, etc.
Hollywood-1 (II)	Schuldt, Laptev, and Caputo, 2004	8	Answer phone, get out car, etc.
Hollywood-2 (II)	Marszalek, Laptev, and Schmid, 2009	12	Answer phone, drive car, eat, run, etc.
YouTube (II)	Liu, Jiebo Luo, and Shah, 2009	8	Basketball shooting, cycling, diving, etc.
MuHAVi (II)	Singh, Velastin, and Ragheb, 2010	17	Walk turn back, run stop, punch, kick, walk fall, etc.
UT-Interaction (II)	Michael and Jake, 2009	6	Shake-hands, point, kick, and punch.
MSR Action3D (II)	Li, Zhang, and Liu, 2010	20	High arm wave, hammer, hand catch, high throw, etc.
Daily Activity3D (II)	Wang et al., 2012	16	Drink, eat, read book, sit still, play game, etc.
Olympic Sports (IV)	Niebles, Chen, and Fei-Fei, 2010	16	High jump, long jump, triple jump, hammer throw, etc.
VIRAT 2.0 (III)	Oh et al., 2011	12	Opening a vehicle trunk, getting into a vehicle, etc.
HMDB-51 (II)	Kuehne et al., 2011	51	Smile, talk, smoke, eat, drink, etc.
CAD-60 (II)	Sung et al., 2011	12	Rinsing mouth, brushing teeth, talking on the phone, etc.
CAD-120 (II)	Koppula, Gupta, and Saxena, 2013	20	Making cereal, reaching, moving, pouring, eating, etc.
SBU-Kinect (II)	Yun et al., 2012a	8	Approach, depart, push, kick, punch, shake hands, etc.
LIRIS (II)	Wolf et al., 2014	10	Discussion between two or more people, taking objects, etc.
UCF-50 (IV)	Reddy and Shah, 2013	50	Diving, drumming, tennis swing, trampoline jumping, etc.
UCF-101 (IV)	Soomro, Zamir, and Shah, 2012	101	Horse riding, hula hoop, ice dancing, skiing, skijet, etc.
UCLA Multiview (II)	Wang et al., 2014	10	Pick up, drop trash, walk around, sit down, stand up, etc.
UWA3D (II)	Rahmani et al., 2016	30	Hand waving, dancing, jumping, etc.
Sports-1M (IV)	Karpathy et al., 2014	487	Juggling club, pole climbing, skipping rope, slack-lining, etc.
THUMOS'14 (IV)	Jiang et al., 2014	101	Daily and sport actions, *e.g.* brushing teeth, driving, etc.
THUMOS'15 (IV)	Gorban et al., 2015	101	Daily and sports actions, *e.g.* brushing teeth, golf swing, etc.
ActivityNet (II)	Heilbron et al., 2015	203	Personal care, eating, drinking, etc.
YouTube-8M (IV)	Abu-El-Haija et al., 2016	N/A†	Span activities, e.g sports and games.
NTU-RGB+D (II)	Shahroudy et al., 2016	60	Drinking, eating, reading, punching, kicking, hugging, etc.

† There are a total of 4716 classes, including human actions.

FIGURE 3.1: Evolution of action recognition benchmarks: (**a**) First-generation action datasets include simple actions with homogeneous background, *e.g.* KTH (Schuldt, Laptev, and Caputo, 2004), Weizman (Gorelick et al., 2007) or IXMAS (Weinland, Ronfard, and Boyer, 2006); (**b**) Second-generation contains more complex actions with background clutter, under controlled environments, *e.g.* UWA3D Multiview Activity II dataset (Rahmani et al., 2016), UCLA Multiview (Wang et al., 2014); (**c**) Third generation provides very complex and large-scale datasets, under realistic scenarios, *e.g.* ActivityNet (Heilbron et al., 2015), or Charades (Sigurdsson et al., 2016).

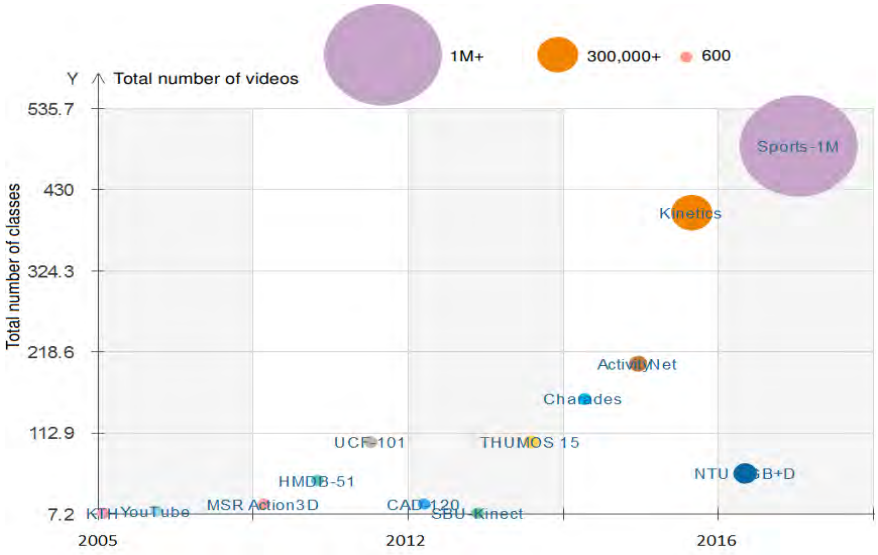


FIGURE 3.2: The number of samples in action recognition benchmarks has moved on from a few hundred videos to millions. The number of action classes also increased over time.

CNNs (Section 3.2.2); RNN-LSTMs (Section 3.2.3); CNN-LSTM (Section 3.2.4); DBNs (Section 3.2.5); SDAs (Section 3.2.6); GANs (Section 3.2.7) and some other architectures (Section 3.2.8).

3.2.1 Deep learning for human action recognition: Challenges

In addition to traditional challenges such as viewpoint variation or occlusion, building deep learning models for video-based action recognition also faces new challenges. The first challenge is the problem of feature representation, in which the goal is to find motion representations that are able to effectively capture and represent the spatio-temporal evolutions of human motion from RGB-D data before feeding to deep learning networks. Second, how to design and optimize high-performance deep learning architectures to model the spatio-temporal dynamics of action from the motion representations for recognition task. In particular, encoding temporal information and model various temporal dynamics of action sequences including both short-term, medium-term, and long-term actions is a big challenge. Third, most of deep learning based approaches require large labeled training datasets. Meanwhile, collecting large labeled training data is costly and time-consuming. Therefore, how to effectively train deep networks on small training data is a problem that needs more research. Last, both deep networks training and inference are computation-intensive processes and how to build a deep learning framework for real applications is an important and challenging task.

3.2.2 Human action recognition based on CNNs

Many research works on human action recognition based on deep learning models have been proposed and published. Among them, one of the most used models is Convolutional Neural Network (CNN). Researchers have successfully applied CNN-based approaches for many visual tasks, including people detection and tracking (Fan et al., 2010; Sermanet et al., 2013; Wang et al., 2015a), human pose estimation (Nowlan and Platt, 1994; Jain et al., 2013; Jain et al., 2014; Gkioxari et al., 2014; Tompson et al., 2014; Chéron, Laptev, and Schmid, 2015), human action recognition (Giese and Poggio, 2003; Sigala et al., 2005; Jhuang, 2007; Kim, Lee, and Yang, 2007; Ji et al., 2013; Simonyan and Zisserman, 2014a; Wang et al., 2014; Wang, Qiao, and Tang, 2015; Tran et al., 2015; Wang et al., 2015e; Dobhal et al., 2015; Liu et al., 2015; Cao et al., July, 2016; Mo et al., 2016; Singh, Arora, and Jawahar, 2016), event detection and crowded scene understanding (Gan et al., 2015; Shao et al., 2015; Castro et al., 2015; Xiong et al., 2015). In this section, we review CNN-based approaches for the task of human action recognition.

Early works on CNN-based human action recognition

Early work on applying CNNs was made in 1995 by Nowlan and Platt, 1994 for hand tracking and recognizing. In their work, a CNN model with two convolutional windows and a subsampling layer was proposed to locate the hand and recognize whether it is closed or open. This architecture achieved a high accuracy on a dataset of 900 video samples. However, the complex structured backgrounds of images has a significant impact on recognition accuracy. Inspired by the first CNN model of Fukushima, 1980, Giese and Poggio, 2003 proposed the use of receptive fields to build a hierarchical feedforward architecture for the recognition of biological movements, such as walking, running or various full-body actions. In a related study, Sigala et al., 2005 also developed a hierarchical model for detecting a walker based on the use of neural detectors, which are able to extract motion features with different levels of complexity. In 2007, Jhuang, 2007 proposed an extension model from the work of Giese and Poggio, 2003 for the recognition of human actions from video sequences. FIGURE 3.3 provides details about this architecture. Kim, Lee, and Yang, 2007 also used a CNN model and a weighted fuzzy min-max neural network (WFMM - Kim, Lee, and Yang, 2006) for human action recognition. The authors used a CNN to generate a set of feature maps from the pretreated data and a WFMM (Kim, Lee, and Yang, 2006) was used as a classifier. These early works (Nowlan and Platt, 1994; Giese and Poggio, 2003; Sigala et al.,

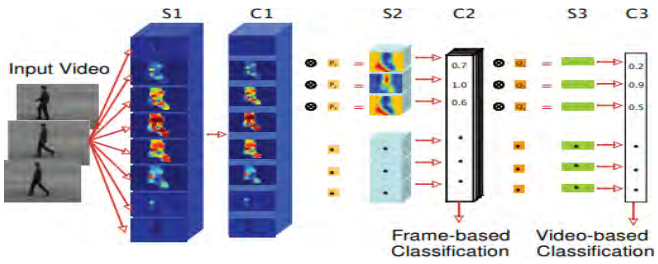


FIGURE 3.3: A CNN-based framework for human action recognition proposed by Jhuang, 2007. Given a gray-value video sequence as input, the first layer S_1 locates objects in image frames by using spatio-temporal filters. Each unit of C_1 is computed by applying a local-max over units of S_1 for down-sampling. From C_1, a template matching operation is performed for identifying intermediate-level features. C_2 is then constructed by computing the global max over S_2. The high-level features are extracted in S_3 through a template matching and the high-level features C_3 are computed from S_3 using the same way like computing C_2. The last layer is a linear multi-class SVM classifier that is used to classify actions with features provided by C_3.

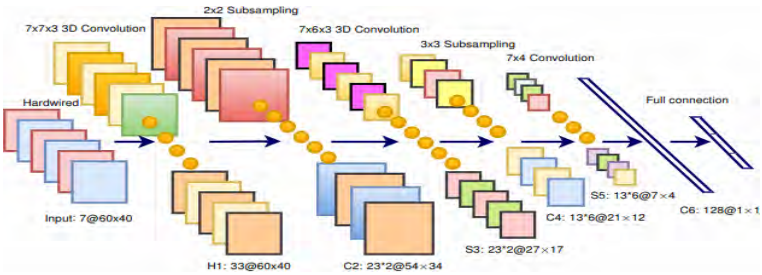


FIGURE 3.4: The 3D-CNN architecture for human action recognition proposed by Ji et al., 2013. The first layer was used to generate multiple channels of information from the input frames (*e.g.* the information of gray-level, gradient, or optical flow). The model applied 3D convolutions for each channel to compute feature maps (C_2). The next layer (S_3) was obtained by applying subsampling operations on each feature map from (C_2). This procedure was repeated until obtaining feature maps (S_5), which was then connected with a full connection layer for classification. The figure was redesigned from Ji et al., 2013.

2005; Jhuang, 2007; Kim, Lee, and Yang, 2007) share the same characteristics – that is, they are mostly based on simple CNN models with several layers. However, they are important platforms for the development of the field later.

3D Convolutional Neural Networks (3D-CNNs) for human action recognition in videos

An important study on applying CNN models for recognizing human actions in videos has been introduced by Ji et al., 2013. In order to exploit the temporal information of human motion, the authors used a novel three-dimensional Convolutional Neural Network (3D-CNN) architecture to learn motion representations. This architecture used 3D kernels in the convolution stages to extract motion features from both spatial and temporal dimensions. This improvement can be applied to contiguous frames in videos to extract multiple features. FIGURE 3.4 illustrates the 3D-CNN architecture in more detail. Experimental results have shown that this model outperforms the frame-based 2D-CNN models. Motivated by Ji et al., 2013, Wang et al., 2014 also built a deep architecture using 3D-CNN that is able to recognize actions from RGB-D data. Tran et al., 2015 investigated in detail the 3D-CNN models and

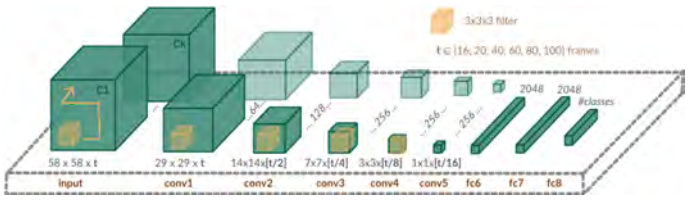


FIGURE 3.5: The 3D-CNN-LTC network proposed by Varol, Laptev, and Schmid, 2018. Convolutions with 3×3×3 kernels (followed by Max Pooling and ReLU layers) were applied in the first 5 layers. Network input channels including optical flow or three-channel appearance (R, G, B) were defined for different temporal resolutions.

showed that it outperforms the 2D-CNNs in modeling human motion information on various recognition tasks. Moreover, this study found that the best kernel length for 3D-CNN is $3 \times 3 \times 3$ size.

A visible disadvantage of 3D-CNNs is the increasing number of parameters of the network. To reduce the complexity of 3D-CNN models, Sun et al., 2015 proposed a factorized spatio-temporal convolutional network for human action recognition that factorizes the 3D convolution kernels into 2D spatial kernels and followed by 1D temporal kernels. 3D-CNNs have also been investigated for action recognition by Varol, Laptev, and Schmid, 2018. The authors extended 3D-CNNs with long-term temporal convolutions (LTC) to learn motion representations. FIGURE 3.5 illustrates a 3D-CNN-LTC network. Varol, Laptev, and Schmid, 2018 also demonstrated that the long-term temporal convolutions and low-level representations (*e.g.* raw values of video pixels, optical flow) are important for accurate learning of human action.

Multi-stream Convolutional Neural Networks (Multi-CNNs) for action recognition

The two-stream convolutional network (two-stream CNN) proposed by Simonyan and Zisserman, 2014a has shown strong performance for video-based action recognition task. This model consists of a spatial stream and a temporal stream where each stream is executed by a CNN. The first stream recognizes actions from a single frame, while the second recognizes actions from motion information of multi-frame optical flow. These two streams are then combined for the classification task. Experimental results showed that the two-stream CNN improves recognition accuracy. This architecture has been seen as the most effective approach of applying deep learning to action recognition with limited training data. In Section 3.3.2, readers can find a quantitative performance analysis of action recognition models based on deep learning where two-stream CNN based approaches play a prominent role. Inspired by the work of Simonyan and Zisserman, 2014a, many two-stream CNN based approaches have been proposed for solving action recognition problems, *e.g.* Chéron, Laptev, and Schmid, 2015, Wang et al., 2016d, Wang et al., 2016a, or Xiong et al., 2016. Unlike the two-stream architecture developed by Simonyan and Zisserman, 2014a, Liu et al., 2015 added a module called stCNN (Spatio-Temporal Convolutional Neural Network) to a standard CNN model for exploiting motion and content-dependent features concurrently. Experiments on KTH (Schuldt, Laptev, and Caputo, 2004) and UCF-101 (Soomro, Zamir, and Shah, 2012) datasets showed that the recognition accuracy for motion-content combined was better when compared with motion alone. Singh, Arora, and Jawahar, 2016 addressed the problem of understanding egocentric activities by using a three-stream CNN architecture. More specifically, the authors proposed a framework for the recognition of wearer's actions. First, a CNN model was trained for learning features from egocentric cues including hand masks, head motions and saliency maps. Then, the proposed network was extended by adding two more streams corresponding to spatial and temporal streams. Experiments showed that the model with more streams was able to improve recognition performance.

In a more recent study, Wang, Farhadi, and Gupta, 2016 divided an input video consisting

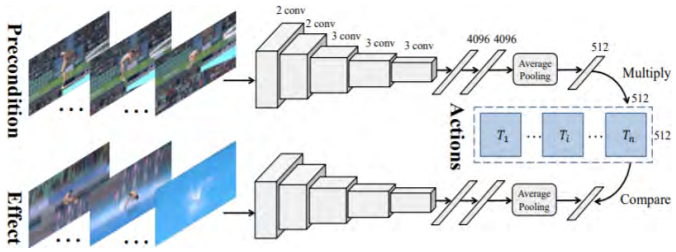


FIGURE 3.6: The Siamese network architecture proposed by Wang, Farhadi, and Gupta, 2016. See text for description.

of t frames $\mathcal{X} = \{x_1, x_2, ..., x_t\}$ into two sets: the precondition state frames $\mathcal{X}_p = \{x_1, ..., x_{z_p}\}$ and effect state frames $\mathcal{X}_e = \{x_{z_e}, ..., x_t\}$. A two-stream CNN architecture called Siamese network has been designed for learning human actions (see FIGURE 3.6). The first stream was trained on the precondition state frames $\mathcal{X}_p$ and the second was trained on the effect state frames $\mathcal{X}_e$. To predict the action classes, the authors applied transformations on the output features of the precondition stream and compare with the output features provided by the effect stream. Karpathy et al., 2014 studied the performance of CNNs by trying to predict and classify on Sports-1M (Karpathy et al., 2014) dataset which consists of more than one million sport videos. A multi-resolution CNN architecture with two-stream of processing has been proposed. The results showed that CNNs are capable of learning powerful features and significantly outperformed hand-crafted features based approaches. Wang et al., 2015d also proposed the use of multi-CNNs to learn actions from sequences of depth maps. Given a sequence of depth maps, 3D points are created and three Depth Motion Maps (DMMs) are constructed by projecting the 3D points to the three orthogonal planes. Three CNNs were constructed based on AlexNet (Krizhevsky, Sutskever, and Hinton, 2012b) to extract motion features from each DMM and then classify them into classes.

Among the local space-time features, trajectory-based features are generally considered to be one of the best ways to describe motion (Wang et al., 2011; Wang and Schmid, 2013; Beaudry, Péteri, and Mascarilla, 2016). Wang, Qiao, and Tang, 2015 combined the benefits of improved trajectories (Wang and Schmid, 2013) and a two-stream CNN architecture to design an effective representation of video feature called *"Trajectory-Pooled Deep Convolutional Descriptor"* for human action recognition in videos. Experimental results showed that this framework has obtained state-of-the-art performance on the UCF-101 (Soomro, Zamir, and Shah, 2012) and HMDB-51 (Kuehne et al., 2011) datasets. Liu, Liu, and Chen, 2017 used a two-stream CNN-based model to learn and recognize actions from skeletal data. To this end, a color encoding method was proposed to map skeleton joints into color images. Visual and motion enhancement techniques were then exploited to generate more discriminative features in obtained images. This method eliminated the effect of view variations, while achieved high performance levels and required less computation for the training phase. Ke et al., 2017 also exploited multiple-stream CNN to learn motion features from skeleton sequences. Unlike Liu, Liu, and Chen, 2017, the authors generated three clips corresponding to the three 3D coordinates of the skeleton joints. For each clip, the relative positions of joints were computed and fed directly into three pre-trained CNN models. This work also indicated that the performance of the proposed model will decrease when concatenating three gray clips into one single color clip. Most recently, Tran and Cheong, 2017 showed that we can improve performance of the two-stream CNN models by sharing information between two streams during the training phase. More details are shown in FIGURE 3.8. Most recently, Carreira and Zisserman, 2017 introduced a two-stream CNN model based on Inception-v1 (Ioffe and Szegedy, 2015), namely *"Two-Stream Inflated 3D ConvNets (I3D)"* to learn the spatio-temporal features of human actions. By inflating 2D-CNN into 3D and bootstrapping 3D filters from 2D filters to build very deep CNN architectures, the proposed I3D network showed its state-of-the-art performance on UCF-101 (Soomro, Zamir, and Shah, 2012) and HMDB-51 (Kuehne et al., 2011) datasets.

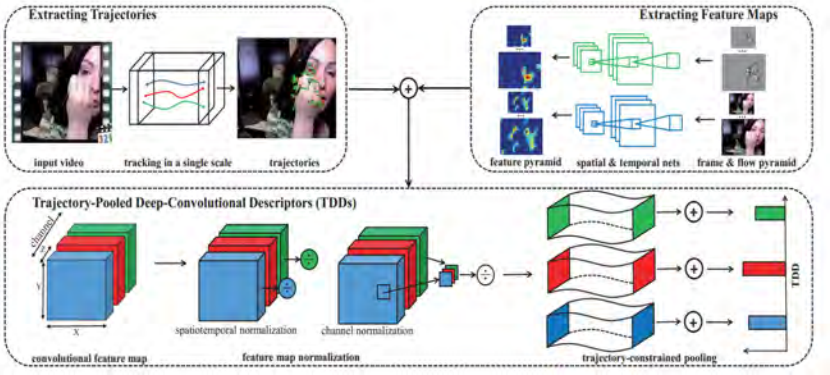


FIGURE 3.7: A deep learning architecture for action recognition proposed by Wang, Qiao, and Tang, 2015. Given an input video, the model extracted motion trajectories. Multi-scale convolutional feature maps were extracted by a CNN at the same time. Trajectory pooled Deep-convolutional Descriptors (TDDs) were then estimated from a set of improved trajectories and convolutional feature maps.

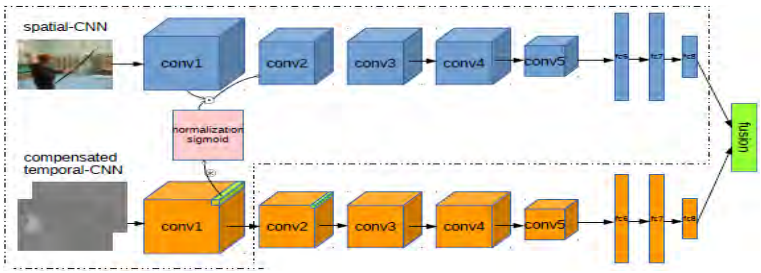


FIGURE 3.8: Two-stream CNN framework for action recognition proposed by Tran and Cheong, 2017. Two streams of RGB frames and optical flows are fed into two separate CNNs, in which the spatial stream models scene and object contexts, while the temporal stream provides some motion-based attentions on foreground actions. The leverage attentions provided by temporal stream is shared to assist recognition processes in spatial-stream via cross-link layers.

Very deep Convolutional Neural Networks (D-CNNs) for action recognition

Very deep convolutional neural networks such as VGG-Net (Simonyan and Zisserman, 2014b), GoogLeNet (Szegedy et al., 2015a) have achieved significant success for many object recognition and classification tasks. Several authors started to exploit these architectures for action recognition problems. For instance, Wang et al., 2015c introduced a very deep two-stream CNN for action recognition based on VGG-16 and GoogLeNet (Szegedy et al., 2015a). Feichtenhofer, Pinz, and Zisserman, 2016 proposed a two-stream CNN architecture in which the deep CNN model VGG-M-2048 (Chatfield et al., 2014) and very deep model VGG-16 (Simonyan and Zisserman, 2014b) have been used. The performance comparison between VGG-M-2048 and VGG-16 models on the UCF-101 (Soomro, Zamir, and Shah, 2012) and HMDB-51 (Kuehne et al., 2011) datasets showed that using of deeper networks helps to improve learning performance (see TABLE 3.3). In addition, GoogLeNet (Szegedy et al., 2015a) and VGG-Net (Simonyan and Zisserman, 2014b) have also been used to design the two-stream CNN in the work of Wang et al., 2015b. Fernando et al., 2016 trained VGG-16 (Simonyan and Zisserman, 2014b) on HMDB-51 (Kuehne et al., 2011), UCF-101 (Soomro, Zamir, and Shah, 2012) and Hollywood-2 (Marszalek, Laptev, and Schmid, 2009) datasets to obtain motion features for an action recognition task. These features were then encoded by a method called "*Hierarchical Rank Pooling*". This framework allows encoding the temporal dynamics of a video sequence for action recognition. Specifically, a video sequence was encoded at multiple levels and the output of each level is a sequence of vectors that captures higher-order dynamics of its previous level. The final representation was used to train an SVM classifier for classification of actions. The residual learning (ResNet - Kaiming et al., 2016), a state-of-the-art CNN and one of the deepest CNN models at the moment[2] has been exploited for human action recognition. For example, Feichtenhofer, Pinz, and Wildes, 2016 suggested to use 50-layer ResNets to design a two-stream CNN. Their experiments showed a state-of-the-art performance on the UCF-101 (Soomro, Zamir, and Shah, 2012) and HMDB-51 (Kuehne et al., 2011) datasets. Based on the success of the previous two-stream ResNet (Feichtenhofer, Pinz, and Wildes, 2016), Feichtenhofer and colleagues continued to exploit very deep CNN networks for building two-stream multiplier networks (Feichtenhofer, Pinz, and Wildes, 2017). ResNet-50 and ResNet-152 networks have been used to learn motion features from appearance and motion streams and reported excellent performances on the UCF-101 (Soomro, Zamir, and Shah, 2012) and HMDB-51 (Kuehne et al., 2011) datasets. However, it is clear that very deep CNN based frameworks require much computation resource to train and optimize.

TABLE 3.3: Performance comparison of VGG-M-2048 with VGG-16 on the UCF-101 and HMDB-51 datasets, reported by Feichtenhofer, Pinz, and Zisserman, 2016.

Dataset	**UCF-101**		**HMDB-51**	
Model	VGG-M-2048	VGG-16	VGG-M-2048	VGG-16
Spatial	74.22%	**82.61**%	36.77%	**47.06**%
Temporal	82.34%	**86.25**%	51.50%	**55.23**%
Spatio-Temporal	85.94%	**90.62**%	54.90%	**58.17**%

3.2.3 Human action recognition based on RNNs

As pointed out in the previous chapter, the main advantage of RNN-LSTMs is the capacity to model long-term contextual information of temporal sequences. This advantage puts RNN-LSTM at one of the best sequence learners for time-series data, including visual information of human action. Many RNN-based action recognition approaches have been proposed and reported their promising performances in the literature. For instance, Du, Wang, and Wang, 2015, Song et al., 2017, Zhu et al., 2016b, Li et al., 2016b, and Liu et al., 2016b used RNN-LSTM networks to model human motions from skeleton sequences provided by depth sensors. These approaches share the same strategy, in which RNN-LSTMs learn directly motion

[2]The year 2016.

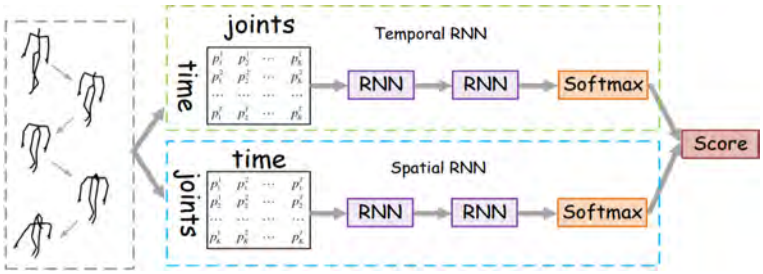


FIGURE 3.9: A two-stream RNN for skeleton-based action recognition by Wang and Wang, 2017. See text for description.

features and classify them into classes from 3D geometric features of skeletons. Experiments on the state-of-the-art benchmark datasets demonstrated the effectiveness of these methods. Like CNN-based methods, two-stream RNN has also been exploited for action recognition, *e.g.* Wang and Wang, 2017 presented a two-stream RNN to model spatial and temporal dynamics of human motion from skeleton sequences. As shown in FIGURE 3.9, the temporal stream used a RNN to learn the temporal dynamics from the 3D coordinates of joints over time. At the same time, another RNN was used to learn the spatial dependency of joints. The two streams were then combined by late fusion for recognition task. Zhu et al., 2016b also proposed the use of a deep LSTM to learn human action from skeletons. Instead of feeding all skeleton joints into the network, the authors defined the co-occurrence of some key joints that can intrinsically characterize the human actions and then modeled these co-occurrence features by RNN-LSTM networks for recognizing actions. In addition, to prevent overfitting for the deep LSTM network, a new dropout technique was introduced. The new dropout layer allows the dropping of the internal gates, cell and output response for each LSTM neuron. Experimental results showed that this architecture helped learning better parameters. Recently, Liu et al., 2017c introduced a "*Global Context-Aware Attention LSTM (GCA-LSTM)*" – a new class of LSTM network for skeleton-based action recognition. The GCA-LSTM consists of two LSTM layers, in which the first layer encodes each skeleton sequence into a global context memory. The second layer learns attention from the original sequence with the assistance of the global context memory. This process is then repeated to generate the global contextual information for classification task. Lee et al., 2017 proposed ensemble Temporal Sliding LSTM networks for action recognition from skeletal data. As shown in FIGURE 3.10, the coordinates of input skeleton sequences were transformed so that the data can be robust to scale, rotation and translation. Then, the motion features were employed and processed with multi-term LSTMs containing short-term, medium-term and long-term LSTMs. Experimental results showed that the proposed feature representation and temporal sliding LSTM networks dramatically enhanced the performance of action recognition.

One drawback with RNN-LSTMs based approaches is that, LSTMs are composed of many parameters per unit. It makes these models more complex in terms of computational complexity, specially in performing action recognition on very large-scale datasets that could easily lead to the problem of vanishing gradients[3].

3.2.4 Fusion of CNNs with LSTM units for human action recognition

As discussed in earlier sections, CNNs have shown their effectiveness in learning spatial features. Meanwhile, RNN-LSTMs are able to effectively model the temporal information of human actions. Therefore, the studies of Baccouche et al., 2011, Ng et al., 2015, Donahue et al., 2015, Sharma, Kiros, and Salakhutdinov, 2015, Ibrahim et al., 2016a, Singh et al., 2016, Li et al., 2016a, Wu et al., 2016, and Wang et al., 2016e tackled the question of understanding human actions by combining CNNs and RNN-LSTMs for building more powerful action

[3]As more layers using certain activation functions are added to neural networks, the gradients of the loss function approaches zero, making the network hard to train.

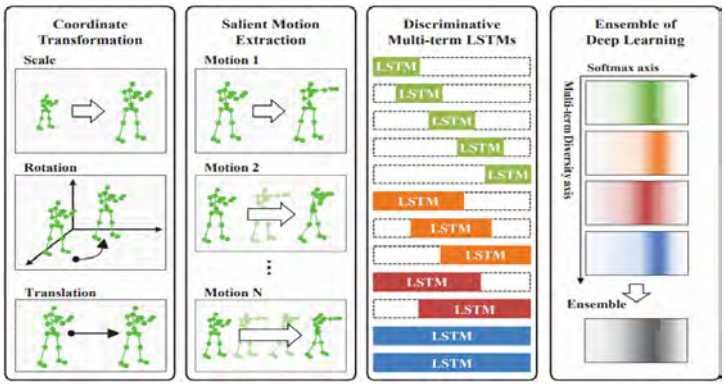


FIGURE 3.10: The main four phases of the proposed ensemble deep learning LSTM by Lee et al., 2017. See text for description.

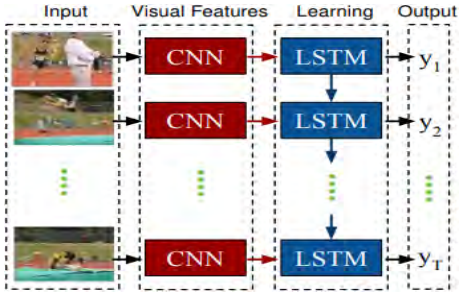


FIGURE 3.11: A deep learning framework that combines CNNs and LSTMs for action recognition proposed by Donahue et al., 2015. The proposed framework processes the visual input with CNNs, whose outputs are then fed into LSTMs, which finally produce a variable-length prediction. Both the CNN and LSTM weights are shared during the training phase.

recognition frameworks. The general idea of these works is to use the standard CNN models such as AlexNet (Krizhevsky, Sutskever, and Hinton, 2012b), VGGNet (Simonyan and Zisserman, 2014b), or GoogLeNet (Szegedy et al., 2015a) for extracting motion features from input video. Then, LSTMs were connected to the output of the CNNs to classify sequences using learned features. FIGURE 3.11 shows a typical example of using CNNs and LSTMs for action recognition task.

Recently, the trend of combining CNNs and LSTMs continues to receive attention (Aliakbarian et al., 2017; Du, Wang, and Qiao, 2017; Sun et al., 2017; Zhu, Vial, and Lu, 2017). For example, Aliakbarian et al., 2017 exploited three-stream CNN-LSTM to predict actions very early in videos. This model extracted the contextual information and then combined with local action information (see FIGURE 3.12). To predict the correct action classes as early as possible, a new loss function for action anticipation has been introduced. Du, Wang, and Qiao, 2017 introduced an end-to-end recurrent network for action recognition, which is in fact a CNN-LSTM network. Each video frame was fed into a CNN to extract features. A pose attention mechanism and the previous hidden stage of LSTM network were used to learn body part features from CNN features. These features were combined by a pooling layer and fed into LSTM for action recognition. Sun et al., 2017 also proposed a new extension of LSTM architecture to learn action patterns using multiple modalities, *i.e.* RGB frames and optical flow. The proposed architecture, namely L^2STM, is a two-stream LSTM with control gates that allows sharing of features between two streams. In each stream, a CNN was used to produce high-level feature maps before feeding them into LSTMs. In another study, Mahasseni and Todorovic, 2016 used a parallel architecture to recognize actions with

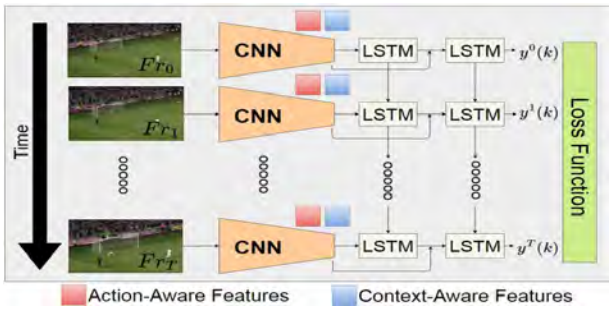


FIGURE 3.12: Overview of three-stream CNN-LSTM framework proposed by Aliakbarian et al., 2017. Given a small portion of a video, the first stage of the network focuses on extracting global, context-information. The second stage extracts local, action-aware information where the action occurs and then combine with obtained context-information.

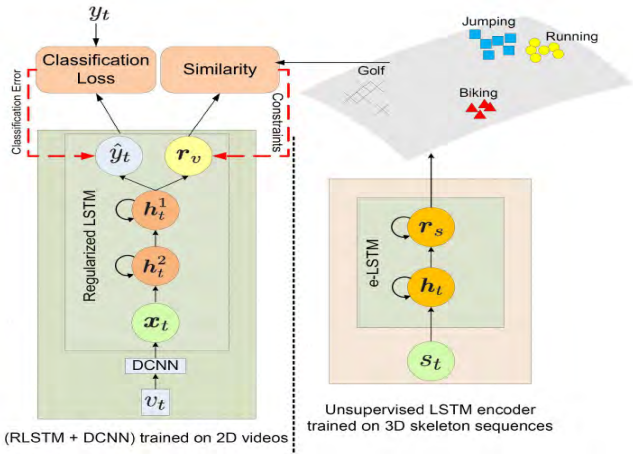


FIGURE 3.13: A parallel deep learning architecture with RNN-LSTM proposed by Mahasseni and Todorovic, 2016. See text for detail.

multi-source data. More specifically, a RNN-LSTM was trained in an unsupervised manner on 3D human skeleton sequences. At the same time, another RNN-LSTM with a CNN was trained on 2D videos. The outputs were then compared to improve the ability of the system. CNN-LSTM based networks have also been used to analyze collective activities (Ibrahim et al., 2016b; Wang, Ni, and Yang, 2017). For instance, Ibrahim et al., 2016b presented a hierarchical model based on CNN-LSTM for group activity recognition. Given a set of detected and tracked people in videos, each person was fed into a CNN, followed by a LSTM layer to model individual action. The outputs of all LSTM layers were then fed to a pooling layer and a group level LSTM layer to recognize the whole action. More recently, Wang, Ni, and Yang, 2017 proposed a recurrent interactional context encoding framework based on CNN-LSTM to model three levels of interaction, including single actions, group human interactions, and group to group interactions. Specifically, given tracklets of a group of persons, each tracklet was fed into a CNN to learn motion features. A LSTM network was connected to each CNN to model single action. To model group level interaction, the authors utilized a context encoder by combining the outputs of single person level LSTM networks. Finally, the encoding results were fed into other LSTM networks to identify group to group interaction.

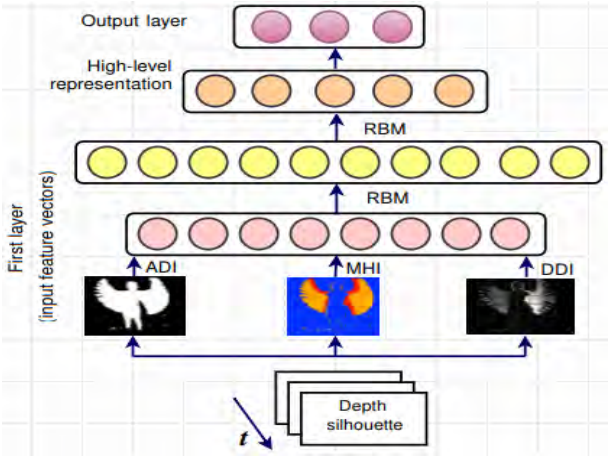


FIGURE 3.14: An overview of the DBN architecture for human action recognition proposed by Foggia et al., 2014. Three derived images (ADI, MHI, and DDI) were computed from depth images and fed into the first level of the network. More abstract representations were then obtained at higher levels. Finally, the classification was done by using a feed-forward neural network.

3.2.5 Human action recognition based on DBNs

DBNs have become popular deep learning models after the key paper by Hinton, Osindero, and Teh, 2006 was presented in 2006. A comparative evaluation by Tang, 2008 showed that DBNs seem ideal for semi-supervised learning, in which we do not need much labeled data. Early work on DBNs was successfully applied for handwritten digits recognition (Hinton, Osindero, and Teh, 2006) and object recognition (Nair and Hinton, 2009; Lee et al., 2009). In 2007, Taylor, Hinton, and Roweis, 2007 extended the RBM model by connecting two more visible layers to the hidden layer for modeling human motion. The new model, called the conditional RBM (cRBM) allows to find a single set of parameters that simultaneously capture several different kinds of motion after training on skeleton sequences. Then, the authors successfully constructed a DBN from cRBMs. Experiments on two motion datasets have demonstrated that this model is able to effectively learn different kinds of action, as well as the transitions between these kinds. In another research, Zhang et al., 2014 used a modified DBN model for recognizing human actions in real-time from skeleton data. To achieve this goal, the authors used cRBMs as proposed by Taylor, Hinton, and Roweis, 2007 to create the new DBN architecture with two hidden layers. The proposed model was trained and tested by using the skeletal representation of MSR Action3D (Li, Zhang, and Liu, 2010) and MIT datasets (Hsu, Pulli, and Popović, 2005). The obtained results showed that the recognition accuracy depends on the number of frames. For example on the MIT datasets (Hsu, Pulli, and Popović, 2005), the accuracy when using one frame is 98.34%. Meanwhile, the accuracy can reach 100% when the number of frames is more than 30.

Foggia et al., 2014 proposed a DBN-based method for recognizing human actions with depth images. A DBN model was constructed as shown in FIGURE 3.14. Three types of well-known feature including the Average Depth Image (ADI), Motion History Image (MHI), and Depth Difference Image (DDI) were computed and encoded as low-level data representations in the first layer. The high-level representations were then extracted by the proposed model for the recognition task. Experimental results on MIVIA (Foggia et al., 2013) and MHAD (Ofli et al., 2013) datasets are very promising. Ali and Wang, 2014 presented a framework based on DBN to recognize and identify human actions. To speed up learning time, the Fast Fourier Transform (FFT – Heckbert, 1995) was used for converting images to the frequency domain. The model was first pre-trained with KTH dataset (Schuldt, Laptev, and Caputo, 2004) and then exploited to predict actions.

3.2.6 Human action recognition based on SDAs

As pointed out Chapter 2, SDAs can be trained to reconstruct the input from a corrupted version of it. The first successful application based on the encoder-decoder model was presented in 2007 by Huang, Boureau, and LeCun, 2007 for object recognition tasks. Based on the principle of this model, Moez et al., 2012 proposed a solution for learning of sparse spatio-temporal features of human motions based on an autoencoder. Experiments on the KTH (Schuldt, Laptev, and Caputo, 2004) and GEMEP-FERA (Valstar et al., 2011) datasets showed a comparable performance to methods using hand-crafted features. Some other autoencoder-based approaches have also been proposed in the works of Wu et al., 2014, Xie et al., 2014, Hasan and Roy-Chowdhury, 2014. For instance, Wu et al., 2014 constructed a 3-layer SDA architecture for human action recognition using skeleton information captured by a Kinect sensor. To recognize human actions, Xie et al., 2014 used an SDA architecture with 3-hidden layers to learn contour features from a single depth frame. The obtained high-level features were then used for the classification task. Hasan and Roy-Chowdhury, 2014 presented an autoencoder-based framework for learning to recognize human actions from streaming videos, also called *"Online Action Recognition – OAR"*. This method was executed through two phases: "*initial learning*" and "*incremental learning*". Specifically, given a streaming video with a few labeled actions, the first phase extracted space-time interest points (STIP – Laptev, 2005) of the motions. These features were then encoded by a sparse autoencoder, followed by a softmax layer for classification. To recognize human actions in unlabeled frames, the incremental learning phase used the sparse autoencoder and the learned parameters in the initial learning phase, but in an unsupervised manner. In addition, the authors also used an active learning technique to reduce the amount of manual labeling of classes. Recently, Shahroudy et al., 2017 introduced a new deep autoencoder to learn RGB and depth features in videos. Each layer of the proposed architecture is an autoencoder based on component factorization unit, which was used to extract multimodal input features into modality-specific parts. Each input modality has specific features that carry discriminative information for the recognition task. Through experiments, Shahroudy et al., 2017 showed that the proposed deep autoencoder outperforms many approaches based on single modality.

The long training time is a disadvantage of SDAs when dealing with large-scale problems. To overcome this limitation, Chen et al., 2012 proposed a novel variant of SDAs, namely "*mSDA*". Experiments on the same dataset showed that mSDA matched the performance of SDA, whilst reducing the training time down to 450 times. Taking advantages of the mSDA, Gu et al., 2015 trained an mSDA network for multi-view action recognition. An mSDA was trained over all the camera views and the trained model was used to generate features for each camera view respectively. These obtained features from all the camera views were then combined to create a single integrated representation and then be used as the input of a classifier. The evaluation on three multi-view action datasets provided that this model achieved state-of-the-art recognition performance.

3.2.7 GANs for human action recognition

So far in the research community, GANs (Goodfellow et al., 2014) have been primarily used for sample generation. However, they can be exploited for learning video representations and applied for human action recognition in videos. For instance, Vondrick, Pirsiavash, and Torralba, 2016 capitalized on recent advances in GANs for both action classification and prediction in videos. A two-stream generative model has been built for learning scene dynamics. This study is an open research opportunity for designing predictive models for understanding human actions because determining when an action may occur in a continuous video is a big challenging task. Yeung et al., 2016 presented the first end-to-end approach for learning to detect actions in videos. The network that takes a long video as input, and outputs the temporal bounds of detected action instances. More specifically, the proposed network consists of two main components: an observation network and a recurrent network. The observation network encodes visual representations of video frames. Meanwhile, the recurrent network sequentially processes these observations and decides both which frame to

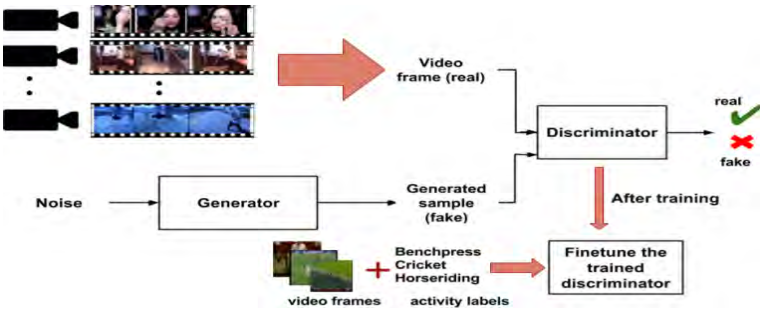


FIGURE 3.15: An overview of the GAN-based approach to learn action representation proposed by Ahsan, Sun, and Essa, 2018. See text for more detail.

observe next and when to emit a prediction. Li et al., 2017c used a Conditional Generative Adversarial Network (cGAN – Mirza and Osindero, 2014) to localize an action by generating its action mask. The generated masks were then used as an additional channel for learning motion features and recognition. Most recently, Ahsan, Sun, and Essa, 2018 exploited GAN to learn action representations in videos with little to no supervision information. To this end, the authors trained a Deep Convolutional Generative Adversarial Network (DCGAN) (Radford, Metz, and Chintala, 2015) on a large video action dataset in an unsupervised manner. To recognize human actions, the pre-trained discriminator network from the GAN framework was fine-tuned on another dataset with supervision information (FIGURE 3.15). Competitive performance on the HMDB-51 dataset (Kuehne et al., 2011) proved that GANs are able to capture enough information to learn useful representations of human actions in videos. Although GANs have shown big potentials in learning video representations, the big disadvantage of GANs is that, these networks are very hard to train and requires a lot of trial-and-error regarding the network structure and training methodology.

3.2.8 Other deep architectures for human action recognition

Some other deep architectures have also been used for human action recognition and related recognition tasks such as group activity analysis, or prediction of physical interactions. Sparse coding (Olshausen and Field, 1996; Lee et al., 2006; Yu, Lin, and Lafferty, 2011) is also another potential deep model for recognizing human action.The success of sparse representation in various fields including pattern recognition (Raina et al., 2007; Yang, Yu, and Huang, 2010) or image classification (Yang et al., 2009) have shown that it could flexibly adapt to diverse low level natural signals. The sparse representations of the signals are then used as image features that are sent directly into the classifiers. Therefore, many authors (Zhu et al., 2010; Lu and Peng, 2013; Guha and Ward, 2012; Alfaro, Mery, and Soto, 2016) have exploited the advantages of sparse coding for solving human action recognition problems. Later, some novel deep architectures for action recognition have been published in the literature (Ullah and Petrosino, 2015; Ullah and Petrosino, 2016; Rahmani, Mian, and Shah, 2018). For instance, Ullah and Petrosino, 2015 employed a CNN and a pyramidal neural network (PyraNet – Phung and Bouzerdoum, 2007) to recognize human action. A strict 3D pyramidal neural network called "*3DPyraNet*" was constructed that allows to learn spatio-temporal features of human motions. These works continued to be expanded by the same authors (Ullah and Petrosino, 2016) and achieved competitive results on some action benchmark datasets. Rahmani, Mian, and Shah, 2018 presented the "*Robust Non-Linear Knowledge Transfer Model (R-NKTM)*" – a deep fully-connected neural network that is capable of understanding human action from cross-views. The proposed framework learned the motion features from both dense trajectories of synthetic 3D human models and real motion capture data. FIGURE 3.16 illustrates the architecture of this framework. Experiments on cross-view human action datasets including IXMAS (Weinland, Ronfard, and Boyer, 2006), UWA3DII (Rahmani et al., 2016), Northwestern-UCLA Multiview Action3D dataset (Wang et al., 2014),

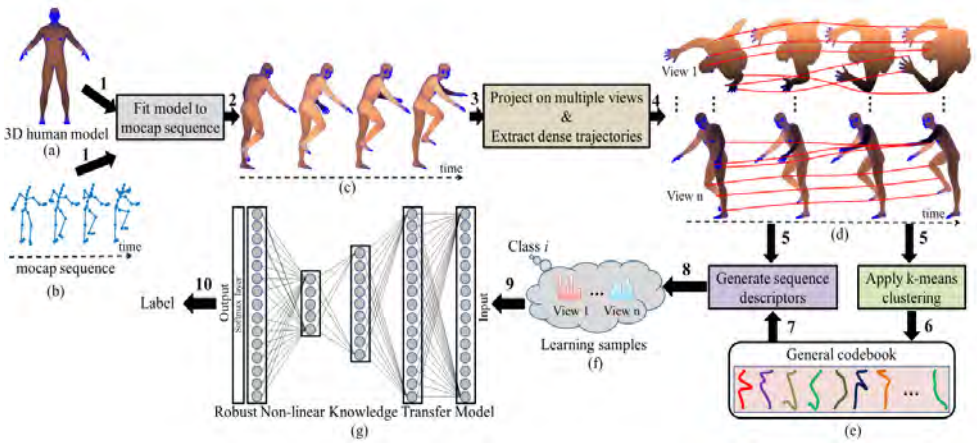
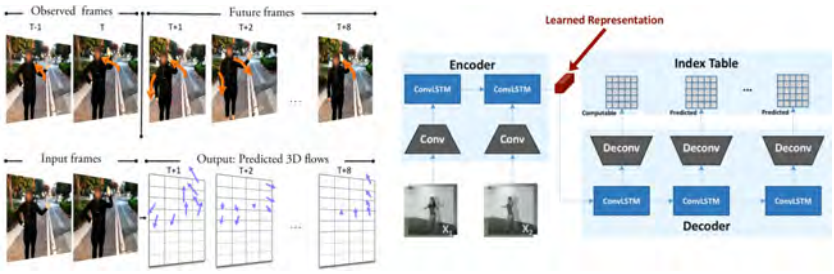


FIGURE 3.16: Illustration of the architecture of R-NKTM and its training process (Rahmani, Mian, and Shah, 2018). Firstly, 3D human models are fitted to real motion capture data for generating realistic 3D videos. These 3D videos are then projected on 2D planes to calculate dense trajectories. A general codebook is learned from trajectories that is then used as the input of R-NKTM. By this way, the R-NKTM can learn features of human motions and use them for recognizing actions.

and UCF Sports (Rodriguez, Ahmed, and Shah, 2008) have indicated that this architecture outperformed previous state-of-the-art approaches. In another study, Srivastava, Mansimov, and Salakhudinov, 2015 constructed a model which consists of two LSTMs, including an encoder LSTM and a decoder LSTM to learn representations of sequences of images. The state of the encoder is the representation of the input video. Then, the LSTM decoder will reconstruct the input sequence from this representation. It can be used for reconstructing the input sequences as well as predicting the future sequences. Very recently, Luo et al., 2017 combined many different models to build a deep learning framework for recognizing human motion in videos. The proposed deep architecture is able to predict the future 3D motions in videos (see FIGURE 3.17). Specifically, given input frames, the model predicts 3D flows in future frames, then uses these features to recognize actions. To do that, a RNN based Encoder-Decoder framework has been proposed. During the encoding process, VGG-16 networks (Simonyan and Zisserman, 2014b) were used to extract a low-dimensionality features from the input frames. Then, the LTSMs have been exploited to learn the temporal representations of motions. The learned representations were then decoded in the decoding process to generate the atomic 3D flows. This approach achieved the state-of-the-art accuracy rate on the NTU-RGB+D dataset (Shahroudy et al., 2016) and MSR Daily Activity3D (Li, Zhang, and Liu, 2010). Most recently, Lea et al., 2017 presented a class of time-series models, called Temporal Convolutional Networks (TCNs) for action recognition. The key advantage of TCNs is the ability to capture long-range patterns using a hierarchy of temporal convolutional filters. Experimental results showed that TCNs outperformed strong baselines including Bidirectional LSTM, whilst requiring less time to train. Kim and Reiter, 2017 also used TCNs to learn spatio-temporal representations for 3D human action recognition. Unlike the original architecture proposed by Lea et al., 2017, Kim and Reiter, 2017 redesigned TCNs by factoring out the deeper layers into additive residual terms which yields both interpretable hidden representations and model parameters. Experiments showed that the new design of TCN is able to produce discriminative spatio-temporal features for 3D human action analysis.



(a) (b)

FIGURE 3.17: **(a)** Illustration of the idea of learning a video representation by predicting a sequence of basic motions described as atomic 3D flows (Luo et al., 2017). The learned representations are then used for action recognition. **(b)** The proposed deep learning framework by Luo et al., 2017 using Recurrent Neural Network based Encoder-Decoder to predict the atomic 3D flows.

3.3 Discussion

In recent years, video-based human action recognition has become one of the most active research topics in computer vision. In particular, the appearance of the deep learning models as well as the advances of parallel computing techniques have opened up new opportunities for advancing the state-of-the-art in understanding human action in videos. Many deep learning-based approaches have been developed for various applications related to action recognition. In this section, we provide a detailed analysis of the mentioned architectures in this chapter. The pros and cons of each class and the link between them will also be discussed. Based on these analyses, we point out challenges, current trends and potential directions for future research in this field.

3.3.1 Current state of deep learning architectures for action recognition

Human action recognition in videos has advanced rapidly from the recognition of actions in controlled environment with small size benchmark datasets to the recognition of actions in realistic videos with very large-scale benchmarks. In this progress, deep learning algorithms have played an important role. In the literature of human action recognition based on deep learning, CNNs seem to be the most important model for learning spatio-temporal features of human action directly from videos, especially multi-stream CNN models. Some outstanding architectures have been proposed such as Ji et al., 2013, Tran et al., 2015, Simonyan and Zisserman, 2014a, Wang et al., 2017a, Feichtenhofer, Pinz, and Wildes, 2016, and Luo et al., 2017. The key ideas behind CNNs allow them to work directly on image structure and to obtain high-level features by composing lower-level ones. Not only working as an end-to-end solution, CNNs were also used as a feature extractor and were a part in another framework, in particular with RNN-LSTM networks. Although CNNs achieved excellent performances on various action recognition task, they require large datasets to optimize despite the fact that some techniques have been developed to prevent overfitting, *e.g.* Dropout (Srivastava et al., 2014) or data augmentation. In addition, training a very deep CNN architecture requires much computation. Therefore, applying very deep CNNs for human action recognition tasks is still a big challenge.

Recurrent Neural Networks with Long Short-Term Memory units (LSTM-RNNs) have been designed for solving time series problems. LSTM-RNNs have been used successfully in modeling the long-term context information of motion sequences, specifically with skeleton data as the works of Du, Wang, and Wang, 2015, Song et al., 2017, Zhu et al., 2016b, Li et al., 2016b, or Liu et al., 2016b. The success of LSTM-RNNs for human action recognition comes from their ability to take advantage of the entire history motion frames. Even so, most of LSTM-RNN based models cannot work directly on raw data. For example, skeleton data

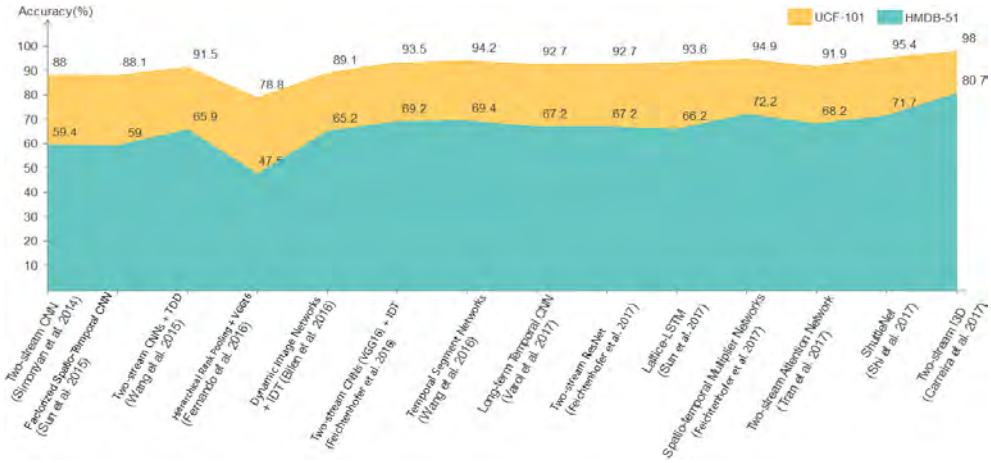


FIGURE 3.18: Comparison of action recognition accuracy (%) of different deep learning based approaches reported on the HMDB-51 (Kuehne et al., 2011) and UCF-101 (Soomro, Zamir, and Shah, 2012) datasets.

need to be preprocessed before feeding into LSTM-RNNs. The combination of CNNs and LSTMs is an excellent example of how we can build more powerful deep learning models by taking advantages of different architectures. In this case, CNNs were used to extract high-level motion features from videos, whilst LSTMs were exploited for sequences learning and prediction.

Deep Belief Network (DBNs) and Stacked Denoising Autoencoders (SDAs) are also promising choices for human action recognition tasks. For DBNs, these networks can be trained in an semi-supervised way with less labeled data. The limitation of DBNs is that, they require hand-crafted features (Foggia et al., 2014) or converting input data to appropriate form (Ali and Wang, 2014). Meanwhile, SDAs can learn motion features in unsupervised manner and are capable of generating robust features. However, they have several drawbacks related to their optimization process. In addition, action recognition approaches based on the recently introduced Generative Adversarial Networks (GANs) have also showed big potentials for learning and recognizing human actions in a semi-supervised manner, despite that they are difficult to train.

3.3.2 A quantitative analysis on HMDB-51, UCF-101 and NTU+RGB-D

To evaluate the learning performance of deep architectures in modeling and recognizing human actions, we provide a quantitative analysis of the different deep learning approaches on three state-of-the-art human action datasets, including HMDB-51 (Kuehne et al., 2011), UCF-101 (Soomro, Zamir, and Shah, 2012), and NTU-RGB+D (Shahroudy et al., 2016). We choose to analyze these benchmarks due to two main reasons. First, all these benchmarks are challenging, large-scale datasets for human action analysis. That allows us to evaluate state-of-the-art approaches. Second, these benchmarks have been used to evaluate the effectiveness of many different deep learning models, involving approaches based on CNNs, RNN-LSTMs, CNN-LSTMs and some other deep architectures.

Recognition performance on HMDB-51 and UCF-101 datasets

FIGURE 3.18 shows a comparison of the performance of many different deep learning architectures on the HMDB-51 (Kuehne et al., 2011) and UCF-101 (Soomro, Zamir, and Shah, 2012) datasets that have been reviewed in this chapter. In this comparison, accuracy rates

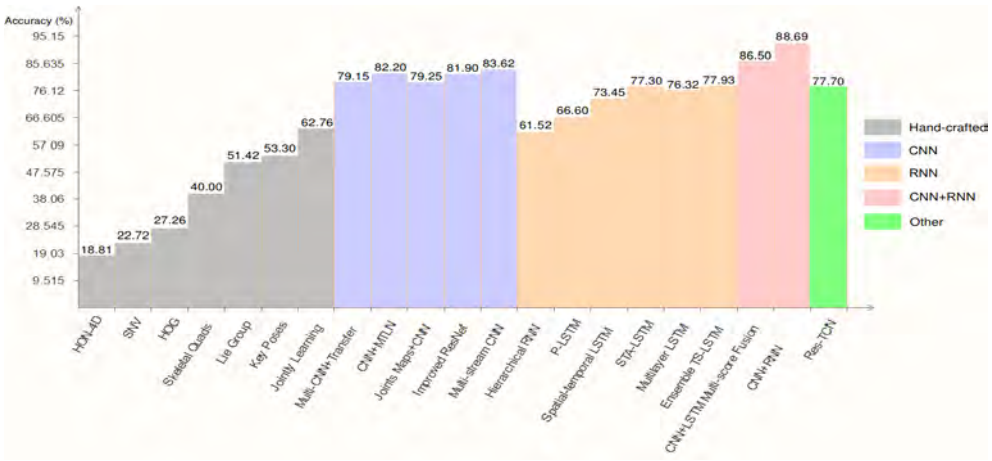


FIGURE 3.19: Comparison of action recognition accuracy (%) of different methods reported on the NTU-RGB+D dataset (Shahroudy et al., 2016).

are reported directly from the original papers and all of these works used the same evaluation protocol. The comparison results help us to identify the current state-of-the-art architectures proposed in the literature for action recognition. More specifically, we found that many approaches get similar recognition accuracy, but the multi-stream CNN based approaches proposed by Feichtenhofer, Pinz, and Zisserman, 2016; Wang et al., 2016a; Feichtenhofer, Pinz, and Wildes, 2017; Carreira and Zisserman, 2017 outperformed many other deep learning-based approaches. The best performing method on these datasets is currently the two-stream 3D-CNN, called "*Two-Stream Inflated 3D ConvNets (I3D)*" proposed by Carreira and Zisserman, 2017, with an accuracy of 80.7% on HMDB-51 (Kuehne et al., 2011) and 98.0% on UCF-101 (Soomro, Zamir, and Shah, 2012), respectively. The special deep architecture called shuttleNet (Shi et al., 2017) has also achieved an excellent performance, 71.7% on the HMDB-51 (Kuehne et al., 2011) and 95.4% on the UCF-101 (Soomro, Zamir, and Shah, 2012).

Recognition performance on NTU-RGB+D dataset

The NTU-RGB+D dataset (Shahroudy et al., 2016) is a very large-scale dataset for human action recognition (see APPENDIX A). To the best of our knowledge, this is currently the largest dataset that contains both RGB, depth and skeleton data. The NTU-RGB+D (Shahroudy et al., 2016) provides more than 56K videos, collected from 40 subjects and 80 viewpoints for 60 action classes. In order to evaluate the recognition performance on this benchmark, two different evaluation protocols have been suggested, including Cross-Subject evaluation and Cross-View evaluation. For the Cross-Subject evaluation, all sequences performed by 20 subjects are used for training and the rest are used for testing. For Cross-View evaluation, all sequences provided by cameras 2 and 3 are used for training while sequences from camera 1 are used for testing. TABLE 3.4 shows the classification accuracies of five groups of methods on this dataset, including hand-crafted based methods (Oreifej and Liu, 2013; Yang and Tian, 2014; Ohn-Bar and Trivedi, 2013; Evangelidis, Singh, and Horaud, 2014; Vemulapalli, Arrate, and Chellappa, 2014; Cippitelli et al., 2016a; Hu et al., 2015a), CNN-based methods (Misra, Zitnick, and Hebert, 2016; Rahmani and Bennamoun, 2017; Ke et al., 2017; Li et al., 2017a; Pham et al., 2018b; Liu, Liu, and Chen, 2017), RNN-based methods (Du, Wang, and Wang, 2015; Shahroudy et al., 2016; Luo et al., 2017; Liu et al., 2016b; Song et al., 2017; Zhang, Liu, and Xiao, 2017; Lee et al., 2017), CNN+RNN-based methods (Li et al., 2017b; Zhao, Ali, and Smagt, 2017) and some others (Shahroudy et al., 2017; Kim and Reiter, 2017). It is clear that the deep learning-based approaches outperformed many hand-crafted based approaches. In addition, CNN-based models works better than RNN-based models, while

the CNN+RNN-based frameworks seem the most powerful architectures for modeling human actions in videos. Specifically, we observe that the best performing method on the NTU-RGB+D (Shahroudy et al., 2016) is currently a deep framework based on a two-stream RNN+CNN model proposed by (Zhao, Ali, and Smagt, 2017). They achieved an accuracy of 83.74% on Cross-Subject settings and 93.65% on Cross-View settings, respectively.

3.3.3 The future of deep learning for video-based human action recognition

Unsupervised learning models for action recognition

Learning features directly from unknown videos in an unsupervised manner is a very important research direction (LeCun, Bengio, and Hinton, 2015) as labeled data is usually expensive and hard to obtain. The success of unsupervised learning methods based on DBNs or deep autoencoders such as the works of Foggia et al., 2014, Ballan et al., 2012, or Hasan and Roy-Chowdhury, 2014 has shown that unsupervised learning methods can achieve a high-level of performance without requiring labeled data or requiring very limited labeled data. Many other unsupervised representation learning approaches such as Misra, Zitnick, and Hebert, 2016, Rahmani, Mian, and Shah, 2018 or Luo et al., 2017 have also achieved impressive performances on realistic video datasets for action recognition. Due to big potentials in learning representations with unlabeled data, we expect that unsupervised learning approaches will continue to be extended for analyzing human actions in videos and advance the state-of-the-art of this field.

Deeper networks

The success of some very deep CNN architectures such as VGG-Net (Simonyan and Zisserman, 2014b), GoogLeNet (Szegedy et al., 2015a), and ResNet (Kaiming et al., 2016) provided that deeper models can boost the recognition performance. In particular, very deep architectures such as ResNet (Kaiming et al., 2016) and DenseNet (Huang et al., 2017) with shortcut connections in their architectures, are able to effectively learn visual features from data and prevent overfitting, whilst requiring less computation to achieve high performance for recognition tasks. Therefore, we expect very deep CNN architectures will be more fully exploited in this field. We also look forward to applications of Capsule Networks (Sabour, Frosst, and Hinton, 2017) – a new state-of-the-art CNN model for visual recognition in solving the action recognition related problems.

Combining different deep learning architectures

Taking full advantage of the different deep learning architectures and combining them into a single deep learning framework is a trend in action recognition. Specifically, the use of CNNs with LSTM-RNNs to model both the spatial and temporal information of human motions has improved the state-of-the-art in many human action datasets (Baccouche et al., 2011; Ng et al., 2015; Donahue et al., 2015; Sharma, Kiros, and Salakhutdinov, 2015; Ibrahim et al., 2016a; Singh et al., 2016; Li et al., 2016a; Wu et al., 2016; Wang et al., 2016e). We believe that this will be continued in the future, in which focusing on some very difficult topics such as online action recognition or early action prediction from streaming videos.

Fusion of hand-crafted features and deep learning

We also found that hand-crafted features such as the trajectory descriptors or optical flow based features have been used in most of state-of-the-art deep learning models as reported in the works of Varol, Laptev, and Schmid, 2018, Feichtenhofer, Pinz, and Zisserman, 2016, Tran et al., 2015, or Wang, Farhadi, and Gupta, 2016. We expect much of future progress in human action recognition to come from systems that exploit both hand-crafted and deep learning solutions to solve challenges of the field.

Transfer learning for human action recognition

One of the main difficulty in training deep networks comes from the scarcity of data. Supervised learning models such as CNNs require a lot of labeled data to optimize. In almost domains, acquiring and labeling for large-scale datasets are costly and time-consuming, in particular in the medical field. To solve this problem, some authors explored the transfer learning technique. In computer vision, many visual categories share low-level notations of edges, visual shapes, etc. Hence, instead of training an entire deep network from scratch, we can pretrain the network on a large-scale dataset such as ImageNet (Rahmani and Mian, 2016) or from existing datasets, and then use the network as an initialization for the task of interest. In action recognition, more and more promising results were achieved by deep learning models using transfer learning, *e.g.* Rahmani, Mian, and Shah, 2018, Liu et al., 2016a, Xu et al., 2016, Sargano et al., 2017, and recently by Rahmani and Bennamoun, 2017. Due to many advantages of transfer learning in video-based human action recognition, we believe that this research direction will continue to receive much attention from researchers for the next years.

New motion representations for deep learning-based action recognition from RGB-D data

Almost deep learning models are designed to learn and recognize human motions from RGB and optical flow, where they are fed directly into deep networks. This seems that has changed as more and more new motion representations have been proposed. For example, Scene Flow to Action Map (SFAM) by Wang et al., 2017b, human pose representation by Rahmani and Mian, 2016, or encoding skeleton sequences into color images such as the works of Liu, Liu, and Chen, 2017. We expect that the question: *"How to construct a robust representation that easily captures the spatio-temporal evolutions of motions from RGB-D data before feeding it to deep neural networks"*? will continue to be studied in the future.

3.4 Conclusion

We have seen in this chapter that in recent years, deep learning-based approaches have shown impressive performance and big potential in analyzing and recognizing human actions in videos. Our goal in carrying out this state of the art and analysis is to acquire a better knowledge on deep learning models applied to the recognition of human actions in videos. A comprehensive review of various deep learning architectures and their applications in action recognition have been provided based on more than 250 related publications. Our analysis and comparisons of the recognition performance of different deep learning-based approaches allow to identify state-of-the-art deep architectures for this task. In addition, the characteristics of the most important deep learning architectures for action recognition have been also analyzed to provide current trends and open problems for future works in this field. Moreover, we also provided in this chapter a list of action recognition datasets with different complexity levels. The main aim of the next chapter, which is the heart of the PhD work, is to explain how we use and adapt the previous frameworks described in chapter 2 and chapter 3. In particular, we devise several deep learning-based methodologies and representations that are evaluated on public benchmark datasets as well as on real-world dataset in the field of safety in public transport.

TABLE 3.4: Recognition performance on the NTU-RGB+D dataset (Shahroudy et al., 2016).

Feature	Method	Cross-Subject	Cross-View	Avg.
Hand-crafted	HON-4D (Oreifej and Liu, 2013, reported in Shahroudy et al., 2016)	30.56%	7.26%	18.81%
Hand-crafted	Super Normal Vector (Yang and Tian, 2014, reported in Shahroudy et al., 2016)	31.82%	13.61%	22.72%
Hand-crafted	HOG^2 (Ohn-Bar and Trivedi, 2013, reported in Shahroudy et al., 2016)	32.24%	22.27%	27.26 %
Hand-crafted	Skeletal Quads (Evangelidis, Singh, and Horaud, 2014, reported in Shahroudy et al., 2016)	38.62%	41.36%	40.00%
Hand-crafted	Lie Group (Vemulapalli, Arrate, and Chellappa, 2014, reported in Shahroudy et al., 2016)	50.08%	52.76%	51.42 %
Hand-crafted	Key Poses + SVM (Cippitelli et al., 2016a)	48.90%	57.70%	53.30%
Hand-crafted	Jointly Learning Features (Hu et al., 2015a, reported in Shahroudy et al., 2016)	60.23%	65.22%	62.76%
CNN	Shuffle and Learn (Misra, Zitnick, and Hebert, 2016, reported in Shahroudy et al., 2016)	47.52%	N/A	N/A
CNN	Multi-stream CNN + Transfer Learning (Rahmani and Bennamoun, 2017)	75.20%	83.10%	79.15%
CNN	Clips + CNN + MTLN (Ke et al., 2017)	79.57%	84.83%	82.20%
CNN	Joint Maps + CNN (Li et al., 2017a)	76.20%	82.30%	79.25%
CNN	Improved ResNet (Pham et al., 2018b)	78.20%	85.60%	81.90%
CNN	Multi-stream CNN (Liu, Liu, and Chen, 2017)	80.03%	87.21%	83.62%
RNN	Hierarchical RNN (Du, Wang, and Wang, 2015, reported in Shahroudy et al., 2016)	59.07%	63.97%	61.52%
RNN	P-LSTM (Shahroudy et al., 2016)	62.93%	70.27%	66.60%
RNN	RNN Encoder-Decoder (Luo et al., 2017)	66.20%	N/A	N/A
RNN	Spatio-temporal LSTM (Liu et al., 2016b)	69.20%	77.70%	73.45%
RNN	STA-LSTM (Song et al., 2017)	73.40%	81.20%	77.30%
RNN	Multilayer LSTM (Zhang, Liu, and Xiao, 2017)	70.26%	82.39%	76.32%
RNN	Ensemble TS-LSTM v2 LSTM (Lee et al., 2017)	74.60%	81.25%	77.93%
CNN+RNN	Multi-Score Fusion (LSTM+CNN - Li et al., 2017b)	82.89%	90.10%	86.50%
CNN+RNN	Two-Stream RNN+CNN (Zhao, Ali, and Smagt, 2017)	**83.74**%	**93.65**%	**88.69**%
Other	Res-TCN (Kim and Reiter, 2017)	74.30%	81.10%	77.70%
Other	DSSCA - SSLM (Shahroudy et al., 2017)	74.86%	N/A	N/A

Chapter 4

Proposed Deep Learning-based Approach for 3D Human Action Recognition from Skeletal Data Provided by RGB-D Sensors

Contents

Chapter overview: Depth sensors are able to provide detailed information about the 3D structure of the human movements using real-time skeleton estimation algorithms. This data source is a high-level representation allowing to describe human action in a more precise and effective way than that in RGB images. This is suitable for the problem of action analysis and recognition. However, designing motion representations for the problem of 3D human action recognition from skeleton sequences is still a challenging task. An effective representation should be robust to noise, invariant to viewpoint changes and result in a good performance with low-computational demand. Two main challenges in this task include how to efficiently represent spatio-temporal patterns of skeletal movements and how to learn their discriminative features for classification task. In this chapter, we propose novel skeleton-based representations for 3D human action recognition in videos using Deep Convolutional Neural Networks (D-CNNs). Two key issues have been addressed: First, how to build a robust representation that easily captures the spatio-temporal evolutions of motions from skeleton sequences. Second, how to design D-CNNs capable of learning discriminative features from the new representation in an effective manner. To address these tasks, we propose to encode the 3D coordinates of the human body joints carried by skeleton sequences to color images (Section 4.1). These color images are able to capture the spatio-temporal evolutions of skeletons and can be efficiently learned by D-CNNs. We then propose a deep learning architecture based on ResNets (Kaiming et al., 2016) to learn features from obtained color-based representations and classify them into action classes. Experimental results on the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014) and NTU-RGB+D (Shahroudy et al., 2016) datasets demonstrate that our method achieves state-of-the-art performance for all these benchmarks.

We then continue to investigate and expand the technique above. As a result, we introduce two new 3D skeleton-based representations, called SPMF ("*Skeleton Posture-Motion Feature*" – Section 4.2) and Enhanced-SPMF (Section 4.3). The SPMF and Enhanced-SPMF are compact image representations built from skeleton poses and their motions. The Enhanced-SPMF is an extension of the SPMF in which a smoothing filter and an Adaptive Histogram Equalization (AHE) algorithm were applied to reduce the effect of noise on skeletal data as well as to enhance their local patterns. For learning and classification tasks, we exploit state-of-the-art D-CNNs such as the Inception-ResNet-v2, DenseNet, and Neural Architecture Search (NAS) to learn directly an end-to-end mapping between input skeleton sequences and their action labels via the proposed representations. Our method is evaluated on four challenging benchmark datasets, including both individual actions (MSR Action3D – Wanqing, Zhengyou, and Zicheng, 2010, KARD – Gaglio, Re, and Morana, 2014), interactions (SBU Kinect Interaction – Yun et al., 2012a), and multiview & large-scale dataset (NTU-RGB+D dataset – Shahroudy et al., 2016). The experimental results demonstrate that the proposed method outperforms previous state-of-the-art approaches on all benchmark tasks.

4.1 Learning and recognizing 3D human actions from skeleton movements with Deep Residual Neural Networks

The computer vision community is currently focusing on solving action recognition problems in real videos, which contain thousands of samples with many challenges. In this process, D-CNNs have played a significant role in advancing the state-of-the-art in various vision-based action recognition systems. In 2016, the introduction of residual connections in conjunction with a more traditional CNN model in a single architecture called ResNet (Kaiming et al., 2016) has shown impressive performance and great potential for image recognition tasks. In this section, we investigate and apply ResNets for human action recognition using skeletal data provided by depth sensors. Firstly, the 3D coordinates of the human body joints carried in skeleton sequences are transformed into image-based representations as RGB images. These color images are able to capture the spatio-temporal evolutions of 3D motions from skeleton sequences and can be efficiently learned by D-CNNs. We then propose a novel deep learning architecture based on the ResNet (Kaiming et al.,

2016) to learn features from obtained color-based representations and classify them into action classes. The proposed method is evaluated on three challenging benchmark datasets including MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014), and NTU-RGB+D (Shahroudy et al., 2016) datasets. Experimental results demonstrate that our method achieves state-of-the-art performance for all these benchmarks. In particular, the proposed method surpasses previous approaches by a significant margin of 3.4% on MSR Action3D dataset, 0.67% on KARD dataset, and 2.5% on NTU-RGB+D dataset.

4.1.1 Introduction

Traditional studies on video-based human action recognition mainly focus on the use of hand-crafted local features such as Cuboids (Dollár et al., 2005) or HOG/HOF (Laptev et al., 2008) that are provided by 2D cameras. These approaches typically recognize human actions based on the appearance and movements of human body parts in videos. Another approach is to use Genetic Programming (GP) for generating spatio-temporal descriptors of motions (Liu, Shao, and Rockett, 2012). However, one of the major limitations of the 2D data is the absence of 3D structure from the scene. Therefore, single modality action recognition on RGB sequences is not enough to overcome the challenges in human action recognition, especially in realistic videos. Recently, the rapid development of depth-sensing time-of-flight camera technology has helped in dealing with problems, which are considered complex for traditional cameras. Depth cameras, *e.g.* Microsoft Kinect ^{TM}sensor (Cruz, Lucio, and Velho, 2012; Han et al., 2013) or ASUS Xtion (ASUS, 2018), are able to provide detailed information about the 3D structure of the human motion. Thus, many approaches have been proposed for recognizing actions based on RGB sequences, depth (Baek et al., 2016), or combining these two data types (Wang, Liu, and Wu, 2014), which are provided by depth sensors. Moreover, they are also able to provide real-time skeleton estimation algorithms (Shotton et al., 2011) that help to describe actions in a more precise and effective way. The skeleton-based representations have the advantage of lower dimensionality than RGB/RGB-D-based representations. This benefit makes action recognition systems become simpler and faster. Therefore, exploiting the 3D skeletal data provided by depth sensors for human action recognition is a promising research direction. In fact, many skeleton-based action recognition approaches have been proposed (Wang et al., 2012; Xia, Chen, and Aggarwal, 2012b; Chaudhry et al., 2013; Vemulapalli, Arrate, and Chellappa, 2014; Ding et al., 2016).

In recent years, approaches based on CNNs have achieved outstanding results in many image recognition tasks (Krizhevsky, Sutskever, and Hinton, 2012a; Karpathy et al., 2014). After the success of AlexNet (Krizhevsky, Sutskever, and Hinton, 2012a) in the ImageNet competition (Russakovsky et al., 2015), a new direction of research has been opened for finding higher performing CNN architectures. As a result, there are many signs that seem to indicate that the learning performance of CNNs can be significantly improved by increasing their depth (Simonyan and Zisserman, 2014b; Szegedy et al., 2015a; Telgarsky, 2016). In the literature of human action recognition, many studies have indicated that CNNs have the ability to learn complex motion features better than hand-crafted approaches (see FIGURE 4.1). However, most authors have just focused on the use of relatively small and simple CNNs such as AlexNet (Krizhevsky, Sutskever, and Hinton, 2012a) and have not yet fully exploited the potential of recent state-of-the-art very deep CNN architectures. In addition, most existing CNN-based approaches use RGB, depth or RGB-D sequences as the input to learning models. Although RGB-D images are informative for human action recognition, however, the computation complexity of these models will increase rapidly when the dimension of the input features is large. This makes models become more complex, slower and less practical for solving large-scale problems.

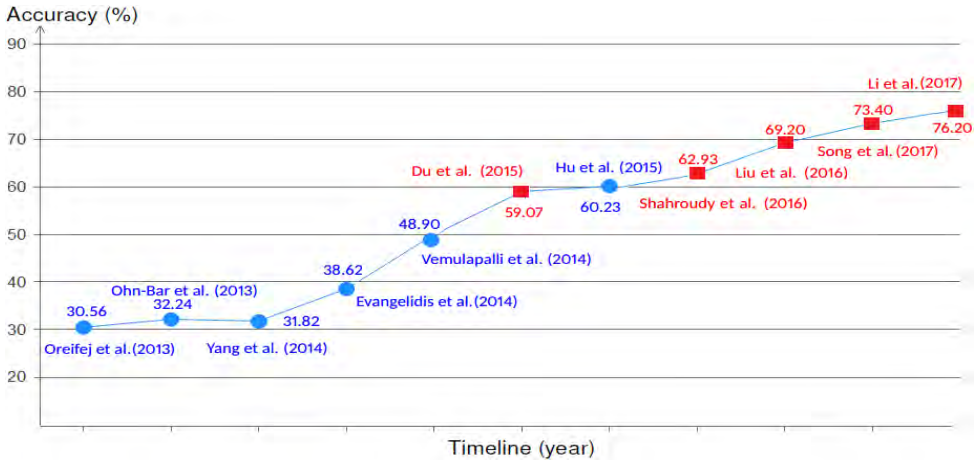


FIGURE 4.1: The recognition performance of hand-crafted and deep learning approaches reported on the Cross-View evaluation criteria of NTU-RGB+D dataset (Shahroudy et al., 2016). The traditional approaches are marked with circles (Oreifej and Liu, 2013; Ohn-Bar and Trivedi, 2013; Yang and Tian, 2014; Evangelidis, Singh, and Horaud, 2014; Hu et al., 2015b). The deep learning-based approaches are marked with squares (Du, Wang, and Wang, 2015; Shahroudy et al., 2016; Liu et al., 2016b; Song et al., 2017; Li et al., 2017a). Note that, this comparison was made in 2017.

In this work, we aim to take full advantages of 3D skeleton-based representations and the ability of learning highly hierarchical image features of D-CNNs to build an end-to-end learning framework for human action recognition from skeletal data. To this end, all the 3D coordinates of the skeletal joints in the body provided by Kinect sensors are represented as 3D arrays and then stored as RGB images by using a simple skeleton-to-image encoding method. The main goal of this processing step is to ensure that the color images effectively represents the spatio-temporal structure of the human action included in skeleton sequences and they are compatible with the deep learning networks as D-CNNs. To learn image features and recognize their labels, we propose to use ResNets (Kaiming et al., 2016) – a very deep and recent state-of-the-art CNN for image recognition. In the hope of achieving higher levels of performance, we propose a new deep architecture based on the original ResNet, which is easier to optimize and able to better prevent overfitting. We evaluate the proposed method on three benchmark skeleton datasets, MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), Kinect Activity Recognition Dataset - KARD (Gaglio, Re, and Morana, 2014), NTU-RGB+D (Shahroudy et al., 2016), and obtain state-of-the-art recognition accuracies on all these datasets. Furthermore, we also point out the effectiveness of our learning framework in terms of computational complexity, the ability to prevent overfitting and to reduce the effect of degradation phenomenon in training very deep networks.

Section 4.1 is organized as follows: Section 4.1.2 discusses related works. In section 4.1.3, we present the details of our proposed method. Datasets and experiments are described in Section 4.1.4. Experimental results are shown in Section 4.1.5. This section also discusses classification accuracy, overfitting issues, degradation phenomenon and computational efficiency of the proposed deep learning networks. Additionally, we also discuss about different factors that affect the recognition rate. Finally, a conclusion, which gives a synthesis of the main achievements is provided in Section 4.1.6 with a transition to the follow up of our research.

4.1.2 Related work

Our study is closely related to two major topics: skeleton-based action recognition and designing D-CNN architectures for visual recognition tasks. This section presents some key studies on these topics. We first discuss previous works on skeleton-based action recognition. Then, we introduce an overview of the development of D-CNNs and their potential

for human action recognition. Related to human action recognition based on RGB/RGB-D sequences, we refer the interested reader to the most successful approaches including Bag of Words (BoWs – Peng et al., 2016; Liu et al., 2017b; Wang and Schmid, 2013), Dynamic Image Networks (Bilen et al., 2016) and D-CNNs to learn RGB representation from raw data (Ng et al., 2015; Simonyan and Zisserman, 2014a).

Skeleton-based action recognition

Recent skeleton-based action recognition methods can be divided into two main groups. The first group combines hand-crafted skeleton features and graphical models to recognize actions. The spatio-temporal representations from skeleton sequences are often modeled by several common probabilistic graphical models such as Hidden Markov Model (HMM – Lv and Nevatia, 2006; Wang et al., 2012; Yang, Saleemi, and Shah, 2013), Latent Dirichlet Allocation (LDA – Blei, Ng, and Jordan, 2003; Liu, Shao, and Rockett, 2012) or Conditional Random Field (CRF – Koppula and Saxena, 2013). In addition, Fourier Temporal Pyramid (FTP – Wang et al., 2012; Vemulapalli, Arrate, and Chellappa, 2014; Hu et al., 2015b) has also been used to capture the temporal dynamics of actions and then to predict their labels. Another solution based on shape analysis methods has been exploited for skeleton-based human action (Devanne et al., 2013). Specifically, the authors defined an action as a sequence of skeletal shapes and analyzed them by a statistical shape analysis tool such as the geometry of Kendall's shape manifold. Typical classifiers, *e.g.* K-Nearest-Neighbor (KNN – Altman, 1992) or Support Vector Machine (SVM – Cortes and Vapnik, 1995) were then used for classification. Although promising results have been achieved, however, most of these works require a lot of feature engineering. *E.g.*, the skeleton sequences often need to be segmented and aligned for HMM- and CRF-based approaches. Meanwhile, FTP-based approaches cannot globally capture the temporal sequences of actions.

The second group of methods is based on Recurrent Neural Networks with Long Short-Term Memory Network (RNN-LSTMs – Hochreiter and Schmidhuber, 1997). The architecture of an RNN-LSTM network allows to store and access the long range contextual information of a temporal sequence. As human skeleton-based action recognition can be regarded as a time-series problem (Gong, Medioni, and Zhao, 2014), RNN-LSTMs can be used to learn human motion features from skeletal data. For that reason, many authors have explored RNN-LSTMs for 3D human action recognition from skeleton sequences (Du, Wang, and Wang, 2015; Veeriah, Zhuang, and Qi, 2015; Ling, Tian, and Li, 2016; Zhu et al., 2016b; Shahroudy et al., 2016; Liu et al., 2016b). To better capture the spatio-temporal dynamics carried in skeletons, some authors used a CNN as a visual feature extractor, combined with a RNN-LSTM in a unified framework for modeling human motions (Mahasseni and Todorovic, 2016; Shi and Kim, 2017; Kim and Reiter, 2017). Although RNN-LSTM-based approaches have been reported to provide good performance, there are some limitations that are difficult to overcome. The use of RNNs for instance can lead to overfitting problems when the number of input features is not enough for training the network. Meanwhile, the computational time can become a serious problem when the number of input features increases.

D-CNNs for visual recognition

Recently, there is growing evidence that D-CNN models can improve performance in image recognition (Simonyan and Zisserman, 2014b; Szegedy et al., 2015a). However, deep networks are very difficult to train. Two main reasons that impede the convergence of deeper networks are vanishing gradients problems (Glorot and Bengio, 2010) and degradation phenomenon (He and Sun, 2015). The vanishing gradients problem occurs when the network is deep enough, the error signal from the output layer can be completely attenuated on its way back to the input layer. This obstacle has been solved by normalized initialization (LeCun et al., 1998b; He et al., 2015), especially by using Batch Normalization (Ioffe and Szegedy, 2015). When the deep networks start converging, a degradation phenomenon occurs (see APPENDIX D). If we add more layers to a deep network, this can lead to higher training and/or testing error (He and Sun, 2015). This phenomenon is not as simple as an

overfitting problem. To reduce the effect of vanishing gradients problems and degradation phenomenon, Kaiming et al., 2016 introduced Residual Networks (ResNets) with the presence of shortcut connections parallel to their traditional convolutional layers. This idea helps ResNets to improve the information flow across layers. Experimental results on two well-known datasets including CIFAR-10 (Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015) confirmed that this architecture can improve the recognition performance and reduce degradation phenomenon.

Several authors have exploited the feature learning ability of CNNs on skeletal data (Hou et al., 2017; Wang et al., 2016c; Song et al., 2017; Li et al., 2017a). However, such studies mainly focus on finding good skeletal representations and learning features with simple CNN architectures. In contrast, in this work we concentrate on exploiting the power of D-CNNs for action recognition using a simple skeleton-based representation. We investigate and design a novel deep learning framework based on ResNet (Kaiming et al., 2016) to learn action features from skeleton sequences and then classify them into classes. Our experimental results show state-of-the-art performance on the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014) and NTU-RGB+D (Shahroudy et al., 2016) datasets. Besides, our proposed solution is general and can be applied on various different types of input data. For instance, this idea could be applied on the motion capture (MoCap) data provided by inertial sensors.

4.1.3 Proposed method

This section presents our proposed method. We first describe a technique allowing to encode the spatio-temporal information of skeleton sequences into RGB images. Then, a novel ResNet architecture is proposed for learning and recognizing actions from obtained RGB images. Before that, in order to put our method into context, it is useful to review the central ideas behind the original ResNet (Kaiming et al., 2016) architecture.

Encoding skeletal data into RGB images

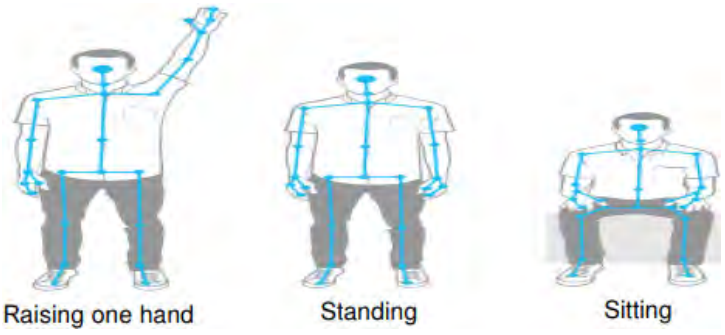


FIGURE 4.2: Illustration of the joint positions in the human body extracted by Kinect v2 sensor (Microsoft, 2014). A sequence of skeletons is able to describe correctly what a person performs in unseen videos.

Currently, the real-time skeleton estimation algorithms have been integrated into commercial depth cameras (Shotton et al., 2011). This technology allows to quickly and easily extract the position of the joints in the human body (FIGURE 4.2), which is suitable for the problem of 3D action recognition. One of the major challenges in exploiting CNN-based methods for skeleton-based action recognition is how a temporal skeleton sequence can be effectively represented and fed to CNNs for learning data features and perform classification. As CNNs are able to work well on images, the idea therefore is to encode the spatial and temporal dynamics of skeleton sequences into 2D image structures. In general, two essential elements for recognizing actions are static postures and their temporal dynamics. These two elements can be transformed into the static spatial structure of a color image (Bilen et al., 2016; Hou

et al., 2017; Wang et al., 2016c). Then, a representation learning model such as CNNs can be deployed to learn image features and classify them into classes in order to recognize the original skeleton sequences.

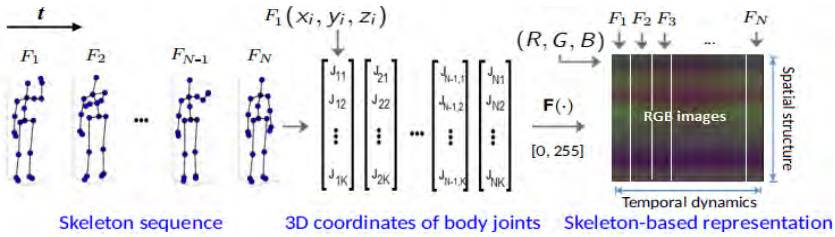


FIGURE 4.3: Illustration of the color encoding process. Here, N denotes the number of frames in each skeleton sequence. K denotes the number of joints in each frame. The value of K depends on the depth sensors and data acquisition settings.

Given a skeleton sequence $\mathcal{S}$ with N frames, denoting as $\mathcal{S} = \{F_1, F_2, ..., F_N\}$. To represent the spatio-temporal information of a skeleton sequence as an RGB image, we transform the 3D joint coordinates (x_i, y_i, z_i) carried in each skeleton $\{F_n\}$, $n \in [1, N]$ into the range of [0, 255] by normalizing these coordinates via a transformation function $\mathbf{F}(\cdot)$ as follows:

$$(x'_i, y'_i, z'_i) = \mathbf{F}(x_i, y_i, z_i) \tag{4.1}$$

$$x'_i = 255 \times \frac{(x_i - \min\{\mathcal{C}\})}{\max\{\mathcal{C}\} - \min\{\mathcal{C}\}} \tag{4.2}$$

$$y'_i = 255 \times \frac{(y_i - \min\{\mathcal{C}\})}{\max\{\mathcal{C}\} - \min\{\mathcal{C}\}} \tag{4.3}$$

$$z'_i = 255 \times \frac{(z_i - \min\{\mathcal{C}\})}{\max\{\mathcal{C}\} - \min\{\mathcal{C}\}} \tag{4.4}$$

where $\min\{\mathcal{C}\}$ and $\max\{\mathcal{C}\}$ are the maximum and minimum values of all coordinates over the training dataset, respectively. The new coordinate space is quantified as a digital image representation (8-bit: [0, 255]) and three coordinates (x'_i, y'_i, z'_i) are considered as the three components R, G, B of a color-pixel (x'_i = R; y'_i = G; z'_i = B). As shown in FIGURE 4.3, each skeleton sequence is encoded into an RGB image. By this transformation, the raw data of skeleton sequences are converted to 3D tensors, which will then be fed into the learning model as the input features.

The order of joints in each frame is non-homogeneous for many skeleton datasets. Thus, it is necessary to rearrange joints and find a better representations in which different actions can be easily distinguished by the learning model. In other words, the image-based representation needs to contain discriminative features – a key factor to ensure the success of the CNNs during the learning process. Naturally, the human body is structured by four limbs and one trunk. Simple actions can be performed through the movements of a limb while more complex actions come from the movements of a group of limbs in conjunction with the whole body. Inspired by this idea, Du, Wang, and Wang, 2015 proposed a simple and effective technique for representing skeleton sequences according to human body physical structure. To keep the local motion characteristics and to generate more discriminative features in image-based representations, we divide each skeleton frame into five parts, including two arms (P1, P2), two legs (P4, P5), and one trunk (P3). In each part from P1 to P5, the joints are concatenated according to their physical connections. We then rearrange these parts in a sequential order, *i.e.* P1 $\rightarrow$ P2 $\rightarrow$ P3 $\rightarrow$ P4 $\rightarrow$ P5. The whole process of rearranging all frames in a sequence can be done by rearranging the order of the rows of pixels in RGB-based representations as illustrated in FIGURE 4.4. Some skeleton-based representations obtained from the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010) are shown in FIGURE 4.5.

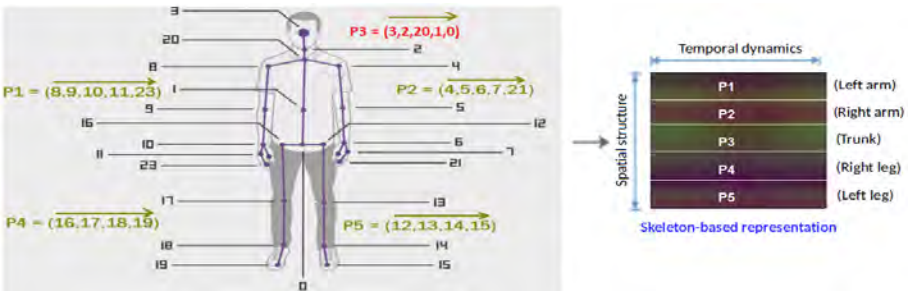


FIGURE 4.4: Arranging pixels in RGB images according to the human body physical structure.

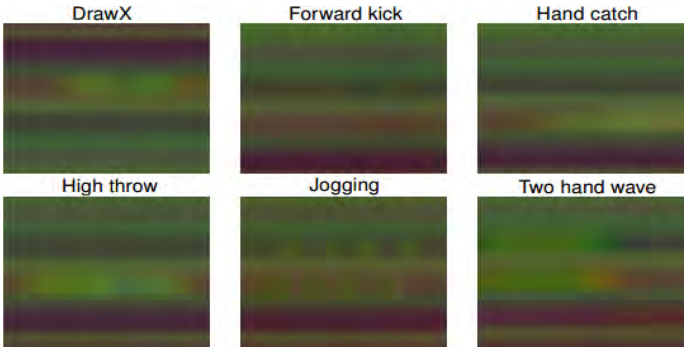


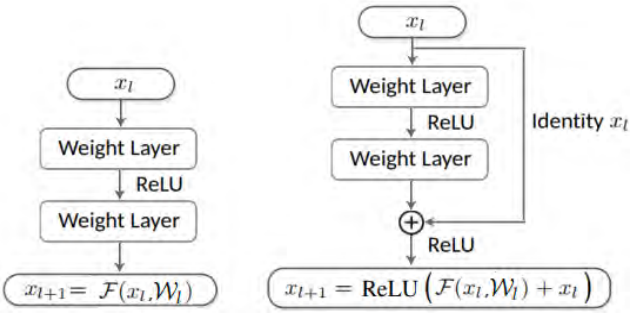
FIGURE 4.5: Output of the encoding process obtained from some samples of the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010). In our experiments, all images were resized to 32×32 pixels before feeding into the deep learning networks. Best viewed in color.

Deep Residual Network (ResNet)

A simple difference between ResNets and traditional CNNs is that ResNets provide a clear path for gradients to back propagate to early layers during training. A deep ResNet is a modularized architecture that is constructed from multiple ResNet building units. Each unit has shortcut connection in parallel with traditional convolutional layers, which connects the input feature directly to its output. Considering the input feature of the l^{th} layer as x_l, traditional CNNs (FIGURE 4.6a) learn a mapping function: $x_{l+1} = \mathcal{F}(x_l, \mathcal{W}_l)$, where x_{l+1} is the output of the l^{th} layer and $\mathcal{W}_l$ is a set of weights and biases associated with the l^{th} ResNet unit. $\mathcal{F}(\cdot)$ is a non-linear transformation that can be implemented by the combination of Batch Normalization (BN) (Ioffe and Szegedy, 2015), Rectified Linear Units (ReLU) (Nair and Hinton, 2010) and Convolutions. Different from traditional CNNs, a ResNet building unit (FIGURE 4.6b) performs the following computations:

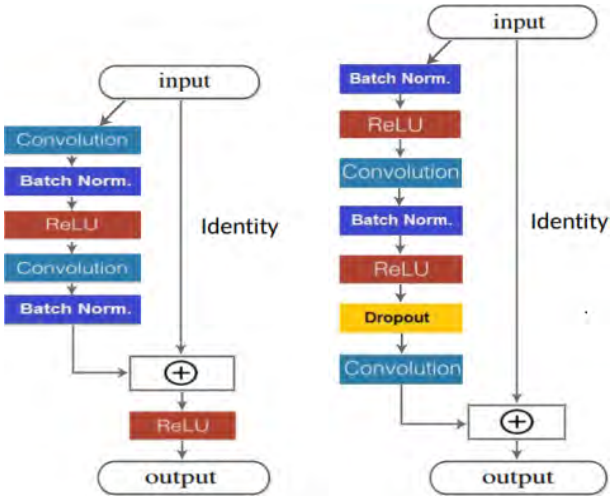
$$x_{l+1} = \text{ReLU}\left(\mathcal{F}(x_l, \mathcal{W}_l) + id(x_l)\right) \tag{4.5}$$

where x_l and x_{l+1} are input and output features of the l^{th} ResNet unit, respectively; $id(x)$ is the identity function $id(x_l) = x_l$. The detailed architecture of an original ResNet unit is shown in FIGURE 4.7a. In this architecture, $\mathcal{F}(\cdot)$ consists of a series of layers: *Convolution-BN-ReLU-Convolution-BN*. The ReLU (Nair and Hinton, 2010) is applied after each element-wise addition $\oplus$.



(a) (b)

FIGURE 4.6: **(a)** Information flow executed by a traditional CNN; **(b)** Information flow executed by a ResNet building unit (Kaiming et al., 2016).



(a) (b)

FIGURE 4.7: **(a)** A ResNet building unit that was proposed in the original paper (Kaiming et al., 2016); **(b)** Our proposed ResNet building. The symbol ⊕ denotes element-wise addition.

An improved ResNet for skeleton-based action recognition in videos

The original ResNet architecture has a direct path for propagating information within a residual unit. However, the presence of non-linear activations as ReLUs (Nair and Hinton, 2010) behind element-wise additions ⊕ (see FIGURE 4.7a) means that the signal cannot be directly propagated from one block to any other block. To solve this problem, we propose an improved ResNet building block in which the signal can be directly propagated from any unit to another, both forward and backward for the entire network. The idea is to replace ReLU layers after each element-wise addition ⊕ by identity mappings $id(\cdot)$ for all units. That way, the information flow across each new ResNet unit can be rewritten as:

$$x_{l+1} = id(y_l) = \mathcal{F}(x_l, \mathcal{W}_l) + x_l. \tag{4.6}$$

Eqn. (4.6) suggests that the feature x_L of any deeper unit L can be represented according to the feature x_l of any shallower unit l:

$$x_L = x_l + \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i). \tag{4.7}$$

Also, the feature x_L can be represented according to the input feature x_0 of the first ResNet unit:

$$x_L = x_0 + \sum_{i=0}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i). \tag{4.8}$$

Eqn. (4.8) indicates that we have created a "*direct path*" that helps the signal to be directly propagated in forward pass through the entire network. Considering now the backpropagation information, let $\mathcal{L}$ be the loss function that the network needs to optimize during the supervised training stage. From the chain rule of backpropagation (LeCun et al., 1989b) and Eqn. (4.7), we can express the backpropagation information through layers as:

$$\frac{\partial \mathcal{L}}{\partial x_l} = \frac{\partial \mathcal{L}}{\partial x_L}\frac{\partial x_L}{\partial x_l} = \frac{\partial \mathcal{L}}{\partial x_L}\frac{\partial \left(x_l + \sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i)\right)}{\partial x_l} \tag{4.9}$$

or:

$$\frac{\partial \mathcal{L}}{\partial x_l} = \frac{\partial \mathcal{L}}{\partial x_L}\left(1 + \frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i)\right) \tag{4.10}$$

The gradient $\frac{\partial \mathcal{L}}{\partial x_l}$ depends on two elements $\frac{\partial \mathcal{L}}{\partial x_L}$ and $\frac{\partial \mathcal{L}}{\partial x_L}\left(\frac{\partial}{\partial x_l}\sum_{i=l}^{L-1} \mathcal{F}(x_i, \mathcal{W}_i)\right)$, in which the term $\frac{\partial \mathcal{L}}{\partial x_L}$ is independent of any weight layers. This additive term ensures that the information flow can be easily propagated back from any deeper unit L to any shallower unit l. Based on the above analyses, it can be concluded that if we replace ReLU layers after element-wise additions by identity mappings, each ResNet unit will have a direct path to the gradients from the loss function and to the input signal. In other words, the information flow can be directly propagated from any unit to another, both forward and backward for the entire network.

To implement the computations as described in Eqn. (4.6), we remove all ReLU layers behind element-wise additions $\oplus$. In addition, BN is used before each convolutional layer, ReLU is adopted right after BN. This order allows to improve regularization of the network. Dropout (Hinton et al., 2012) with a rate of 0.5 is used to prevent overfitting and located between two convolutional layers. With this architecture, the mapping function $\mathcal{F}(\cdot)$ is executed via a sequence of layers: *BN-ReLU-Convolution-BN-ReLU-Dropout-Convolution* as shown in FIGURE 4.7b.

4.1.4 Experiments

In this section, we experiment the proposed method on three 3D skeleton datasets. We first present the datasets and their evaluation criteria. Some data augmentation techniques that are used for generating more training data are then described. Finally, implementation details of our deep networks are provided.

Datasets and evaluation criteria

In this section, we evaluate the proposed deep learning framework on MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014) and NTU-RGB+D (Shahroudy et al., 2016). For each dataset, we follow the same evaluation criteria as provided in the original papers. For the interested reader, some public RGB-D datasets for human action recognition can be found in recent surveys (Zhang et al., 2016; Liu et al., 2017a).

MSR Action3D dataset: The MSR Action3D dataset[1] (Wanqing, Zhengyou, and Zicheng, 2010) consists of 20 different action classes. Each action is performed by 10 subjects for three times. There are 567 skeleton sequences in total. However, 10 sequences are not valid since the skeletons were either missing. Therefore, our experiment was conducted on 557 sequences. For each skeleton frame, the 3D coordinates of 20 joints are provided. The authors of this dataset suggested dividing the whole dataset into three subsets, named AS1, AS2, and AS3. The list of actions for each subset is shown in TABLE 4.1. For each subset, we follow the cross-subject evaluation method used by many other authors working with this dataset. More specifically, a half of the dataset (subjects with IDs: 1, 3, 5, 7, 9) is selected for training and the rest (subjects with IDs: 2, 4, 6, 8, 10) for test.

TABLE 4.1: The list of actions in three subsets AS1, AS2, and AS3 of the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010).

AS1	**AS2**	**AS3**
Horizontal arm wave	*High arm wave*	*High throw*
Hammer	*Hand catch*	*Forward kick*
Forward punch	*Draw X*	*Side kick*
High throw	*Draw tick*	*Jogging*
Hand clap	*Draw circle*	*Tennis swing*
Bend	*Two hand wave*	*Tennis serve*
Tennis serve	*Forward kick*	*Golf swing*
Pickup & Throw	*Side-boxing*	*Pickup & Throw*

Kinect Activity Recognition Dataset (KARD): This dataset[2] (Gaglio, Re, and Morana, 2014) contains 18 actions, performed by 10 subjects and each subject repeated each action three times. KARD provides 540 skeleton sequences in total. Each frame comprises 15 joints. The authors suggested the different evaluation methods on this dataset in which the whole dataset is divided into three subsets as shown in TABLE 4.2. For each subset, three experiments have been proposed. Experiment A uses one-third of the dataset for training and the rest for test. Meanwhile, experiment B uses two-third of the dataset for training and one-third for test. Finally, experiment C uses a half of the dataset for training and the remainder for testing.

TABLE 4.2: The list of action classes in each subset of the KARD dataset (Gaglio, Re, and Morana, 2014).

Action Set 1	**Action Set 2**	**Action Set 3**
Horizontal arm wave	*High arm wave*	*Draw tick*
Two-hand wave	*Side kick*	*Drink*
Bend	*Catch cap*	*Sit down*
Phone call	*Draw tick*	*Phone call*
Stand up	*Hand clap*	*Take umbrella*
Forward kick	*Forward kick*	*Toss paper*
Draw X	*Bend*	*High throw*
Walk	*Sit down*	*Horiz. arm wave*

NTU-RGB+D Action Recognition Dataset: The NTU-RGB+D[3] (Shahroudy et al., 2016) is a very large-scale dataset. To the best of our knowledge, this is the largest and state-of-the-art RGB-D/skeleton dataset for human action recognition currently available. It provides

[1]The MSR Action3D dataset can be obtained at: https://www.uow.edu.au/~wanqing/#Datasets.

[2]The KARD dataset can be obtained at: https://data.mendeley.com/datasets/k28dtm7tr6/1.

[3]The NTU-RGB+D dataset can be obtained at: http://rose1.ntu.edu.sg/Datasets/actionRecognition.asp with authorization.

FIGURE 4.8: Some action classes of the NTU-RGB+D dataset (Shahroudy et al., 2016).

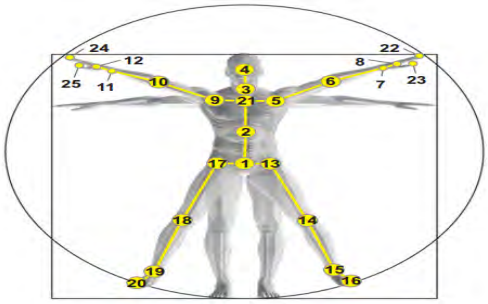


FIGURE 4.9: Configuration of 25 body joints in each frame of NTU-RGB+D dataset (Shahroudy et al., 2016).

more than 56 thousand video samples and 4 million frames, collected from 40 distinct subjects for 60 different action classes. FIGURE 4.8 shows some action classes of this dataset. The 3D skeletal data contains the 3D coordinates of 25 major body joints (FIGURE 4.9) provided by Kinect v2 sensor. Therefore, its skeletal data describes more accurately about human movements. The author of this dataset suggested two different evaluation criteria including Cross-Subject and Cross-View. For Cross-Subject evaluation, the sequences performed by 20 subjects with IDs: 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, and 38 are used for training and the rest sequences are used as testing data. In the Cross-View evaluation, the sequences provided by cameras 2 and 3 are used for training while sequences from camera 1 are used for test. The complete list of actions of the NTU-RGB+D (Shahroudy et al., 2016) is provided in APPENDIX A.

Data augmentation techniques

Very deep neural networks require a lot of data to train. Unfortunately, we have only 557 skeleton sequences on MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010) and 540 sequences on KARD (Gaglio, Re, and Morana, 2014). Thus, to prevent overfitting, some data augmentation techniques have been applied. The random cropping, flip horizontally and vertically techniques were used to generate more training samples. Specifically, 8-times cropping has been applied on 40×40 images to create 32×32 images. Then, their horizontally and vertically flipped images are created. For the NTU-RGB+D dataset (Shahroudy et al., 2016), due to the very large-scale of this dataset, data augmentation techniques were not applied.

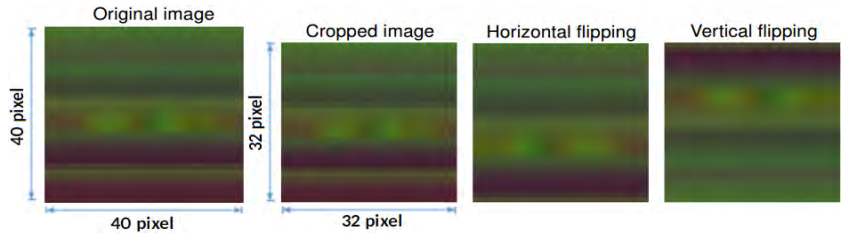


FIGURE 4.10: Data augmentation applied on MSR Action3D dataset.

Implementation details

Different configurations of ResNet with 20-layers, 32-layers, 44-layers, 56-layers, and 110-layers were designed, based on the original Resnet (Kaiming et al., 2016) building unit (FIGURE 4.7a) and the proposed ResNet building unit (FIGURE 4.7b). In total, we have ten different ResNets. The baseline of the proposed architectures can be found in APPENDIX B1. All networks are designed for the acceptable images with the size of 32 $\times$ 32 pixels as input features and classifying them into n categories corresponding to n action classes in each dataset. In the experiments, we use a mini-batch of size 128 for 20-layer, 32-layer, 44-layer, and 56-layer networks and a mini-batch of size 64 for 110-layer networks. We initialize the weights randomly and train all networks in an end-to-end manner using Stochastic Gradient Descent (SGD) algorithm (Bottou, 2010) for 200 epochs from scratch. The learning rate starts from 0.01 for the first 75 epochs, 0.001 for the next 75 epochs and 0.0001 for the remaining 50 epochs. The weight decay is set at 0.0001 and the momentum at 0.9. In this work, MatConvNet[4] (Vedaldi and Lenc, 2015) is used to implement the solution. Our code and models are shared with the community at: `https://github.com/huyhieupham/Improved-ResNet-Action-Recognition-Skeletal-Data`.

4.1.5 Experimental results and analysis

This section reports our experimental results. To show the effectiveness of the proposed method, the achieved results are compared with the state-of-the-art methods in the literature. All these comparisons are made following the same evaluation criteria.

Results on MSR Action3D dataset

The experimental results on MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010) are shown in TABLE 4.3. We achieved the best classification accuracy with the 44-layer ResNet model which is constructed from the proposed ResNet building unit. Specifically, classification accuracies are 99.9% on AS1, 99.8% on AS2, and 100% on AS3. We obtained a total average accuracy of 99.9%. TABLE 4.5 compares the performance between our best result with the state-of-the-art methods reported on this benchmark. This comparison indicates that the proposed model outperforms many prior works, in which we improved the accuracy rate by 3.4% compared to the best previous published results. FIGURE 4.11a and FIGURE 4.11b show the learning curves of all networks on AS1 subset.

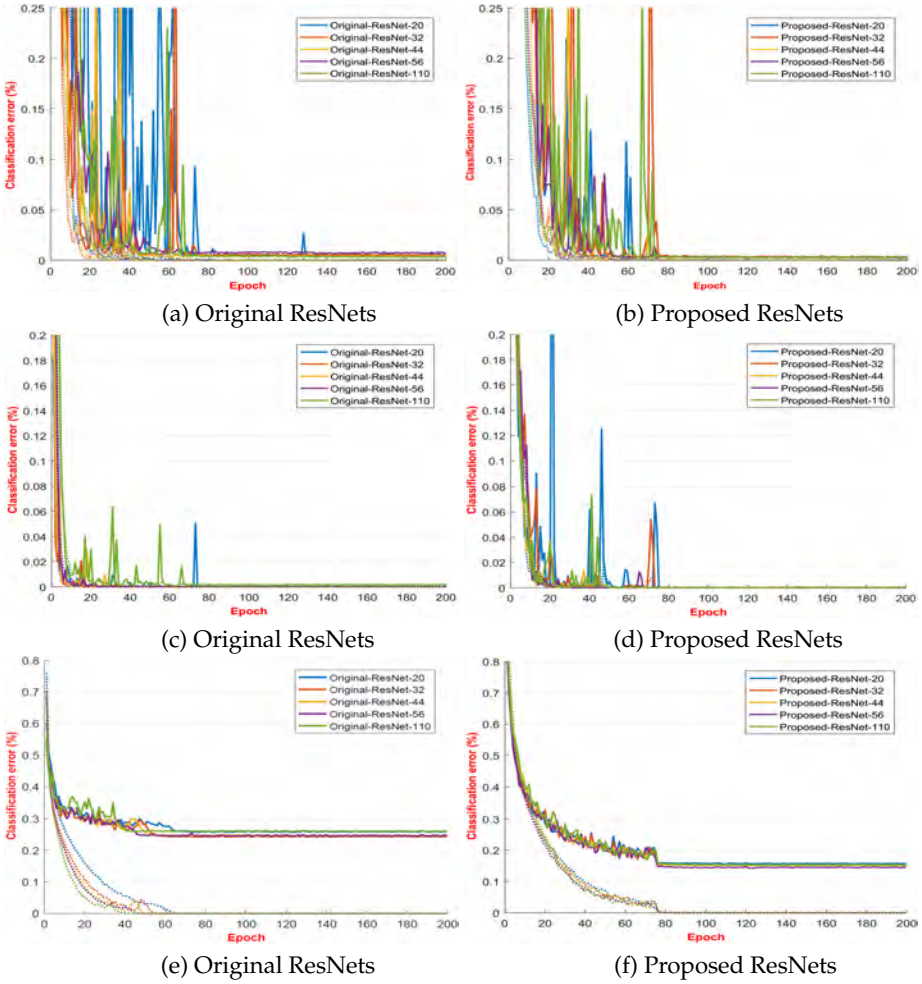
Results on KARD dataset

The experimental results on KARD dataset (Gaglio, Re, and Morana, 2014) are reported in TABLE 4.4. It can be observed the same learning behavior as experiments on MSR Action3D dataset, in which the best results are achieved by the proposed 44-layer ResNet model. TABLE 4.6 provides the accuracy comparison between this model and other approaches on the whole KARD dataset. Based on these comparisons, it can be concluded that our approach

[4]MatconvNet is an open source library for implementing Convolutional Neural Networks in the Matlab environment and can be downloaded at address: `http://www.vlfeat.org/matconvnet/`.

TABLE 4.3: Recognition accuracy obtained by the proposed method on AS1, AS2, and AS3 subsets of the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010).

Model	AS1	AS2	AS3	Aver.
Original-ResNet-20	99.5%	98.6%	99.9%	99.33%
Original-ResNet-32	99.5%	99.1%	99.9%	99.50%
Original-ResNet-44	99.6%	98.5%	100%	99.37%
Original-ResNet-56	99.3%	98.4%	99.5%	99.07%
Original-ResNet-110	99.7%	99.2%	99.8%	99.57%
Proposed-ResNet-20	99.8%	99.4%	100%	99.73%
Proposed-ResNet-32	99.8%	99.8%	100%	99.87%
Proposed-ResNet-44	**99.9%**	**99.8%**	**100%**	**99.90%**
Proposed-ResNet-56	99.9%	99.1%	99.6%	99.53%
Proposed-ResNet-110	99.9%	99.5%	100%	99.80%



(a) Original ResNets (b) Proposed ResNets

(c) Original ResNets (d) Proposed ResNets

(e) Original ResNets (f) Proposed ResNets

FIGURE 4.11: (**a**) and (**b**): Learning curves on MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010); (**c**) and (**d**): KARD (Gaglio, Re, and Morana, 2014); (**e**) and (**f**): NTU-RGB+D (Shahroudy et al., 2016) datasets. Dashed lines denote training errors, bold lines denote test errors. We recommend the reader to use a computer and zoom in to see these figures.

TABLE 4.4: Recognition accuracy obtained by the proposed method on KARD dataset (Gaglio, Re, and Morana, 2014).

	Activity Set 1			Activity Set 2			Activity Set 3		
Model	**A**	**B**	**C**	**A**	**B**	**C**	**A**	**B**	**C**
Original-Resnet-20	100	100	100	100	100	100	99.8	100	99.8
Original-ResNet-32	100	100	100	100	100	100	99.8	99.9	99.8
Original-ResNet-44	100	100	100	100	100	100	99.7	99.7	99.7
Original-ResNet-56	99.9	100	100	100	100	99.9	99.5	99.9	99.8
Original-ResNet-110	99.8	100	99.8	99.9	100	99.9	99.3	100	99.7
Proposed-Resnet-20	100	100	100	100	100	100	99.8	100	99.9
Proposed-ResNet-32	100	100	100	100	100	100	99.8	99.9	99.8
Proposed-ResNet-44	**100**	**100**	**100**	**100**	**100**	**100**	**99.9**	**99.9**	**100**
Proposed-ResNet-56	100	100	100	100	100	100	99.7	100	99.8
Proposed-ResNet-110	99.9	100	99.9	100	100	100	99.7	100	99.8

TABLE 4.5: Comparing our best performance (Proposed-ResNet-44) with other approaches on the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010). All methods use the same experimental protocol.

Method	**AS1**	**AS2**	**AS3**	**Aver.**
Bag of 3D Points (Wanqing, Zhengyou, and Zicheng, 2010)	72.90%	71.90%	79.20%	74.67%
Space-time (Vieira et al., 2012)	84.70%	81.30%	88.40%	84.80%
Histograms of 3D Joints(Xia, Chen, and Aggarwal, 2012a)	87.98%	85.48%	63.46%	78.97%
Silhouette-Skeletal Data (Chaaraoui, Padilla-Lopez, and Florez-Revuelta, 2013)	92.38%	86.61%	96.40%	91.80%
Depth Motion Maps (Chen, Liu, and Kehtarnavaz, 2013)	96.20%	83.20%	92.00%	90.47%
Group Sparsity (Luo, Wang, and Qi, 2013)	97.20%	95.50%	99.10%	97.26%
HOD 3D (Gowayyed et al., 2013)	92.39%	90.18%	91.43%	91.26%
Joint Locations (Hussein et al., 2013)	88.04%	89.29%	94.29%	90.53%
Depth + Shape Features (Qin, Yang, and Jiang, 2013)	81.00%	79.00%	82.00%	80.66%
3D Motion Trail (Liang and Zheng, 2013)	73.70%	81.50%	81.60%	78.93%
Skeletal Quads (Evangelidis, Singh, and Horaud, 2014)	88.39%	86.61%	94.59%	89.86%
Pose-based Recognition (Theodorakopoulos et al., 2014)	91.23%	90.09%	99.50%	93.61%
Multi-modality (Gao et al., 2014)	92.00%	85.00%	93.00%	90.00%
Space-Time Depth Map (Vieira et al., 2014)	91.70%	72.20%	98.60%	87.50%
Depth Motion Maps (Chen, Jafari, and Kehtarnavaz, 2015)	98.10%	92.00%	94.60%	94.90%
Hierarchical RNN (Du, Wang, and Wang, 2015)	93.33%	94.64%	95.50%	94.49%
S-T Pyramid (Xu et al., 2015a)	99.10%	92.90%	96.40%	96.10%
Vague Division Depth Maps (Jin et al., 2017)	99.10%	92.30%	98.20%	96.50%
Our best model	**99.90**%	**99.80**%	**100**%	**99.90**%

TABLE 4.6: Average recognition accuracy of the best proposed model (Proposed-ResNet-44) for experiments A, B and C compared to other approaches on the whole KARD dataset (Gaglio, Re, and Morana, 2014).

Method	Exp. A	Exp. B	Exp. C	Aver.
Hand-crafted Feature (Gaglio, Re, and Morana, 2014)	89.73%	94.50%	88.27%	90.83%
Key Feature + Multi-class SVM (Cippitelli et al., 2016b)	96.47%	98.27%	96.87%	97.20%
Key Postures + Multi-class SVM(Ling, Tian, and Li, 2016)	98.90%	99.60%	99.43%	99.31%
Our best model	**99.97**%	**99.97**%	**100**%	**99.98**%

TABLE 4.7: Recognition accuracy on NTU-RGB+D dataset (Shahroudy et al., 2016) for Cross-Subject and Cross-View evaluations.

Model	Cross-Subject	Cross-View
Original-ResNet-20	73.90%	80.80%
Original-ResNet-32	75.40%	81.60%
Original-ResNet-44	75.20%	81.50%
Original-ResNet-56	75.00%	81.50%
Original-ResNet-110	73.80%	80.00%
Proposed-ResNet-20	76.80%	83.80%
Proposed-ResNet-32	76.70%	84.70%
Proposed-ResNet-44	77.20%	84.80%
Proposed-ResNet-56	**78.20**%	**85.60**%
Proposed-ResNet-110	78.00%	84.60%

outperformed the prior state-of-the-art on this benchmark. FIGURE 4.11c and FIGURE 4.11d show the learning curves of all networks on Activity Set 1 subset for Experiment C.

Results on NTU-RGB+D dataset

TABLE 4.7 shows the experimental results on NTU-RGB+D dataset (Shahroudy et al., 2016). The best network achieved an accuracy of 78.2% on the Cross-Subject evaluation and 85.6% on the Cross-View, respectively. The performance comparison between the proposed method and the state-of-the-art methods on these two evaluations are provided in TABLE 4.8. These results showed that our proposed method can deal with very large-scale datasets and outperforms various state-of-the-art approaches for both evaluations. Comparing with the best published result reported by Li et al., 2017a for the Cross-Subject evaluation, our method significantly surpassed this result by a margin of +2.0%. For the Cross-View evaluation, we outperformed the state-of-the-art accuracy in Kim and Reiter, 2017 by a margin of +2.5%. FIGURE 4.11e and FIGURE 4.11f show the learning curves of all networks in these experiments.

Discussion of the results

The results are discussed here following three main criteria: classification accuracy, overfitting, effect of image resizing methods, effect of joint order, and computational efficiency.

Classification accuracy: In section 4.1.4, we evaluated the proposed learning framework on three well-known benchmark datasets. We demonstrate empirically that our method outperforms many previous studies on all these datasets under the same experimental protocols. The improvements on each benchmark are shown in TABLE 4.9. It is clear that in terms of accuracy, our learning model is effective for solving the problems of human action recognition.

Overfitting issues and degradation phenomenon: Considering the accuracies obtained by our proposed ResNet architecture and comparing them to the results obtained by the

TABLE 4.8: Performance comparison of our proposed ResNet models with the state-of-the-art methods on the Cross-Subject evaluation criteria of NTU-RGB+D dataset (Shahroudy et al., 2016).

Method (protocol of Shahroudy et al., 2016)	**Cross-Subject**	**Cross-View**
HON4D (Oreifej and Liu, 2013)	30.56%	7.26%
Super Normal Vector (Yang and Tian, 2014)	31.82%	13.61%
HOG^2 (Ohn-Bar and Trivedi, 2013)	32.24%	22.27%
Skeletal Quads (Evangelidis, Singh, and Horaud, 2014)	38.62%	41.36%
Shuffle and Learn (Misra, Zitnick, and Hebert, 2016)	47.50%	N/A
Key poses + SVM (Cippitelli et al., 2016a)	48.90%	N/A
Lie Group (Vemulapalli, Arrate, and Chellappa, 2014)	50.08%	52.76%
HBRNN-L (Du, Wang, and Wang, 2015)	59.07%	63.97%
FTP Dynamic Skeletons (Hu et al., 2015b)	60.23%	65.22%
P-LSTM (Shahroudy et al., 2016)	62.93%	70.27%
RNN Encoder-Decoder (Luo et al., 2017)	66.20%	N/A
ST-LSTM (Liu et al., 2016b)	69.20%	77.7%
STA-LSTM (Song et al., 2017)	73.40%	81.2%
Res-TCN (Kim and Reiter, 2017)	74.30%	83.1%
DSSCA - SSLM (Shahroudy et al., 2017)	74.86%	N/A
Joint Distance Maps + CNN (Li et al., 2017a)	76.20%	N/A%
Our best model (Proposed-ResNet-56)	**78.20**%	**85.60**%

TABLE 4.9: The best of our results compared to the best prior results on MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014), and NTU-RGB+D (Shahroudy et al., 2016) datasets.

	MSR 3D (overall)	**KARD (overall)**	**NTU-RGB+D Cross-Subject**	**NTU-RGB+D Cross-View**
Prior works	96.50%	99.31%	76.20%	83.10%
Our results	99.90%	99.98%	78.20%	85.60%
Improvements	**3.40**%	**0.67**%	**2.00**%	**2.50**%

original ResNet architecture, we observed that our proposed networks are able to reduce the effects of the degradation phenomenon for both training and test phases. *E.g.* the proposed 56-layer networks achieved better results than 20-layer, 32-layer, and 44-layer networks on NTU-RGB+D dataset. Meanwhile, the original ResNet with 32-layer is the best network on this benchmark. The same learning behaviors are found in experiments on the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) and KARD (Gaglio, Re, and Morana, 2014) datasets (TABLE 4.10). It should be noted that degradation phenomena depend considerably on the size of datasets[5]. This is the reason why the 110-layer network got higher errors than several other networks.

The difference between training error and test error on the learning curves shows the ability of overfitting prevention. Our experimental results on three action benchmarks showed that the proposed ResNet architectures are capable of reducing overfitting in comparison with the original architecture. We believe this result comes from the combination between the use of BN (Ioffe and Szegedy, 2015) before convolutional layers and Dropout (Hinton et al., 2012) in each ResNet unit.

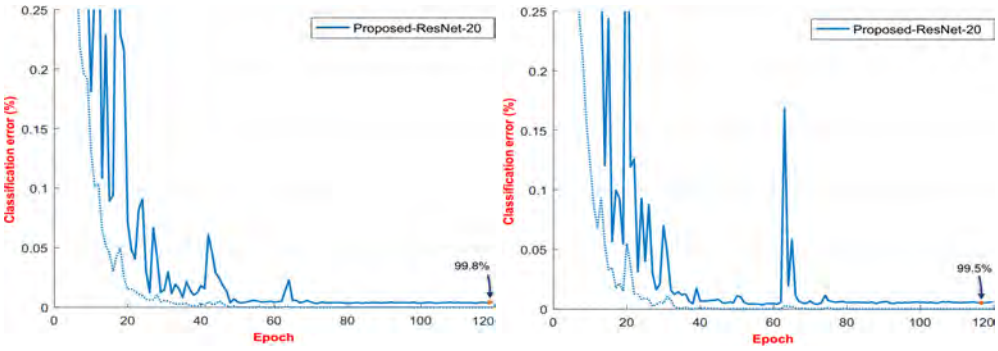
Effect of image resizing methods on recognition performance: D-CNNs work with fixed size tensors. Thus, before feeding image-based representations to ResNets, all these images were resized to a fixed size of 32×32 pixels. The resizing step may lead to the change

[5]Personal communication with H. Zang from the Rutgers University, USA and Amazon AI.

TABLE 4.10: Relationship between the number of layers and its performance on three benchmarks. The symbol ✓ denotes the best network based on the original ResNet architecture and ✗ denotes the best network based on the proposed ResNet architecture.

# Network layers	MSR Action3D	KARD	NTU-RGB+D
110			
56			✗
44	✗	✗	
32	✓		✓
20		✓	

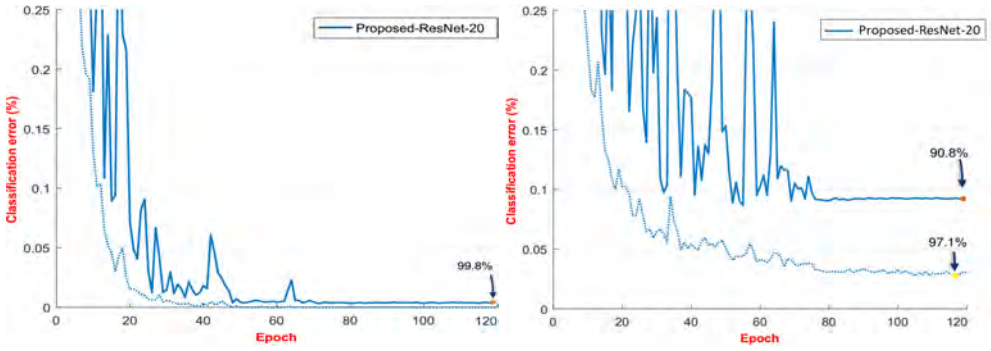
in the accuracy rate. To identify the effects of different resizing methods on the recognition performance of the proposed model, we conducted an additional experiment on the MSR Action3D/AS1 dataset (Wanqing, Zhengyou, and Zicheng, 2010) with Proposed-ResNet-20 network. In this experiment, two different resizing methods, including Nearest-Neighbor interpolation and Bicubic interpolation were used for resizing image-based representations before feeding to deep networks. Experimental results indicate that the difference between the accuracy rates is very small ($\Delta = 0.3\%$; see FIGURE 4.12). Whatever the ResNet depth, we show that the effect of resizing methods is the same. That is why we have only tested with the proposed ResNet-20 network. This choice obey to processing time.



(a) (b)

FIGURE 4.12: Training and test errors (%) by the Proposed-ResNet-20 network on the MSR Action3D/AS1 dataset (Wanqing, Zhengyou, and Zicheng, 2010): **(a)** resizing images using bicubic interpolation; **(b)** resizing images using nearest-neighbor interpolation.

Effect of joint order on recognition performance: In our study, each skeleton was divided into five parts and concatenated in a certain order in order to keep the local motion characteristics and to generate discriminative features in image-based representations. To clarify the effect of the order of joints in skeletons, we have tried to remove the step of rearranging joints in our implementation and perform experiments with the order of joints provided by the Kinect SDK. We observed a dramatically decrease in the recognition accuracy ($\Delta = 9.0\%$) as shown in FIGURE 4.13.



(a) (b)

FIGURE 4.13: Training and test errors (%) by the Proposed-ResNet-20 network on the MSR Action3D/AS1 dataset (Wanqing, Zhengyou, and Zicheng, 2010): **(a)** rearranging skeletons according to human body physical structure; **(b)** using the joints order provided by the Kinect SDK without rearranging skeletons.

Computational efficiency evaluation: We take the Cross-View evaluation criterion of the NTU-RGB+D dataset (Shahroudy et al., 2016) and the Proposed-ResNet-56 network to illustrate the computational efficiency of our learning framework. As shown in FIGURE 4.14, the proposed method has main components, including Stage 1 the encoding process from skeletons to RGB images, Stage 2 the supervised training stage, and Stage 3 the prediction stage. With the implementation in Matlab using MatConvNet toolbox (Vedaldi and Lenc, 2015) on a single NVIDIA GeForce GTX 1080 Ti GPU system[6], without parallel processing, we take 7.83×10^{-3} sec. per skeleton sequence during training. After about 80 epochs, our network starts converging with an accuracy around 85%. While the prediction time, including the time for encoding skeletons into RGB images and classification by pre-trained ResNet, takes 8.31×10^{-3} sec. per skeleton sequence. This speed is fast enough to meet many different applications.

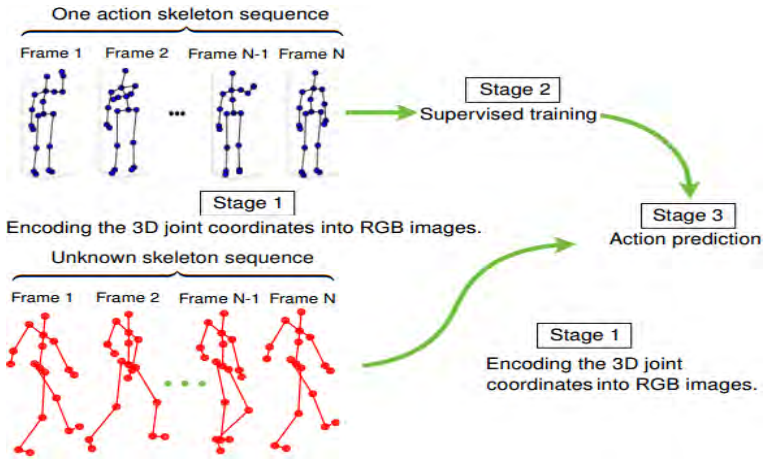


FIGURE 4.14: Three main phases of the proposed method for predicting human action from a new skeleton sequence.

[6]For more information about the specifications of this GPU, please refer to: https://www.nvidia.com/en-us/geforce/products/10series/geforce-gtx-1080-ti/.

TABLE 4.11: Execution time of each component of our method.

Component	Average processing time
Stage 1	7.83×10^{-3}s per sequence (Intel Core i7 3.2GHz CPU)
Stage 2	1.27×10^{-3}s per sequence (GTX 1080 Ti GPU)
Stage 3	8.31×10^{-3}s per sequence (GTX 1080 Ti GPU)

4.1.6 Conclusion

In this section, we have presented a novel deep learning framework based on ResNets for human action recognition in videos with skeletal data. The idea is to combine two important factors: a compact spatio-temporal representation of 3D motion combined with a powerful deep learning model. By encoding skeleton sequences into RGB images and proposing a novel ResNet architecture for learning human action from these images, higher levels of performance have been achieved. We showed that the approach was effective for recognizing actions on three well-established datasets. This work has been submitted to the Computer Vision and Image Understanding journal in September 2017 and accepted for publication in March 2018 (Pham et al., 2018b).

However, the proposed color encoding process is quite simple, which transforms the 3D body joint coordinates into RGB images via a normalization function. We believe that a better "*skeleton-to-image*" transformation that encodes richer motion features could help improve the learning performance. Furthermore, a more robust deep network architecture could lead to higher recognition accuracy. Therefore, the main aim of the work described in the next section is to extend the skeleton encoding method in which the Euclidean distance and the orientation relationship between joints are exploited. As a result we introduce a new 3D representation called SPMF (*Skeleton Pose-Motion Feature*). In addition, to achieve a better feature learning and classification framework, we aim to design and train some new and potential D-CNN architectures based on the idea of ResNet (Kaiming et al., 2016) such as Inception-ResNet-v2 (Szegedy et al., 2017).

4.2 SPMF: A new skeleton-based representation for 3D action recognition with Inception Residual Networks

In this section, we propose a novel skeleton-based representation for 3D action recognition, namely, SPMF (*Skeleton Pose-Motion Feature*) in videos using D-CNNs. The SPMFs are built from two of the most important properties of a human action: postures and their motions. Therefore, they are able to effectively represent complex actions. For learning and recognition tasks, we design and optimize new D-CNNs based on the idea of Inception Residual networks (Szegedy et al., 2017) to predict actions from SPMFs. Our method is evaluated on two challenging datasets including MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) and NTU-RGB+D (Shahroudy et al., 2016). Experimental results indicated that the proposed method surpasses state-of-the-art methods whilst requiring less computation.

4.2.1 Introduction

We believe that an effective representation of motion is the key factor influencing the performance of a skeleton-based action recognition model. To better represent the characteristics of 3D actions, we exploit body poses (Pose Features – PFs) and their motions (Motion Features – MFs) for building a new representation called SPMF (Skeleton Pose-Motion Feature). Each SPMF contains important characteristics related to the spatial structure and temporal dynamics of skeletons. Additionally, a well-designed and deeper CNN can improve learning accuracy. Therefore, a new deep framework based on the Inception Residual networks (Szegedy et al., 2017) is then proposed for learning and classifying (FIGURE 4.15). We exploit this architecture because it has been proved to be more robust than the ResNet architecture on some common benchmark datasets for object recognition tasks such as CIFAR

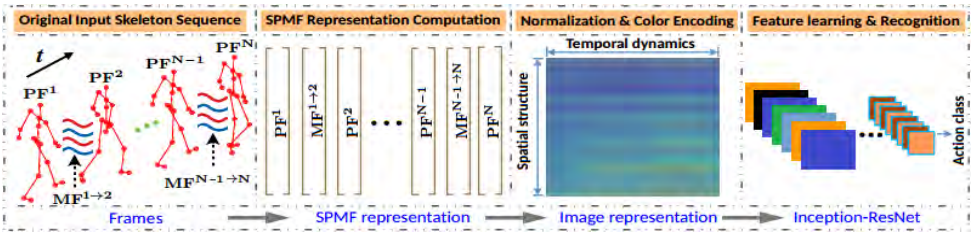


FIGURE 4.15: Schematic overview of our method. Each skeleton sequence is encoded into a color image via a skeleton-based representation called SPMF. Each SPMF is built from pose vectors (PFs) and motion vectors (MFs). They are then fed to a D-CNN, which is designed based on the combination of Residual learning (Kaiming et al., 2016) and Inception architecture (Szegedy et al., 2016) for learning discriminative features from color-coded SPMFs and performing action classification.

(Krizhevsky, 2009) and ImageNet (Russakovsky et al., 2015). Our experimental results on two benchmark datasets confirmed these statements.

The main contributions of this section are a novel skeleton-based representation, a new deep learning framework based on Inception-ResNet (Szegedy et al., 2017) and the proposed method set a new state-of-the-art on the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) and the NTU-RGB+D (Shahroudy et al., 2016) datasets.

4.2.2 Proposed method

SPMF: From skeleton movement to color map

Two key elements to determine an action are static postures and their motions. We propose SPMF, a novel representation based on these features that are extracted from skeletons. Note that, combining too many geometric features will lead to lower performance than using only a single feature or several main features (Zhang, Liu, and Xiao, 2017). In our study, each SPMF is built from pose and motion vectors, as described below:

Pose Feature (PF): Given a skeleton sequence $\mathcal{S}$ with N frames, denoted by $\mathcal{S} = \{F^t\}$, where $t = 1, 2, 3, ..., N$. Let $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ be the 3D coordinates of the j-th and k-th joints in F^t. The Joint-Joint Distance JJD_{jk}^t between $\mathbf{p}_j^t$ and $\mathbf{p}_k^t$ at timestamp t is computed as

$$JJD_{jk}^t = ||\mathbf{p}_j^t - \mathbf{p}_k^t||_2, \quad (t = 1, 2, 3, ..., N), \tag{4.11}$$

where $||\cdot||_2$ denotes the Euclidean distance between two joints. The joint distances obtained by Eq. (4.11) for all types of actions of a specific dataset range from $D_{\min} = 0$ to $D_{\max} = \max\{JJD_{jk}^t\}$. We note this distance space as $\mathcal{D}_{original}$. In fact, $\mathcal{D}_{original}$ can be transformed into a tensor-structure and fed directly to D-CNNs for learning action features. However, since $\mathcal{D}_{\text{original}}$ is a high-dimensional space, it could lead D-CNNs to overfit as well as being time-consuming. Thus, we need to describe the input skeleton sequences as low-dimensional signals such that they are easy to parameterize by learning models and discriminative enough for a classification task. To do that, we normalize all elements of $\mathcal{D}_{original}$ to the range $[0, 1]$, denoted as $\mathcal{D}_{[0,1]}$. To reflect the change in joint distances, we encode $\mathcal{D}_{[0,1]}$ into a color space using a sequential discrete color palette called JET color map[7]. The encoding process converts the joint distances $JJD_{jk}^t \in \mathcal{D}_{[0,1]}$ for all possible combinations j and k into color points $\mathbf{JJD}_{RGB}^t \in \mathbb{N}_{[0,255]}^3$ performed by 256-color JET scale. To this end, we first normalize the distance values with respect to the maximum and minimum values of a grayscale image ranging from 0 to 1. As illustrated in FIGURE 4.16, the scalar distances are converted to a three channel map via a JET mapping. This technique is similar to depth

[7] A JET color map is based on the order of colors in the spectrum of visible light, ranging from blue to red, and passing through the cyan, yellow, and orange.

encoding method presented in Eitel et al., 2015. The use of a discrete color palette allows us to reduce complexity of input features. This helps accelerate the convergence rate of deep learning networks during the training stage. Moreover, it should be noted that point-point distances are invariant when they are moved into a new coordinate system in the 3D Euclidean space. Therefore, the use of the Joint-Joint Distance JJD^t_{jk} can help our final representation be more independent to the camera viewpoint.

Apart from the distance information, the orientation between joints is also important for

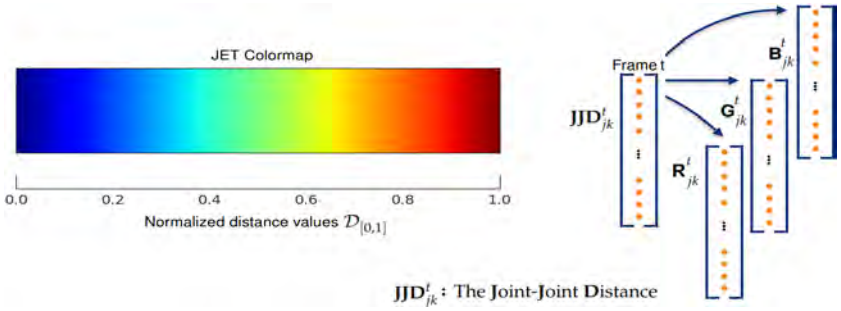


FIGURE 4.16: Illustration of the encoding process that converts joint-joint distance values to color points using a JET colormap.

describing human motions. The Joint-Joint Orientation $\mathbf{JJO}^t_{jk}$ from joint $\mathbf{p}^t_j$ to $\mathbf{p}^t_k$ at timestamp t is computed as

$$\mathbf{JJO}^t_{jk} = \mathbf{p}^t_j - \mathbf{p}^t_k, \quad (t = 1, 2, 3, ..., N). \tag{4.12}$$

The $\mathbf{JJO}^t_{jk}$ is a vector where all of its components p can be normalized to the range $[0, 255]$. This can be done via the following transformation

$$p_{\text{norm}} = \texttt{floor}(255 \times \frac{p - c_{\min}}{c_{\max} - c_{\min}}), \tag{4.13}$$

where p_{norm} indicates the normalized value, $c_{\max}$ and $c_{\min}$ are the maximum and minimum values of all coordinates over the training set, respectively. The function $\texttt{floor}(\cdot)$ rounds down to the nearest integer. We consider three components (x, y, z) of $\mathbf{JJO}^t_{jk}$ after normalization as the corresponding three components (R, G, B) of a color pixel and build $\mathbf{JJO}^t_{RGB}$ as a 3D array that is formed by all $\mathbf{JJO}^t_{jk}$ values. We then define "*a human pose*" at timestamp t by vector $\mathbf{PF}^t$ that describes the distance and orientation relationship between skeleton joints,

$$\mathbf{PF}^t = \left[\mathbf{JJD}^t_{RGB} +\!\!+ \mathbf{JJO}^t_{RGB}\right], \quad (t = 1, 2, 3, ..., N). \tag{4.14}$$

Here the symbol $(+\!\!+)$ horizontally concatenates vectors $\mathbf{JJD}^t_{RGB}$ and $\mathbf{JJO}^t_{RGB}$ together.

Motion Feature (MF): Let $\mathbf{p}^t_j$ and $\mathbf{p}^{t+1}_k$ denote the 3D coordinates of the j-th and k-th joints at two consecutive frames F^t and F^{t+1}. Similarly to JJD^t_{jk} in Eq. (4.11), the Joint-Joint Distance $JJD^{t,t+1}_{jk}$ between $\mathbf{p}^t_j$ and $\mathbf{p}^{t+1}_k$ is computed as

$$JJD^{t,t+1}_{jk} = ||\mathbf{p}^t_j - \mathbf{p}^{t+1}_k||_2, \quad (t = 1, 2, 3, ..., N-1). \tag{4.15}$$

Also, similarly to Eq. (4.12), the Joint-Joint Orientation $\mathbf{JJO}^{t,t+1}_{jk}$ from joint $\mathbf{p}^t_j$ to $\mathbf{p}^{t+1}_k$ is computed as

$$\mathbf{JJO}^{t,t+1}_{jk} = \mathbf{p}^t_j - \mathbf{p}^{t+1}_k, \quad (t = 1, 2, ..., N-1). \tag{4.16}$$

We define "*a human motion*" from t to $t+1$ by vector $\mathbf{MF}^{t\to t+1}$, in which

$$\mathbf{MF}^{t\to t+1} = \left[\mathbf{JJD}^{t,t+1}_{RGB} +\!\!+ \mathbf{JJO}^{t,t+1}_{RGB}\right], \quad (t = 1, 2, ..., N-1), \tag{4.17}$$

where $\mathbf{JJD}_{RGB}^{t,t+1}$ and $\mathbf{JJO}_{RGB}^{t,t+1}$ are encoded to qualify the color representation as $\mathbf{JJD}_{RGB}^{t}$ and $\mathbf{JJO}_{RGB}^{t}$, respectively.

Modeling human action with PFs and MFs: Based on the obtained PFs and MFs, we propose a skeleton-based representation called SPMF for 3D human action recognition. To this end, all PFs and MFs computed from the skeleton sequence $\mathcal{S}$ are concatenated into a single feature vector in temporal order from the beginning to the end of the action. It is a global representation for the whole skeleton sequence $\mathcal{S}$ without dependence on the range of action and can be obtained by

$$\begin{aligned}\mathbf{SPMF} = [\mathbf{PF}^1 +\!\!+ \mathbf{MF}^{1\to2} +\!\!+ \mathbf{PF}^2 +\!\!+ ... +\!\!+ \mathbf{PF}^t +\!\!+ \mathbf{MF}^{t\to t+1} +\!\!+ \mathbf{PF}^{t+1} ... \\ +\!\!+ \mathbf{PF}^{N-1} +\!\!+ \mathbf{MF}^{N-1\to N} +\!\!+ \mathbf{PF}^N].\end{aligned} \tag{4.18}$$

FIGURE 4.17 shows some SPMFs obtained from the MSR Action3D dataset after resizing them to 32×32 pixels.

FIGURE 4.17: The SPMFs obtained from some samples of the MSR Action3D dataset. Color-changing reflects the change in distance between skeleton joints. Best viewed in color.

Inception Residual Network (Inception-ResNet)

D-CNNs have demonstrated state-of-the-art performance on many visual recognition tasks. In particular, the recent Inception architecture (Szegedy et al., 2016) significantly improved both the accuracy and computational cost through three key ideas: (**1**) reducing the input dimension; (**2**) increasing not only the network depth, but also its width and (**3**) concatenating feature maps learned by different layers. However, very deep networks as Inception are very difficult to train due to the vanishing problem and degradation phenomenon (He and Sun, 2015). To this end, ResNet (Kaiming et al., 2016) has been introduced. The key idea is to improve the flow of information and gradients through layers by using identity connections. A layer, or a sequence of layers of a traditional CNN learns to calculate a mapping function $y = \mathcal{F}(x)$ from the input feature x. Meanwhile, a ResNet building block approximately calculates the function $y = \mathcal{F}(x) + id(x)$ where $id(x) = x$. This idea helps the learning process to be faster and more accurate. To learn spatio-temporal features from the SPMFs, we propose the combination of Residual learning (Kaiming et al., 2016) and Inception architecture (Szegedy et al., 2016) to build D-CNNs (see APPENDIX B2 to see details of the proposed network architectures). Batch normalization (Ioffe and Szegedy, 2015) and Exponential Linear Units (ELUs – Clevert, Unterthiner, and Hochreiter, 2015) are applied after each Convolution. Dropout (Hinton et al., 2012) with a rate of 0.5 is used to prevent overfitting. A Softmax layer is employed for classification task. Our networks can be trained in an end-to-end manner by the gradient descent using Adam update rule (Kingma and Ba, 2014). During training, our goal is to minimize the cross-entropy loss function between the ground-truth label $\mathbf{y}$ and the predicted label $\hat{\mathbf{y}}$ by the network over the training samples $\mathcal{X}$, which is expressed as follows:

$$\mathcal{L}_{\mathcal{X}}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{M}\left(\sum_{i=1}^{M}\left(\sum_{j=1}^{C}\mathbf{y}_{ij}\log\hat{\mathbf{y}}_{ij}\right)\right) \tag{4.19}$$

where M indicates the number of samples in training set $\mathcal{X}$ and C denotes the number of action classes.

4.2.3 Experiments

Datasets and settings

The proposed method is evaluated on the MSR Action3D and NTU-RGB+D datasets. We follow the evaluation protocols as provided in the original papers as desribed in section 4.1. The performance is measured by average classification accuracy over all action classes. We did not use the KARD dataset because it is quite small and simple, and it doesn't reflect well the learning performance of deep neural networks. Using the method proposed in the previous section, we already obtained an average accuracy of 99.8% on this dataset.

Implementation details

Three different network configurations were implemented and evaluated in Python with Keras framework[8] using the TensorFlow[9] backend. During training, we use mini-batches of 256 images for all networks. The weights are initialized by the He initialization technique (He et al., 2015). Adam optimizer (Kingma and Ba, 2014) is used with default parameters, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The initial learning rate is set to 0.001 and is decreased by a factor of 0.5 after every 20 epochs. All networks are trained for 250 epochs from scratch. We applied some simple data augmentation techniques (*i.e.* randomly cropping, flipping and Gaussian filtering) on the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010) due to its small size. For the NTU-RGB+D (Shahroudy et al., 2016), we do not apply any data augmentation method.

4.2.4 Experimental results and analysis

TABLE 4.12 reports the experimental results and comparisons with state-of-the-art methods on the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010). We achieved the best recognition accuracy by SPMF Inception-ResNet-222 network configuration with a total average accuracy of 98.56%. This result outperforms many previous studies (Chen, Liu, and Kehtarnavaz, 2013; Vemulapalli, Arrate, and Chellappa, 2014; Du, Wang, and Wang, 2015; Liu et al., 2016b; Wang et al., 2016b; Weng, Weng, and Yuan, 2017; Xu et al., 2015a; Li et al., 2017b). For the NTU-RGB+D dataset (Shahroudy et al., 2016), we achieved an accuracy of 78.89% on cross-subject evaluation and 86.15% on cross-view evaluation as shown in TABLE 4.13. These results are better than previous state-of-the-art works reported in Vemulapalli, Arrate, and Chellappa, 2014; Du, Wang, and Wang, 2015; Liu et al., 2016b; Shahroudy et al., 2016; Hu et al., 2015b; Rahmani and Bennamoun, 2017. Finally, the comparison between SPMF associated with Inception-ResNet-v2 and skeleton-based ResNet (section 4.1) shows that SPMF-based model is slightly better: 78.89% versus 78.20% for the cross-subject evaluation and 86.15% versus 85.60% for the cross-view evaluation. However, there is no improvement on the MSR Action3D dataset. We believe that deeper and larger networks such as Inception-ResNet-v2 are more adapted to larger datasets.

[8] https://keras.io/
[9] https://www.tensorflow.org/

TABLE 4.12: Accuracy rate (%) on the MSR Action3D dataset. The symbol ‡ denotes the number of building blocks Inception-ResNet-A, Inception-ResNet-B, and Inception-ResNet-C, respectively. Details are provided in APPENDIX B2.

Method (protocol of Wanqing, Zhengyou, and Zicheng, 2010)	AS1	AS2	AS3	Aver.
Bag of 3D Points (Wanqing, Zhengyou, and Zicheng, 2010)	72.90	71.90	71.90	74.70
Depth Motion Maps (Chen, Liu, and Kehtarnavaz, 2013)	96.20	83.20	92.00	90.47
Lie Group (Vemulapalli, Arrate, and Chellappa, 2014)	95.29	83.87	98.22	92.46
Hierarchical RNN (Du, Wang, and Wang, 2015)	99.33	94.64	95.50	94.49
ST-LSTM Trust Gates (Liu et al., 2016b)	N/A	N/A	N/A	94.80
Graph-Based Motion (Wang et al., 2016b)	93.60	95.50	95.10	94.80
ST-NBNN (Weng, Weng, and Yuan, 2017)	91.50	95.60	97.30	94.80
S-T Pyramid (Xu et al., 2015a)	99.10	92.90	96.40	96.10
Ensemble TS-LSTM v2 (Li et al., 2017b)	95.24	96.43	**100.0**	97.22
Skeleton-based ResNet (section 4.1)	**99.90**	99.80	**100.0**	**99.90**
SPMF Inception-ResNet-121 ‡	97.06	99.00	98.09	98.05
SPMF Inception-ResNet-222	**97.54**	**98.73**	**99.41**	**98.56**
SPMF Inception-ResNet-242	96.73	97.35	98.77	97.62

TABLE 4.13: Accuracy rate (%) on NTU-RGB+D dataset.

Method (protocol of Shahroudy et al., 2016)	Cross-Subject	Cross-View
Lie Group (Vemulapalli, Arrate, and Chellappa, 2014)	50.10	52.80
Hierarchical RNN (Du, Wang, and Wang, 2015)	59.07	63.97
ST-LSTM Trust Gates (Liu et al., 2016b)	69.20	77.70
Two-Layer P-LSTM (Shahroudy et al., 2016)	62.93	70.27
Dynamic Skeletons (Hu et al., 2015b)	60.20	65.20
STA-LSTM (Song et al., 2017)	73.40	81.20
Depth and Skeleton Fusion (Rahmani and Bennamoun, 2017)	75.20	83.10
Skeleton-based ResNet (section 4.1)	78.20	85.60
SPMF Inception-ResNet-121	77.02	82.13
SPMF Inception-ResNet-222	**78.89**	**86.15**
SPMF Inception-ResNet-242	77.24	83.45

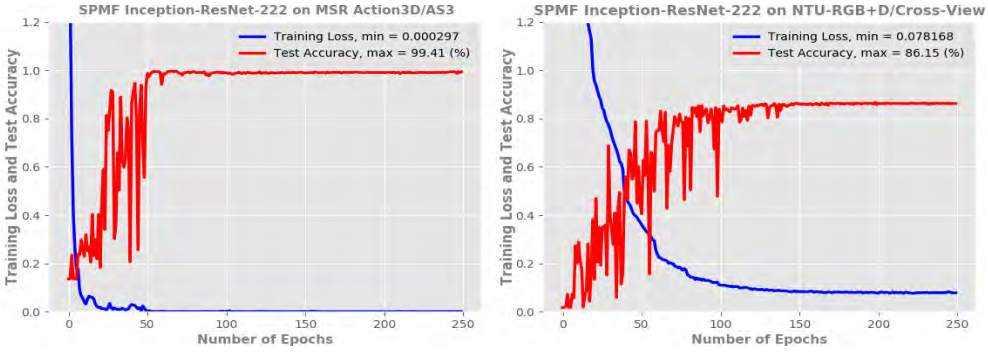


FIGURE 4.18: Training loss and test accuracy of SPMF-Inception-ResNet-222 on MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) and NTU-RGB+D (Shahroudy et al., 2016) datasets.

FIGURE 4.19: Visualizing intermediate features generated by Inception-ResNet-222 after feeding several SPMFs into the network. These SPMFs come from samples in the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010). Best viewed in color.

4.2.5 Processing time: training and prediction

We take the NTU-RGB+D dataset (Shahroudy et al., 2016) with cross-view settings and SPMF-Inception-ResNet-222 network for illustrating the computational efficiency of our learning framework. With the implementation in Python on a single GeForce GTX Ti GPU, no parallel processing, the training phase takes 1.85×10^{-3} sec. per sequence in which skeletons are already encoded into RGB images. While the testing phase, including the time for encoding skeletons into color images and classification, takes 0.128 sec. per sequence. These results verify the effectiveness of the proposed method, not only in terms of accuracy, but also in terms of computational cost.

4.2.6 Conclusion

Section 4.2 introduced a new method for recognizing human actions from skeletal data. A novel skeleton-based representation, namely SPMF, is proposed for encoding spatio-temporal dynamics of skeleton joints into color images. Deep Convolutional Neural Networks (D-CNNs) based on Inception Residual architecture are then exploited to learn and recognize actions from obtained image-based representations. Experiments on two publicly available benchmark datasets have demonstrated the effectiveness of the proposed representation as well as feature learning networks. This work has been published in the International Conference on Image Processing (ICIP) in October 2018 (Hieu Pham et al., 2018).

The comparison with the skeleton-based ResNet presented in section 4.1 showed a slight improvement accuracy on the NTU-RGB+D dataset (Shahroudy et al., 2016). We think that rich features such as pose and motion features could make the proposed SPMF more robust to the change of camera viewpoints. This represents the work carried out and presented in the next section. Indeed, we aim to improve the SPMF representation by making it more robust to noise and more discriminative for classification task. This can be done by using a smoothing filter and an image enhancement method. This new method, called Enhanced-SPMF is presented in section 4.3. A new state-of-the-art deep neural network, namely DenseNet is used for representation, learning and classification tasks.

4.3 Enhanced-SPMF: An extended representation of the SPMF for 3D human action recognition with Deep Convolutional Neural Networks

4.3.1 Introduction

In this section, we consider a new motion representation called Enhanced-SPMF. The proposed Enhanced-SPMF has a 2D image structure with three color channels, which is built from a set of spatio-temporal stages, combining 3D skeleton poses and their motions. Moreover, an Adaptive Histogram Equalization (AHE) algorithm (Pizer et al., 1987) is then applied to the color images to enhance their local patterns and generate more discriminative features for classification task. FIGURE 4.20 illustrates an overview of the proposed Enhanced-SPMF. To learn image features and recognize action labels from the proposed

representation, different D-CNN models based on the DenseNet architecture (Huang et al., 2017) have been designed and evaluated. There are two important hypotheses that moti-

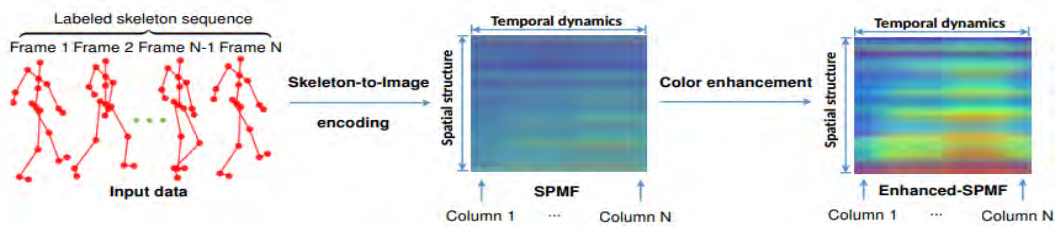


FIGURE 4.20: Overview of the proposed Enhanced-SPMF representation. Each skeleton sequence is transformed into a single RGB that is a motion map called SPMF. A color enhancement technique (Pizer et al., 1987) is then used to highlight the motion map and form the Enhanced-SPMF, which will be learned and classified by a deep learning model. Before computing the SPMF, a smoothing filter is applied to reduce the effect of noise on skeletal data.

vate us to propose the Enhanced-SPMF and design DenseNets (Huang et al., 2017) for 3D human action recognition with skeletal data. First, the SPMF can be more discriminative if its local features are enhanced by a color enhancement method. More accurate skeleton joints can be also obtained if a filter is applied before starting the process of color encoding. Second, DenseNet (Huang et al., 2017) can improve accuracy in the image recognition task since this kind of network is able to prevent overfitting and degradation phenomena (He and Sun, 2015) by maximizing information flow and facilitating features reuse as each layer in its architecture has direct access to the features from previous layers. Experimental results on the ImageNet dataset (Russakovsky et al., 2015) for common classification tasks showed that DenseNet is able to achieve better performance than ResNet (Kaiming et al., 2016) and Inception-ResNet-v2 (Szegedy et al., 2017). Therefore, we explore the use of DenseNet in this work and optimise this architecture for learning and recognizing human actions on the proposed image-based representation.

The effectiveness of the proposed method is evaluated on three public benchmark RGB-D datasets, including MSR Action3D (Li, Zhang, and Liu, 2010), SBU Kinect Interaction (Yun et al., 2012a) and NTU-RGB+D datasets (Shahroudy et al., 2016). Except for the SBU Kinect Interaction dataset (Yun et al., 2012a) in section 4.1 we have already provided the description of the rest datasets. The specificity of the SBU Kinect Interaction is that it contains many interactions between people which is not the case for the two other datasets.

The main contributions of our study include two aspects:

• **Firstly**, we present Enhanced-SPMF, a new skeleton-based representation for 3D human action recognition from skeletal data. The Enhanced-SPMF is an extended representation of SPMF which was presented in section 4.2. Compared to our previous work, the current work aims to improve the efficiency of the 3D motion representation via a smoothing filter and a color enhancement technique. The smoothing filter helps us to reduce the effect of noise on skeletal data, meanwhile the color enhancement technique could make the proposed Enhanced-SPMF more robust and discriminative for recognition task. An ablation study on the Enhanced-SPMF was also carried out leading to a better overall action recognition performance than the SPMF.

• **Secondly**, we present a deep learning framework based on the DenseNet architecture (Huang et al., 2017) for learning discriminative features from the proposed Enhanced-SPMF and performing action classification. The framework directly learns an end-to-end mapping between skeleton sequences and their action labels with little pre-processing. Compared to our previous work that exploited the Residual Inception v2 network, the current work uses a more powerful deep learning model for action recognition task.

In the following, we present the details of the proposed approach in section 4.3.2. Experimental settings are in section 4.3.3 and the experimental results in section 4.3.4.

4.3.2 Proposed method

FIGURE 4.21 illustrates the key components of the proposed learning framework for recognizing actions from skeleton sequences. We first show how skeleton pose and motion features can be combined to build an action map in the form of an image-based representation and how to use a color enhancement technique for improving the discriminative ability of the proposed representation. We then introduce an end-to-end deep leaning framework based on DenseNets to learn and classify actions from the enhanced representations. Before that, in order to put the proposed approach into context, it is useful to review the central ideas behind the original DenseNet architecture.

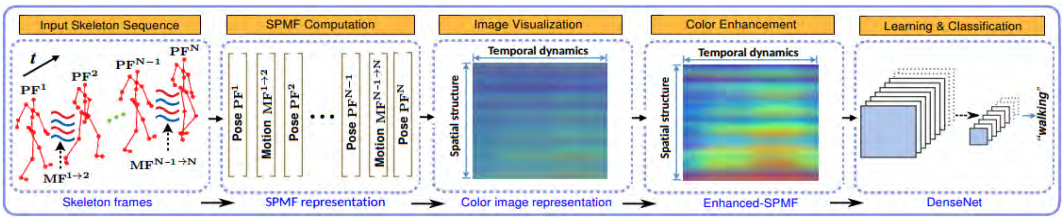


FIGURE 4.21: Schematic overview of the proposed approach. Each skeleton sequence is encoded in a single color image via the SPMF. They are then enhanced by an Adaptive Histogram Equalization (AHE – Pizer et al., 1987) algorithm and fed to a DenseNet (Huang et al., 2017) for learning discriminative features and performing action classification.

Enhanced-SPMF: Building an enhanced 3D action map

The SPMF representations mainly reflect the spatio-temporal distribution of skeleton joints. We visualize these representations and observe that they tend to be low contrast images, as shown in FIGURE 4.22. In this case, a color enhancement method can be useful for increasing contrast and highlighting the texture and edges of the motion maps. Therefore, it is necessary to enhance the local features on the generated color images after encoding. The Adaptive Histogram Equalization (AHE – Pizer et al., 1987) is a common approach for this task. This technique is capable of enhancing the local features of an image. Mathematically, let $\boldsymbol{I}$ be a given digital image, represented as a r-by-c matrix of integer pixels with intensity levels in the range $[0, \mathcal{L}-1]$. The histogram of image $\boldsymbol{I}$ will be defined by

$$H_k = n_k, \tag{4.20}$$

where n_k is the number of pixels in $\boldsymbol{I}$ with intensity k. The probability of occurrence of intensity level k in $\boldsymbol{I}$ can be estimated by

$$p_k = \frac{n_k}{r \times c}, \quad (k = 0, 1, 2, ..., \mathcal{L}-1). \tag{4.21}$$

The histogram equalized image is defined by transforming the pixel intensities, n, of $\boldsymbol{I}$ by the function

$$T(n) = \texttt{floor}((\mathcal{L}-1)\sum_{k=0}^{n} p_k), \quad (n = 0, 1, 2, ..., \mathcal{L}-1), \tag{4.22}$$

The Histogram Equalization (HE) method is used for increasing the global contrast of the image. However, it cannot solve the problem of increasing local contrast. To overcome this limitation, the image needs to be divided into $\mathcal{R}$ regions and the HE is then applied in each and every one of these regions. This technique is called the Adaptive Histogram Equalization algorithm (AHE – Pizer et al., 1987). The bottom row of FIGURE 4.22 shows samples of the enhanced motion map with $\mathcal{R}$ = 8 on 32 $\times$ 32 images, which we refer to

it as Enhanced-SPMF, for some actions from the MSR Action3D dataset (Li, Zhang, and Liu, 2010). Before transforming skeleton joints into the Enhanced-SPMF representations,

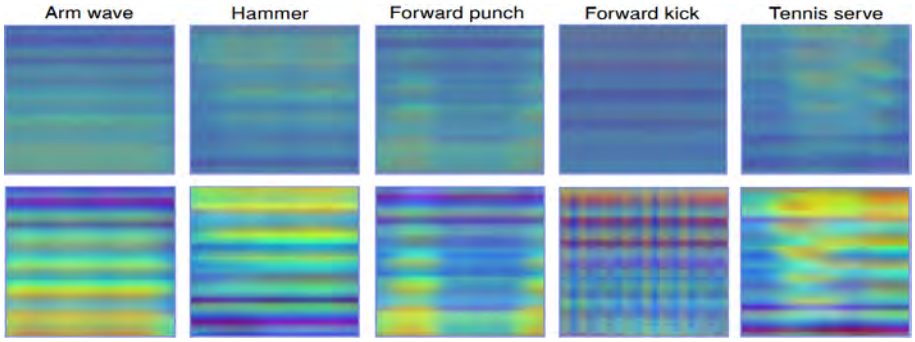


FIGURE 4.22: Results of the skeleton-to-image mapping process. The top row shows the proposed SPMF representations obtained from some samples of the MSR Action3D dataset (Li, Zhang, and Liu, 2010). The change in color reflects the change of distance and orientation between the joints. The bottom row shows generated images after applying the AHE algorithm (Pizer et al., 1987).

we apply the Savitzky-Golay (S-G) filter (see APPENDIX C) on row skeletal data. This is a *low-pass* filter based on local least-squares polynomial approximation that is often used to smooth noisy data. In our case, this filter could reduce the effect of noise on input skeleton sequences.

Densely Connected Convolutional Network (DenseNet)

DenseNet (Huang et al., 2017), considered as the current state-of-the-art CNN architecture, has some interesting properties. In this architecture (Huang et al., 2017), each layer is connected to all the others within a dense block and all layers can access to the feature maps from their preceding layers. Besides, each layer receives direct information flow from the loss function through the shortcut connections. These properties help DenseNet (Huang et al., 2017) to be less prone to overfitting for supervised learning problems. Mathematically, traditional CNN architectures, *e.g.* AlexNet (Krizhevsky, Sutskever, and Hinton, 2012b) or VGGNet (Simonyan and Zisserman, 2014b), connect the output feature maps $\mathbf{x}_{l-1}$ of the $(l-1)^{\text{th}}$ layer as input to the l^{th} layer and try to learn a mapping function

$$\mathbf{x}_l = \mathcal{H}_l(\mathbf{x}_{l-1}), \tag{4.23}$$

where $\mathcal{H}_l(\cdot)$ is a non-linear transformation and usually implemented via a series of operations such as Convolution (Conv.), Rectified Linear Unit (ReLU – Glorot, Bordes, and Bengio, 2011), Pooling (Lecun et al., 1998), and Batch Normalization (BN – Ioffe and Szegedy, 2015). When increasing the depth of the network, the network training process becomes complex due to the vanishing-gradient problem and the degradation phenomenon (He and Sun, 2015). To solve these problems, Kaiming et al., 2016 introduced ResNet. Inspired by the philosophy of ResNet (Kaiming et al., 2016), to maximize information flow through layers, Huang et al., 2017 proposed DenseNet with a simple connectivity pattern: the l^{th} layer in a dense block receives the feature maps of all preceding layers as inputs. That means

$$\mathbf{x}_l = \mathcal{H}_l([\mathbf{x}_0 +\!\!+ \mathbf{x}_1 +\!\!+ \mathbf{x}_2 +\!\!+ ... +\!\!+ \mathbf{x}_{l-1}]), \tag{4.24}$$

where $[\mathbf{x}_0 +\!\!+ \mathbf{x}_1 +\!\!+ \mathbf{x}_2 +\!\!+ ... +\!\!+ \mathbf{x}_{l-1}]$ is a single tensor constructed by concatenation of the output feature maps of the previous layers. Additionally, all layers in the architecture receive direct supervision signals from the loss function through the shortcut connections. In this manner, the network is easy to optimize and resistant to overfitting. In DenseNet (Huang et al., 2017), multiple dense blocks are connected via transition layers. Each transition layer consists of a convolutional layer and followed by an average pooling layer that changes the

size of feature maps[10]. Each block with its transition layer produces k feature maps and the parameter k is called as the *"growth rate"* of the network. The non-linear function $\mathcal{H}_l(\cdot)$ in the original work (Huang et al., 2017) is a composite function of three consecutive operations: BN-ReLU-Conv.

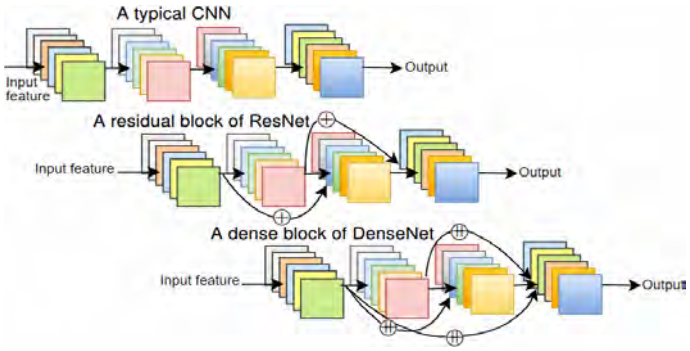


FIGURE 4.23: Illustration of the structure of a typical CNN (Krizhevsky, Sutskever, and Hinton, 2012b – *top row*), a ResNet building block (Kaiming et al., 2016 – *middle row*) and a DenseNet building block (Huang et al., 2017 – *bottom row*). The symbols $\oplus$ and $\bigoplus$ denote the summation and concatenation operators, respectively.

Network design

We propose to design and optimize deep DenseNets (Huang et al., 2017) for learning and classifying human actions on the Enhanced-SPMFs. To study how recognition performance varies with architecture size, we explore different network configurations. The following configurations are used in our experiments: DenseNet ($L = 100$, $k = 12$) ; DenseNet ($L = 250$, $k = 24$); and DenseNet ($L = 190$, $k = 40$), where L is the depth of the network and k is the network growth rate. On all datasets, we use three dense blocks on 32×32 input images. In this design, $\mathcal{H}_l(\cdot)$ is defined as Batch Normalization (BN – Ioffe and Szegedy, 2015), followed by an advanced activation layer called Exponential Linear Unit (ELU – Clevert, Unterthiner, and Hochreiter, 2015) and 3×3 Convolution (Conv.). A Dropout (Hinton et al., 2012) with a rate of 0.2 is used after each Convolution to prevent overfitting. After the feature extraction stage, a Full Connected (FC) layer is used for classification task in which the number of neurons for this FC layer is equal to the number of action classes in each dataset. The proposed networks can be trained in an end-to-end manner by gradient descent using Adam update rule (Kingma and Ba, 2014). During the training stage, we minimize a cross-entropy loss function, which is measured by the difference between the true action label $\mathbf{y}$ and the predicted action $\hat{\mathbf{y}}$ by the networks over the training samples $\mathcal{X}$. In other words, the network will be trained to solve the following optimization problem

$$\text{Arg min}_{\mathcal{W}} \left(\mathcal{L}_{\mathcal{X}}(\mathbf{y}, \hat{\mathbf{y}})\right) = \text{Arg min}_{\mathcal{W}} \left(-\frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{C} \mathbf{y}_{ij} \log \hat{\mathbf{y}}_{ij} \right), \tag{4.25}$$

where $\mathcal{W}$ is the set of weights that will be optimized by the model, M denotes the number of samples in training set $\mathcal{X}$ and C is the number of action classes.

[10]The concatenation operation used in Eq. (4.24) is not viable when the size of feature maps changes.

4.3.3 Experiments

Datasets and settings

For the MSR Action3D (Li, Zhang, and Liu, 2010) and NTU-RGB+D (Shahroudy et al., 2016) datasets, we use the same evaluation settings as in section 4.1 and section 4.2. The SBU Kinect Interaction dataset (Yun et al., 2012a) was collected using the Kinect v1 sensor. It contains 282 skeleton sequences and 6822 frames performed by 7 participants. Each frame of the SBU Kinect dataset (Yun et al., 2012a) contains skeleton joints of two subjects corresponding to an interaction, each skeleton has 15 key joints. There are 8 interactions in total, including *approaching, departing, pushing, kicking, punching, exchanging objects, hugging,* and *shaking hands*. This dataset is challenging due to the fact that the joint coordinates exhibit low accuracy. Moreover, they contain non-periodic actions as well as very similar body movements. For instance, there are some pairs of actions that are difficult to distinguish such as *exchanging objects – shaking hands* or *pushing – punching*. We randomly split the whole dataset into 5 folds, in which 4 folds are used for training and the remaining 1 fold is used for test. It should be noted that each skeleton frame provided by the SBU dataset (Yun et al., 2012a) contains two separate subjects. Therefore, we consider them as two data samples and feature computation is conducted separately for the two skeletons. Additionally, data augmentation (*i.e.* random cropping, vertically flipping, rotation with $\alpha = 90°$) has been also applied on the SBU dataset (Yun et al., 2012a).

Implementation details

For the four considered datasets, the proposed Enhanced-SPMF representations are computed directly from the raw skeleton sequences without using a fixed number of frames. For computational efficiency, all the image representations are resized to 32×32 pixels. The three network configurations: DenseNet ($L = 100$, $k = 12$); DenseNet ($L = 250$, $k = 24$); and DenseNet ($L = 190$, $k = 40$) were implemented and evaluated in Python with the support of the Keras framework using TensorFlow as backend. During the training stage, we use mini-batches of 32 images for all networks. The weights are initialized as per the He initialization technique (He et al., 2015). Adam optimizer (Kingma and Ba, 2014) is used with defaut parameters (*i.e.*, $\beta_1 = 0.9$ and $\beta_2 = 0.999$). Additionally, we use a dynamic learning rate during training. The initial learning rate is set to 0.01 and is decreased by a factor of 0.1 after every 50 epochs. All networks are trained for 300 epochs from scratch.

4.3.4 Experimental results and analysis

Results on MSR Action3D dataset

Experimental results and comparisons of the proposed method with the current state-of-the-art approaches on the MSR Action3D dataset (Li, Zhang, and Liu, 2010) are summarized in TABLE 4.14. We compare the proposed method with Bag of 3D Points (Li, Zhang, and Liu, 2010), Depth Motion Maps (Chen, Liu, and Kehtarnavaz, 2013), Bi-LSTM (Tanfous, Drira, and Amor, 2018), Lie Group (Vemulapalli, Arrate, and Chellappa, 2014), FTP-SVM (Tanfous, Drira, and Amor, 2018), Hierarchical LSTM (Du, Wang, and Wang, 2015), ST-LSTM Trust Gates (Liu et al., 2016b), Graph-Based Motion (Wang et al., 2016b), ST-NBNN (Weng, Weng, and Yuan, 2017), ST-NBMIM (Weng et al., 2018), S-T Pyramid (Xu et al., 2015a), Ensemble TS-LSTM v2 (Li et al., 2017b) and our previous models skeleton-based ResNet (4.1) and SPMF Inception-ResNet-222 (see 4.2) using the same evaluation protocol. The proposed DenseNets ($L = 100$, $k = 12$) and DenseNet ($L = 190$, $k = 40$) achieve average accuracies of 98.76% and 98.94%, respectively. Meanwhile, the best recognition accuracies are obtained by the proposed DenseNet ($L = 250$, $k = 24$) with a total average accuracy of 99.10%. This result outperforms many previous approaches (Li, Zhang, and Liu, 2010; Chen, Liu, and Kehtarnavaz, 2013; Tanfous, Drira, and Amor, 2018; Vemulapalli, Arrate, and Chellappa, 2014; Du, Wang, and Wang, 2015; Liu et al., 2016b; Wang et al., 2016b; Weng, Weng, and Yuan, 2017; Weng et al., 2018; Xu et al., 2015a), demonstrating the superiority of the proposed method.

FIGURE 4.25 (*first row*) shows learning curves of the proposed DenseNets on the AS1 subset/MSR Action3D dataset (Li, Zhang, and Liu, 2010). The recognition accuracy for each action class in the AS1 subset by the DenseNet (L = 250, k = 24) is provided in FIGURE 4.24 via its confusion matrix. For this dataset, the comparison between Enhanced-SPMF and SPMF gives the following results: in the average column (TABLE 4.14) one can notice that the result for Enhanced SPMF (99.10%) is better than that of SPMF (98.56%). Compared to skeleton-based ResNet, there is no improvement on this dataset.

TABLE 4.14: Experimental results and comparison of the proposed method with state-the-art approaches on the MSR Action3D dataset (Li, Zhang, and Liu, 2010).

Method (protocol of Li, Zhang, and Liu, 2010)	**Year**	**AS1**	**AS2**	**AS3**	**Aver.**
Bag of 3D Points (Li, Zhang, and Liu, 2010)	2010	72.90%	71.90%	71.90%	74.70%
Depth Motion Maps (Chen, Liu, and Kehtarnavaz, 2013)	2016	96.20%	83.20%	92.00%	90.47%
Bi-LSTM (Tanfous, Drira, and Amor, 2018)	2018	92.72%	84.93%	97.89%	91.84%
Lie Group (Vemulapalli, Arrate, and Chellappa, 2014)	2014	95.29%	83.87%	98.22%	92.46%
FTP-SVM (Tanfous, Drira, and Amor, 2018)	2018	95.87%	86.72%	100.0%	94.19%
Hierarchical LSTM (Du, Wang, and Wang, 2015)	2015	99.33%	94.64%	95.50%	94.49%
ST-LSTM Trust Gates (Liu et al., 2016b)	2016	N/A	N/A	N/A	94.80%
Graph-Based Motion (Wang et al., 2016b)	2016	93.60%	95.50%	95.10%	94.80%
ST-NBNN (Weng, Weng, and Yuan, 2017)	2017	91.50%	95.60%	97.30%	94.80%
ST-NBMIM (Weng et al., 2018)	2018	92.50%	95.60%	98.20%	95.30%
S-T Pyramid (Xu et al., 2015a)	2015	99.10%	92.90%	96.40%	96.10%
Ensemble TS-LSTM v2 (Li et al., 2017b)	2017	95.24%	96.43%	100.0%	97.22%
Skeleton-based ResNet (section 4.1)	2018	**99.90**%	**99.80**%	**100.0**%	**99.90**%
SPMF Inception-ResNet-222 (section 4.2)	2018	97.54%	98.73%	99.41%	98.56%
Enhanced-SPMF DenseNet (L = 100, k = 12)	2018	98.52%	98.66%	99.09%	98.76%
Enhanced-SPMF DenseNet (L **= 250**, k **= 24**)	2018	**98.83**%	**99.06**%	**99.40**%	**99.10**%
Enhanced-SPMF DenseNet (L = 190, k = 40)	2018	98.60%	98.87%	99.36%	98.94%

Results on SBU Kinect Interaction dataset

As reported in TABLE 4.15, the proposed DenseNet (L = 250, k = 40) achieved an accuracy of 97.86% and outperforms many existing state-of-the-art approaches including Raw Skeleton (Yun et al., 2012a), Joint Features (Yun et al., 2012a), HBRNN (Du, Wang, and Wang, 2015), CHARM (Li et al., 2015), Deep LSTM (Zhu et al., 2016c), Joint Features (Ji, Ye, and Cheng, 2014), ST-LSTM (Liu et al., 2016b), Co-occurrence+Deep LSTM (Zhu et al., 2016c), STA-LSTM (Song et al., 2017), ST-LSTM+Trust Gates (Liu et al., 2016b), ST-NBMIM (Weng et al., 2018), Clips+CNN+MTLN (Ke et al., 2017), Two-stream RNN (Wang and Wang, 2017), and GCA-LSTM network (Liu et al., 2018). Using only skeleton modality, the proposed method outperforms hand-crafted feature based approaches such as Raw Skeleton (Yun et al., 2012a), Joint Features (Yun et al., 2012a) and recent state-of-the-art RNN-based approaches (Du, Wang, and Wang, 2015; Zhu et al., 2016c; Liu et al., 2016b; Song et al., 2017; Wang and Wang, 2017; Liu et al., 2018). In particular, the proposed method achieves a significant accuracy gain of 2.96% compared to the nearest competitor GCA-LSTM network (Liu et al., 2018). This result demonstrates that the proposed deep learning framework is able to learn discriminative spatio-temporal features of skeleton joints containing in the proposed motion representation for classification task. Since skeleton-based ResNet (section 4.1) and SPMF Inception-ResNet-v2 (section 4.2) were not applied to this dataset, there is no comparison between them and Enhanced-SPMF.

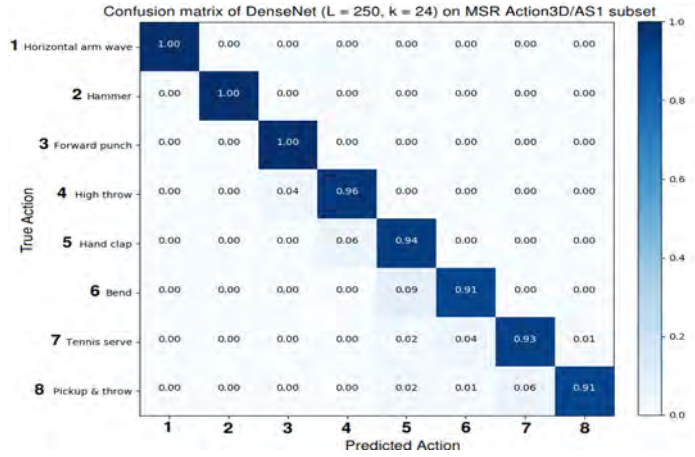


FIGURE 4.24: Confusion matrix of the proposed DenseNet ($L = 250$, $k = 24$) on the MSR Action3D/AS1 dataset. Ground truth action labels are on rows and predictions by the proposed method are on columns. We recommend the readers to use a computer and zoom in to see clearly these figures.

TABLE 4.15: Action recognition accuracies (%) and comparison with previous works on the SBU Kinect Interaction dataset (Yun et al., 2012a).

Method (protocol of Yun et al., 2012a)	**Year**	**Acc. (%)**
Raw Skeleton (Yun et al., 2012a)	2012	49.70%
Joint Features (Yun et al., 2012a)	2012	80.30%
HBRNN (Du, Wang, and Wang, 2015)	2015	80.40%
CHARM (Li et al., 2015)	2015	83.90%
Deep LSTM (Zhu et al., 2016c)	2017	86.03%
Joint Features (Ji, Ye, and Cheng, 2014)	2014	86.90%
ST-LSTM (Liu et al., 2016b)	2016	88.60%
Co-occurrence+Deep LSTM (Zhu et al., 2016c)	2018	90.41%
STA-LSTM (Song et al., 2017)	2017	91.51%
ST-LSTM+Trust Gates (Liu et al., 2016b)	2018	93.30%
ST-NBMIM (Weng et al., 2018)	2018	93.30%
Clips+CNN+MTLN (Ke et al., 2017)	2017	93.57%
CNN Kernel Feature Map (Tas and Koniusz, 2018)	2018	94.36%
Two-stream RNN (Wang and Wang, 2017)	2017	94.80%
GCA-LSTM network (Liu et al., 2018)	2018	94.90%
Enhanced-SPMF DenseNet ($L = 100$, $k = 12$)	2018	94.81%
Enhanced-SPMF DenseNet ($L = 250$, $k = 24$)	2018	96.67%
Enhanced-SPMF DenseNet (L **= 190**, k **= 40**)	2018	**97.86%**

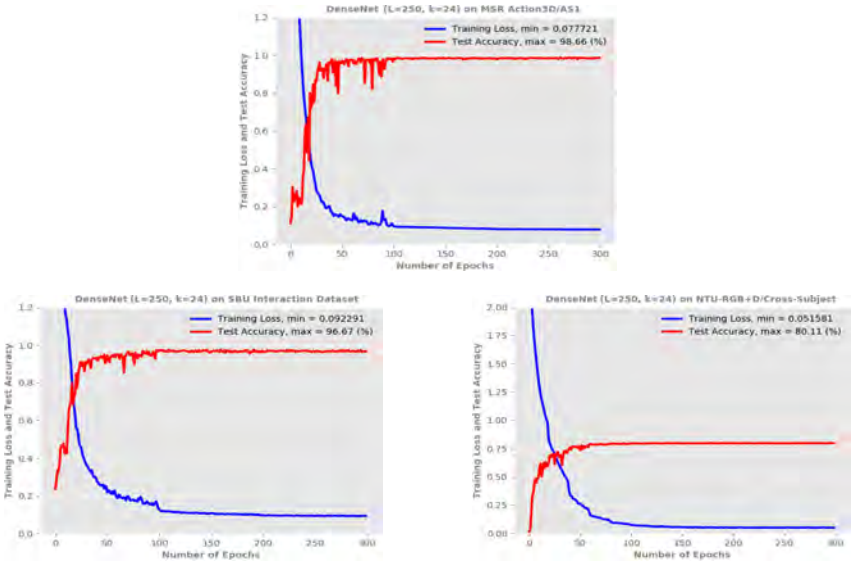


FIGURE 4.25: Training curves of the proposed DenseNet (L = 250, k = 24) on the MSR Action3D (Li, Zhang, and Liu, 2010), SBU Kinect Interaction (Yun et al., 2012a), and NTU-RGB+D (Shahroudy et al., 2016) datasets. Almost all designed networks are able to reach the optimal weights after the first 100 epochs. The symbols k and L and denote the *"growth rate"* and the depth of the network, respectively.

Results on NTU-RGB+D dataset

For the NTU-RGB+D dataset Shahroudy et al., 2016, the best configuration DenseNet (L = 250, k = 40) achieves an accuracy of 80.11% on the Cross-Subject evaluation and 86.82% on the Cross-View evaluation, as summarized in TABLE 4.16. These results demonstrate the effectiveness of the proposed representation and deep learning framework since they surpass previous state-of-the-art techniques such as Lie Group Representation (Vemulapalli, Arrate, and Chellappa, 2014), Hierarchical RNN (Du, Wang, and Wang, 2015), Dynamic Skeletons (Hu et al., 2015b), Two-Layer P-LSTM (Shahroudy et al., 2016), ST-LSTM Trust Gates (Liu et al., 2016b), Geometric Features (Zhang, Liu, and Xiao, 2017), Two-Stream RNN (Wang and Wang, 2017), Enhanced Skeleton (Liu, Liu, and Chen, 2017), Lie Group Skeleton+CNN (Rahmani and Bennamoun, 2017), and GCA-LSTM (Liu et al., 2018). The experimental results have also shown that the proposed method leads to better overall action recognition performance than our previous models including Skeleton-based ResNet (section 4.1) and SPMF Inception-ResNet-222 (section 4.2). With a high recognition rate on the Cross-View evaluation (86.82%) where the sequences provided by cameras 2 and 3 are used for training and sequences from camera 1 are used for test, the proposed method shows its effectiveness for dealing with the view-independent action recognition problem. FIGURE 4.25 shows the training loss and test accuracy of the DenseNet (L = 250, k = 24) on this dataset.

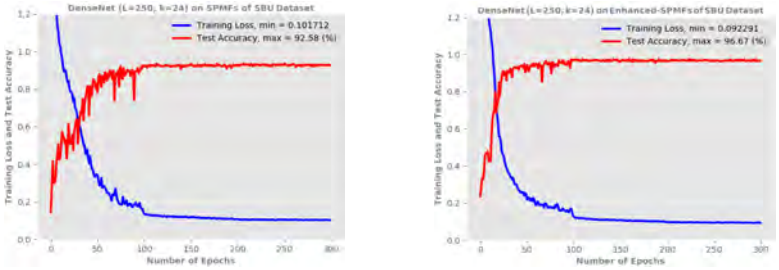
An ablation study on the Enhanced-SPMF representation

We believe that the use of the AHE algorithm (Pizer et al., 1987) and the Savitzky-Golay smoothing filter (Savitzky and Golay, 1964; Du, Wang, and Wang, 2015) helps the proposed representation to be more discriminative, which improves recognition accuracy. To verify this hypothesis, we carried out an ablation study on the Enhanced-SPMF representation provided by the SBU Kinect Interaction dataset (Yun et al., 2012a). Specifically, we trained the proposed DenseNet ($L = 250, k = 24$) on both the SPMFs and Enhanced-SPMFs. During training, the same hyper-parameters and training methodology were applied. The experimental results indicate that the proposed deep network achieves better recognition accuracy when trained on the Enhanced-SPMFs. As reported in FIGURE 4.26, applying the AHE algorithm (Pizer et al., 1987) and and the Savitzky-Golay smoothing filter (Savitzky and Golay,

TABLE 4.16: Experimental results and comparison of the proposed method with previous approaches on the NTU-RGB+D dataset (Shahroudy et al., 2016).

Method (protocol of Shahroudy et al., 2016)	**Year**	**Cross-Subject**	**Cross-View**
Lie Group (Vemulapalli, Arrate, and Chellappa, 2014)	2014	50.10%	52.80%
Hierarchical RNN (Du, Wang, and Wang, 2015)	2016	59.07%	63.97%
Dynamic Skeletons (Hu et al., 2015b)	2015	60.20%	65.20%
Two-Layer P-LSTM (Shahroudy et al., 2016)	2016	62.93%	70.27%
ST-LSTM Trust Gates (Liu et al., 2016b)	2016	69.20%	77.70%
Geometric Features (Zhang, Liu, and Xiao, 2017)	2017	70.26%	82.39%
Two-Stream RNN (Wang and Wang, 2017)	2017	71.30%	79.50%
Enhanced Skeleton (Liu, Liu, and Chen, 2017)	2017	75.97%	82.56%
Lie Group+CNN (Rahmani and Bennamoun, 2017)	2017	75.20%	83.10%
CNN Kernel Feature Map (Tas and Koniusz, 2018)	2018	75.35%	N/A
GCA-LSTM (Liu et al., 2018)	2018	76.10%	84.00%
Skeleton-based ResNet (section 4.1)	2018	78.20%	85.60%
SPMF Inception-ResNet-222 (section 4.2)	2018	78.89%	86.15%
Enhanced-SPMF DenseNet ($L = 100$, $k = 12$)	2018	79.31%	86.64%
Enhanced-SPMF DenseNet ($L = 250$, $k = 24$)	2018	**80.11%**	**86.82%**
Enhanced-SPMF DenseNet ($L = 190$, $k = 40$)	2018	79.28%	86.68%

1964; Du, Wang, and Wang, 2015) helps improving the accuracy by 4.09%. This result validates our hypothesis above.



(a) (b)

FIGURE 4.26: Training loss and test accuracy of the proposed DenseNet ($L = 100$, $k = 12$) on the SBU dataset (Yun et al., 2012a). FIGURE 4.26a shows the obtained result when trained on SPMFs, while FIGURE 4.26b reports the obtained result when trained on Enhanced-SPMFs. The symbols k and L denote the *"growth rate"* and the depth of the network, respectively.

Visualization of deep feature maps

Different action classes have different discriminative characteristics. To better understand the internal operation of the proposed deep networks and to study what they learned from the skeleton-based representation, we input different Enhanced-SPMFs corresponding to different action classes of the MSR Action3D dataset (Li, Zhang, and Liu, 2010) to the DenseNet ($L = 100$, $k = 12$) and visualize the individual feature maps learned by the network at the end of a dense block (intermediate layer). We observe that the designed network is able to extract discriminative features from the Enhanced-SPMF representations. This is expressed through the color of each learned feature map, as can be seen in FIGURE 4.27. These discriminative features play a key role in classifying actions.

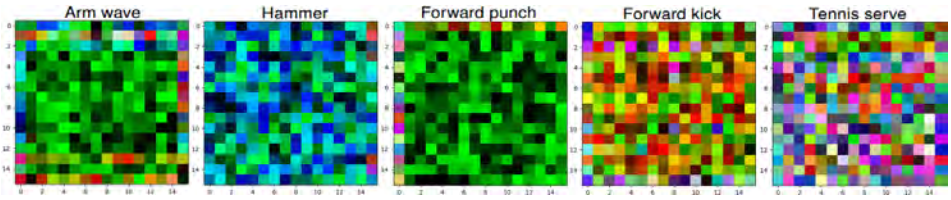


FIGURE 4.27: Visualization of feature maps learned by the proposed DenseNet ($L = 100$, $k = 12$) from several samples of the MSR Action3D dataset (Li, Zhang, and Liu, 2010). Best viewed in color.

Computational efficiency evaluation

In this section, we take the AS1 subset of MSR Action3D dataset (Li, Zhang, and Liu, 2010) and the DenseNet ($L = 100$, $k = 12$) to evaluate the computational efficiency of the proposed method. FIGURE 4.28 illustrates three main stages of the deep learning framework for learning and recognizing actions from skeleton sequences, including an encoding process from input skeleton sequences to color images (stage 1); a supervised training stage (stage 2[11]); and an inference stage (stage 3). With the implementation in Python using Keras and training on a single GeForce GTX 1080 Ti GPU, the proposed deep network that only has 6.0M parameters takes less than six hours to reach convergence. During this stage, it takes 0.164 seconds per skeleton sequence. Latency required to predict a new skeleton sequence using the pre-trained model, including the stage 1 that is executed on a CPU and the stage 3 is about 74.8×10^{-3} seconds per sequence. Additionally, it should be noted that the computation of the Enhanced-SPMFs can be implemented and optimized on a GPU for faster processing.

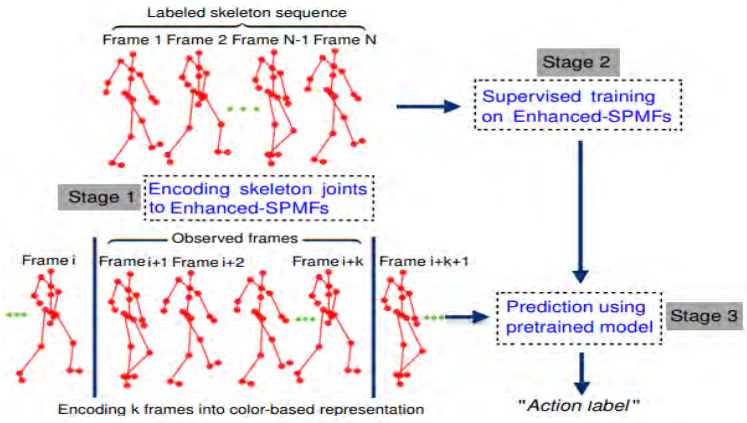


FIGURE 4.28: Three main stages of the proposed deep learning framework for recognizing human actions from skeleton sequences.

TABLE 4.17: Execution time of the proposed deep learning framework.

Stage	Average processing time (second/sequence)		
1	20.8×10^{-3}	per sequence	(Intel Core i7 3.2GHz CPU)
2	0.164	per sequence	(GTX 1080 Ti GPU)
3	74.8×10^{-3}	per sequence	(CPU + GPU time)

[11] Including the color enhancement process: SPMF $\rightarrow$ Enhanced-SPMF

Limitations

The use of the Savitzky-Golay filter (Savitzky and Golay, 1964) helps reduce the effect of noise on the raw skeleton sequences. However, the proposed approach cannot overcome the problem of missing data. In other words, as the Enhanced-SPMF is a global representation for the whole skeleton sequence, data error of local fragments in the input sequences could cut down the recognition rate. Another open problem of the proposed approach is how to scope with Online Action Recognition (OAR) task. Specifically, how to detect and recognize human actions from unsegmented streams in a continuous manner, where boundaries between different kinds of actions within the stream are unknown. A common solution for OAR is the sliding window based methods (Kulkarni et al., 2015; Kviatkovsky, Rivlin, and Shimshoni, 2014). These approaches consider the temporal coherence within the window for prediction. We can also apply this idea to solve the current problem. *E.g.*, during the online inference phase, we use a sliding window on the original skeleton sequences or on image-coded representations (*i.e.* Enhanced-SPMFs) and then predicting action by pretrained deep learning model, as we showed in FIGURE 4.28 (stage 3). However, we understand that the performance of this approach is sensitive to the window size. Either too large or too small window size could lead to a significant drop in recognition performance. Another solution is to use Temporal Attention Networks (Mnih, Heess, and Graves, 2014; Xu et al., 2015b; Luong, Pham, and Manning, 2015; Zang et al., 2018) that incorporates temporal attention model for video-based action recognition.

4.3.5 Conclusion

This section presents an efficient and effective deep learning framework for 3D human action recognition from skeleton sequences. An advanced motion representation, called Enhanced-SPMF, which captures the spatio-temporal information of skeleton movements and transforms them into color images has been proposed. Different Deep Convolutional Neural Networks (D-CNNs) based on the DenseNet architecture have been designed and optimised to learn and recognize actions from the proposed representation, in an end-to-end manner. We used the Adaptive Histogram Equalization (AHE) technique to enhance the local textures of color images and generate more discriminative features for learning and classification tasks. Extensive empirical evaluations on three challenging public datasets demonstrate the effectiveness of the proposed approach on both individual actions, interactions, multi-view and large-scale datasets.

Since the beginning of our work and in order to compare it to the state of the art, we have tested our methods on several academic datasets with existing ground truth. Cerema, which is an institution belonging to Ministry of Ecology and Transport, has often the opportunity to work with transport operators. During this PhD thesis, we had the opportunity to work with Tisséo, main operator of Toulouse transport network. The initial idea is to implement RGB-D sensors to detect some specific passengers' behaviour like people jumping over the turnstiles, sneaking under the turnstiles, and so on. All this is described in section 4.4.

4.4 CEMEST dataset

4.4.1 Introduction to CEMEST dataset

We have collected a new RGB-D dataset[12], called CEMEST (CErema MEtro STation dataset) using Kinect v2 sensor and carried out experiments on this dataset to verify the effectiveness of the proposed method on a real-world dataset. The CEMEST was made at a metro station in France without any control of the passenger behavior as well as illumination. It contains three actions including both "*normal*" and "*abnormal*" behaviors: *crossing normally over the turnstiles*, *jumping over the turnstiles*, and *sneaking under the turnstiles*. These three behaviors are taken into account for acquisition because they have a significant impact on

[12]The dataset and its description are available at: https://sites.google.com/site/hhpham172/image-processing-and-computer-vision/tisseo-cerema-dataset.

monitoring and management in public transport. As an example, the French National Railway Company (SNCF) reported that they lost €500 million every year through people trying to cheat the ticket system (The Local, 2015). In summary, this dataset provides RGB, depth and skeletal data. The skeleton sequences are extracted by Kinect SDK with 25 key joints for each subject, at a frame rate of 30 FPS. All recorded sequences are manually segmented and labeled. FIGURE 4.29 shows some samples from the CEMEST.

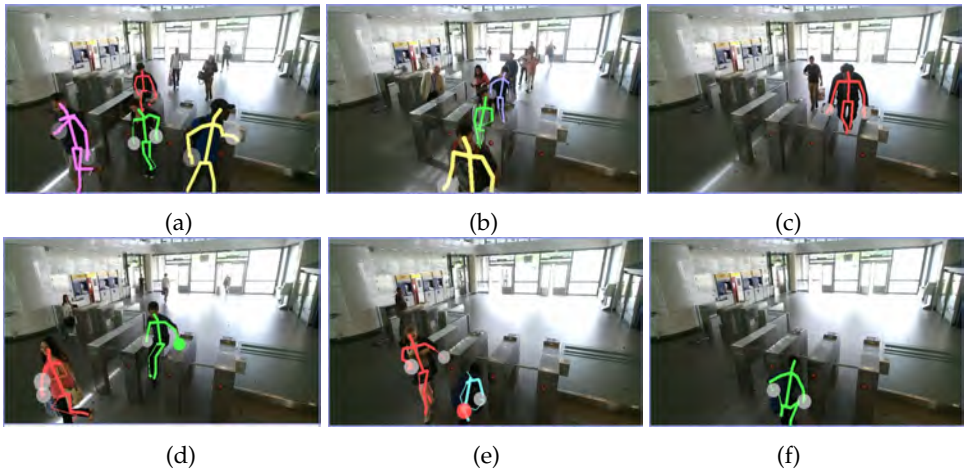


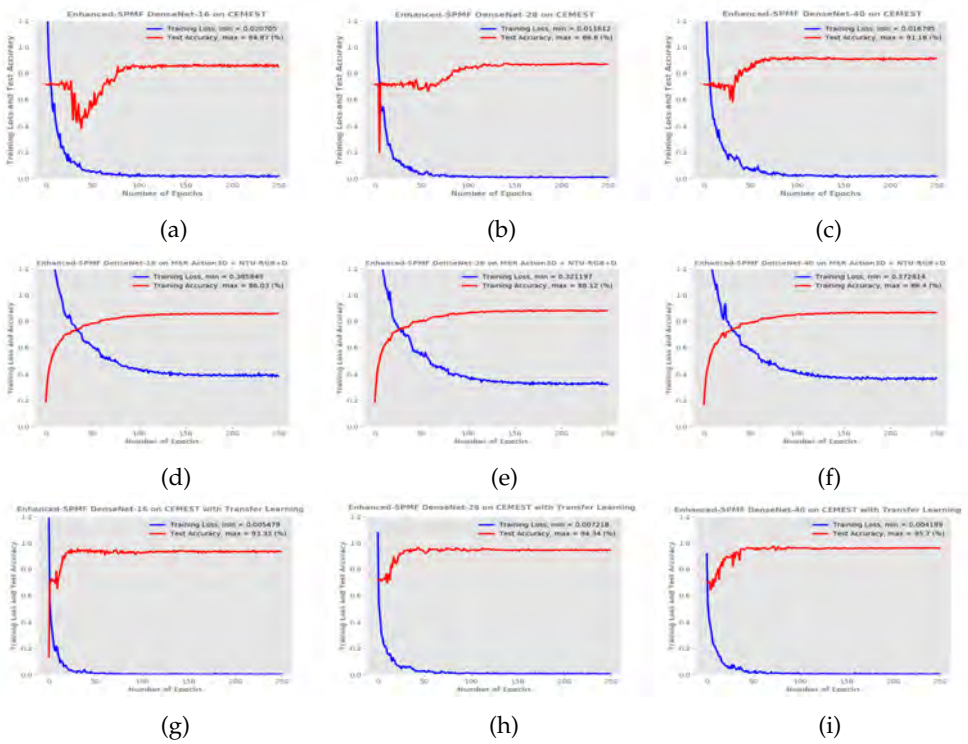
FIGURE 4.29: Some samples from the CEMEST dataset: **(a)**, **(b)** *crossing over the barriers*; **(c)**, **(d)** *jumping over the ticket barriers*; **(e)**, **(f)** *sneaking under ticket barriers*.

4.4.2 Experiments on CEMEST

We carried out two experimental evaluations on this dataset. In the first setting, we randomly chose 67% of the data as training set and the remaining 33% is used for testing. In the second setting, the proposed networks are trained on a combination dataset, which is created from a portion of the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) and NTU RGB+D (Shahroudy et al., 2016) datasets. The full list of action classes in the combination dataset is provided in APPENDIX A2. To ensure the number of samples in each action class is balanced, we augmented samples in the MSR Action3D to match the size of the larger dataset. The pre-trained model is then deployed on the CEMEST dataset in the hope that transfer learning will help to solve overfitting problem when training on small dataset. In both experiments, data augmentation (*i.e.* cropping, flipping, Gaussian filtering) has been used.

4.4.3 Experimental results

On the CEMEST dataset, an accuracy of 91.18% has been made by the DenseNet-40 in the first setting. In the second setting, transfer learning is used. The experimental results show that the proposed method reached an accuracy of 95.70%, increasing the performance by nearly 5% compared to the first experiment. This could be explained by the fact that since the CEMEST dataset is quite small, it benefits from the knowledge transfer coming from larger datasets such as the MSR Action3D and NTU RGB+D datasets. This result indicates that the use of data augmentation and transfer learning is crucial to address the small amount of samples in real-world datasets. FIGURE 4.30 shows learning curves of the proposed deep learning networks on the CEMEST dataset from scratch (FIGURE 4.30a – FIGURE 4.30c), pre-training on the combined dataset (FIGURE 4.30d – FIGURE 4.30f) and fine-tuning on CEMEST dataset (FIGURE 4.30g – FIGURE 4.30i).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 4.30: Learning curves of the three proposed deep networks (DenseNet-16, DenseNet-28, DenseNet-40) on CEMEST dataset when trained from scratch **(a)-(b)-(c)**. Pre-trained on the combined dataset **(d)-(e)-(f)**; and fine-tuned on CEMEST dataset **(g)-(h)-(i)**. Our best configuration (DenseNet-40) achieved an accuracy of 91.18% when trained on the CEMEST dataset from scratch. With the support of transfer learning, the proposed method reached an accuracy of 95.70%, increasing the recognition accuracy by nearly 5%.

4.4.4 Conclusion

Our proposed deep learning framework for 3D action recognition from skeletal data was valided on a real-world dataset (CEMEST) containing normal and abnormal human behaviours. Experimental results on this dataset show that the proposed deep learning-based approach achieved promising results. One limitation of the CEMEST dataset is that it has limited action classes (*e.g. crossing normally over the turnstiles*, *jumping over the turnstiles*, and *sneaking under the turnstiles*). We plan to extend it with more action categories, which have a significant impact on public safety such as *fighting*, *stealing*, *falling down*, *accident*, etc. and under multiple viewpoints.

During the data collection period, we discovered the limits of RGB-D sensors when used in this semi-open environment with a high-level of illumination. We also noticed that depth sensors are only able to operate up to a limited distance and within a limited field of view. We therefore decided, when the RGB-D sensors are not suitable for some specific environments, to estimate the 3D human poses from RGB sensors which are less sensitive to high level of illumination and to long distances. All this work is described in the next chapter.

Chapter 5

A Unified Deep Learning Framework for Joint 3D Pose Estimation and Action Recognition from a Single RGB Camera

Contents

Chapter overview: This chapter describes the possibility to directly extract 3D skeletons from RGB sensors which are installed in many different sites. If we succeed to have accurate 3D skeletal data we will be able to benefit from all the frameworks developed and described in the previous chapter such as the SPMF, Enhanced SPMF as well as the proposed deep learning framework. The aim of this chapter is therefore to propose a 3D skeleton-based action recognition approach without depth sensors. Specifically, we present a deep learning-based multitask framework for joint 3D human pose estimation and action recognition from RGB video sequences. Once the 3D estimated poses obtained from RGB sensors, the idea is to use all the frameworks developed previously with the RGB-D sensors. Our approach proceeds along two stages. First, we run a real-time 2D pose detector to determine the precise pixel location of important keypoints of the body. A two-stream neural network is then designed and trained to map detected 2D keypoints into 3D poses. Second, we deploy the Efficient Neural Architecture Search (ENAS – Pham et al., 2018a) algorithm to find an optimal network architecture that is used for modeling the spatio-temporal evolution of the

estimated 3D poses via an image-based intermediate representation and performing action recognition. Experiments on Human3.6M (Ionescu et al., 2014), MSR Action3D (Li, Zhang, and Liu, 2010), and SBU Kinect Interaction (Yun et al., 2012a) datasets verify the effectiveness of the proposed method on the targeted tasks. Moreover, we show that the proposed method requires a low computing cost for training and inference.

5.1 Introduction

The rapid development of depth-sensing time-of-flight camera technology has helped in dealing with this problem, which is considered complex for traditional cameras. Low-cost and easy-to-use depth cameras are able to provide detailed 3D structural information of human motion. In particular, most of the current depth cameras have integrated real-time skeleton estimation and tracking frameworks (Ye and Yang, 2014), facilitating the collection of skeletal data. This is a high-level representation of the human body, which is suitable for the problem of motion analysis. Hence, exploiting skeletal data for 3D action recognition opens up opportunities for addressing the limitations of RGB-based solutions and many skeleton-based action recognition approaches have been proposed (Wang et al., 2012; Xia, Chen, and Aggarwal, 2012a; Chaudhry et al., 2013; Vemulapalli, Arrate, and Chellappa, 2014; Ding et al., 2016). However, depth sensors have some significant drawbacks with respect to 3D pose estimation. For instance, they are only able to operate up to a limited distance (0.5m – 4.5m) and within a limited field of view. Moreover, a major drawback of depth cameras is the inability to work in very illuminated scenes, especially with sunlight (Zhang, 2012).

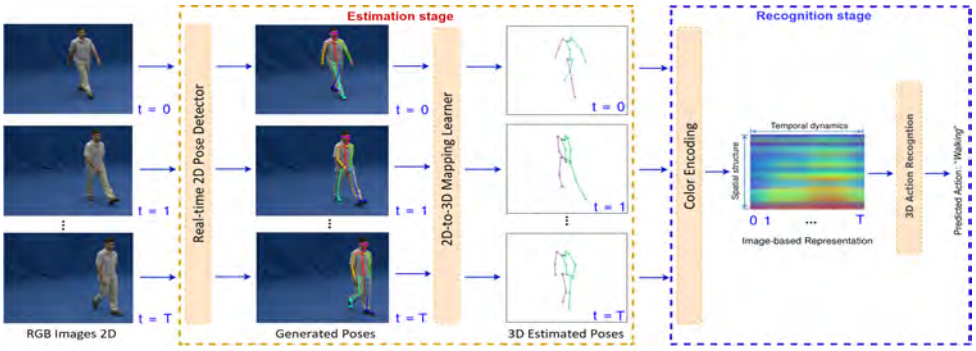


FIGURE 5.1: Overview of the proposed method. In the estimation stage, we first run OpenPose (Cao et al., 2017) – a real-time, state-of-the-art multi-person 2D pose detector to generate 2D human body keypoints. A deep neural network is then trained to produce 3D poses from the 2D detections. In the recognition stage, the 3D estimated poses are encoded into a compact image-based representation and finally fed into a deep convolutional network for supervised classification task, which is automatically searched by the ENAS algorithm (Pham et al., 2018a).

The main goal of this chapter is therefore to propose a 3D skeleton-based action recognition approach without depth sensors. Specifically, we are interested in building a unified deep framework for both 3D pose estimation and action recognition from RGB video sequences. As shown in FIGURE 5.1, our approach consists of two stages. In the first one, *estimation stage*, the system recovers the 3D human poses from the input RGB video. In the second one, *recognition stage*, an action recognition approach is developed and stacked on top of the 3D pose estimator in a unified framework, where the estimated 3D poses are used as inputs to learn the spatio-temporal motion features and predict action labels.

In the literature, state-of-the-art 2D pose detectors (*e.g.* Cao et al., 2017; Newell, Yang, and Deng, 2016) are able to provide 2D poses with a high degree of accuracy in real-time.

Meanwhile, deep networks have proved their capacity to learn complex functions from high-dimensional data. Hence, a simple network model can also learn a *mapping* to convert 2D poses into 3D.

The effectiveness of the proposed method is evaluated on public benchmark datasets: Human3.6M (Ionescu et al., 2014), MSR Action3D (Li, Zhang, and Liu, 2010), and SBU Kinect Interaction (Yun et al., 2012a). Far beyond our expectations, the experimental results demonstrate state-of-the-art performances on the targeted tasks and support our hypotheses above. Furthermore, we show that this approach has a low computational cost. More precisely, our main contributions in this chapter are the followings:

• **First**, we present a two-stream, lightweight neural network to recover 3D human poses from RGB images provided by a monocular camera. Our proposed method achieves state-of-the-art result on 3D human pose estimation task and benefits action recognition.

• **Second**, we propose to put an action recognition approach on top of the 3D pose estimator to form a unified framework for 3D pose-based action recognition. It takes the 3D estimated poses as inputs, encodes them into a compact image-based representation and finally feeds to a deep convolutional network, which is designed automatically by using a neural architecture search algorithm. Surprisingly, our experiments show that we reached state-of-the-art results on this task, even when compared with methods using depth cameras.

The rest of this chapter is organized as follows. We present a review of the related work in Section 5.2. The proposed method is explained in Section 5.3. Experiments are provided in Section 5.4 and Section 5.5 concludes the chapter.

5.2 Related work

This section reviews two main topics that are directly related to ours, *i.e.* performing 3D pose estimation from RGB images and using the 3D estimated poses from RGB sensors for the problem of human action recognition.

5.2.1 3D human pose estimation from a single RGB camera

The problem of 3D human pose estimation has been intensively studied in the recent years. Almost all early approaches for this task were based on feature engineering (Sminchisescu, 2006; Ramakrishna, Kanade, and Sheikh, 2012; Ionescu et al., 2014), while the current state-of-the-art methods are based on deep neural networks (Li and Chan, 2014; Tekin et al., 2016; Pavlakos et al., 2017; Pavllo et al., 2018; Mehta et al., 2017b; Katircioglu et al., 2018). Many of them are regression-based approaches that directly predict 3D poses from RGB images via 2D/3D heatmaps. For instance, Li and Chan, 2014 designed a deep convolutional network for human detection and pose regression. The regression network learns to predict 3D poses from single images using the output of a body part detection network. Tekin et al., 2016 proposed to use a deep network to learn a regression mapping that directly estimates the 3D pose in a given frame of a sequence from a spatio-temporal volume centered on it. Pavlakos et al., 2017 used multiple fully convolutional networks to construct a volumetric stacked hourglass architecture, which is able to recover 3D poses from RGB images. Pavllo et al., 2018 exploited a temporal dilated convolutional network (Fisher and Vladlen, 2015) for estimating 3D poses. However, this approach led to a significant increase in the number of parameters as well as the required memory. Mehta et al., 2017b introduced a real-time approach to predict 3D poses from a single RGB camera. They used ResNets (Kaiming et al., 2016) to jointly predict 2D and 3D heatmaps as regression tasks. Recently, Katircioglu et al., 2018 introduced a deep regression network for predicting 3D human poses from monocular images via 2D joint location heatmaps. This architecture is in fact an overcomplete autoencoder that learns a high-dimensional latent pose representation and accounts for joint dependencies, in which a Long Short-Term Memory network (Hochreiter and Schmidhuber, 1997) is used to enforce temporal consistency on 3D pose predictions.

To the best of our knowledge, several studies (Pavlakos et al., 2017; Mehta et al., 2017b; Katircioglu et al., 2018) stated that regressing the 3D pose from 2D joint locations is difficult and not enough accurate. However, motivated by Martinez et al., 2017, we believe that a simple neural network can learn effectively a *direct 2D-to-3D mapping*. Therefore, this work aims at proposing a simple, effective and real-time approach for 3D human pose estimation that benefits action recognition. To this end, we design and optimize a two-stream deep neural network that performs 3D pose predictions from the 2D human poses. These 2D poses are generated by a state-of-the-art 2D detector that is able to run in real-time for multiple people. We empirically show that although the proposed approach is computationally inexpensive, it is still able to improve the state-of-the-art.

5.2.2 3D pose-based action recognition from RGB sensors

In the literature, 3D human pose estimation and action recognition are closely related. However, both problems are generally considered as two distinct tasks (Chéron, Laptev, and Schmid, 2015). Although some approaches have been proposed for tackling the problem of jointly predicting 3D poses and recognizing actions in RGB images or video sequences (Yao and Fei-Fei, 2010; Nie, Xiong, and Zhu, 2015; Luvizon, Picard, and Tabia, 2018), they are data-dependent and require a lot of feature engineering, except the work of Luvizon, Picard, and Tabia, 2018. Unlike in previous studies, we propose a multitask learning frameworks for 3D pose-based action recognition by reconstructing 3D skeletons from RGB images and exploiting them for action recognition in a joint way. Experimental results on public and challenging datasets show that our framework is able to solve the two tasks in an effective way.

5.3 Proposed method

In this section, our approach for 3D human pose estimation is presented. We then introduce our solution for 3D pose-based action recognition.

5.3.1 Problem definition

Given an RGB video clip of a person who starts to perform an action at time $t = 0$ and ends at $t = T$, the problem studied in this work is to generate a sequence of 3D poses $\mathcal{P} = (\mathbf{p}_0, ..., \mathbf{p}_T)$, where $\mathbf{p}_i \in \mathbb{R}^{3\times M}$, $i \in \{0, ..., T\}$ at the estimation stage. The generated $\mathcal{P}$ is then used as input for the recognition stage to predict the corresponding action label $\mathcal{A}$ by a supervised learning model. See FIGURE 5.1 for an illustration of the problem.

5.3.2 Deep learning model for 3D human pose estimation from RGB images

Given an input RGB image $\mathbf{I} \in \mathbb{R}^{W\times H\times 3}$, we aim to estimate the body joint locations in the 3-dimensional space, noted as $\hat{\mathbf{p}}_{3D} \in \mathbb{R}^{3\times M}$. To this end, we first run the state-of-the-art human 2D pose detector, namely OpenPose, which is based on a multi-stage CNN algorithm (Cao et al., 2017), to produce a series of 2D keypoints $\mathbf{p}_{2D} \in \mathbb{R}^{2\times N}$. To recover the 3D joint locations, we try to learn a *direct 2D-to-3D mapping* f_r: $\mathbf{p}_{2D} \overset{f_r}{\mapsto} \hat{\mathbf{p}}_{3D}$. This transformation can be implemented by a deep neural network in a supervised manner

$$\hat{\mathbf{p}}_{3D} = f_r(\mathbf{p}_{2D}, \theta), \tag{5.1}$$

where θ is a set of trainable parameters of the function f_r. To optimize f_r, we minimize the prediction error over a labelled dataset of $\mathcal{C}$ poses by solving the optimization problem

$$\arg\min_{\theta} \frac{1}{\mathcal{C}} \sum_{n=1}^{\mathcal{C}} \mathcal{L}(f_r(\mathbf{x}_i), \mathbf{y}_i). \tag{5.2}$$

Here $\mathbf{x}_i$ and $\mathbf{y}_i$ are the input 2D poses and the ground truth 3D poses, respectively; $\mathcal{L}$ denotes a loss function. In our implementation the robust Huber loss (Huber, 1992) is used to deal with outliers.

Network design

State-of-the-art deep learning architectures such as ResNet (Kaiming et al., 2016), Inception-ResNet-v2 (Szegedy et al., 2015a), DenseNet (Huang et al., 2017), or NASNet (Barret and Quoc, 2017) have achieved an impressive performance in supervised learning tasks with high dimensional data, *e.g.* 2D or 3D images. However, the use of these architectures on low dimensional data like the coordinates of the 2D human joints could lead to overfitting. Therefore, our design is based on a simple and lightweight multilayer network architecture without the convolution operations. In the design process, we exploit some recent improvements in the optimization of the modern deep learning models (Kaiming et al., 2016; Huang et al., 2017). Concretely, we propose a two-stream network. Each stream comprises linear layers, Batch Normalization (BN – Ioffe and Szegedy, 2015), Dropout (Hinton et al., 2012), SELU (Klambauer et al., 2017) and Identity connections (Kaiming et al., 2016). During the training phase, the first stream takes the ground truth 2D locations as input. The 2D human joints predicted by OpenPose (Cao et al., 2017) are inputted to the second stream. The outputs of the two streams are then averaged. FIGURE 5.2 illustrates our network design. Note that learning with the ground truth 2D locations for both of these streams could lead to a higher level of performance. However, training with the 2D OpenPose detections could improve the generalization ability of the network and makes it more robust during the inference, when only the 2D output of the OpenPose is used to deal with action recognition in the wild.

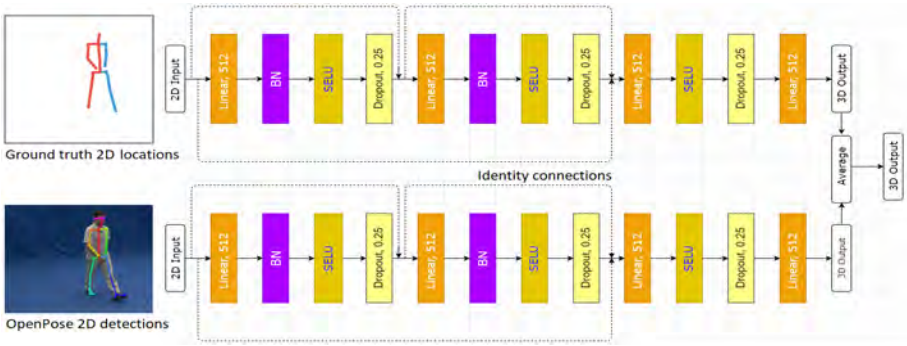


FIGURE 5.2: Diagram of the proposed two-stream network for training our 3D pose estimator.

5.3.3 Deep learning framework for 3D pose-based action recognition

In this section, we explain how to integrate the estimation stage with the recognition stage in a unified framework. Specifically, the proposed recognition approach is stacked on top of the 3D pose estimator. To explore the high-level information of the estimated 3D poses, we encode them into a compact image-based representation. These intermediate representations are then fed to a Deep Convolutional Neural Network (D-CNNs) for learning and classifying actions. More specifically, the spatio-temporal patterns of a 3D pose sequence are transformed into a single color image as a global representation via the proposed Enhanced-SPMF (see section 4.2).

In chapter 4, we have used three architectures: Resnet, Inception-Resnet-v2 and DenseNet. The state of the art in deep learning for recognition tasks is moving very fast. When we began the work described in chapter 5, we discovered that some new architectures are setting new state of the art on some common recognition datasets (e.g. CIFAR-10 (Krizhevsky, 2009) or ImageNet (Krizhevsky, Sutskever, and Hinton, 2012b)). Then, we therefore decided to use

them instead of the previous architectures. More specifically, for learning and classifying the obtained images, we propose to use the Efficient Neural Architecture Search (ENAS – Pham et al., 2018a) – a recent state-of-the-art technique for automatic design of deep neural networks. The ENAS is in fact an extension of an important advance in deep learning called NAS (Barret and Quoc, 2017), which is able to automatize the designing process of convolutional architectures on a dataset of interest. This method proposes to search for optimal building blocks (called *cells*, including *normal cells* and *reduction cells*) and the final architecture is then constructed from the best cells achieved. In NAS, an RNN is used. It first samples a candidate architecture called *child model*. This child model is then trained to convergence on the desired task and reports its performance. Next, the RNN uses the performance as a guiding signal to find a better architecture. This process is repeated for many times, making NAS computationally expensive and time-consuming (*e.g.* on CIFAR-10 (Krizhevsky, 2009), NAS needs 4 days with 450 GPUs to discover the best architecture). ENAS has been proposed to improve the efficiency of NAS. The key idea of ENAS (Pham et al., 2018a) is the use of sharing parameters among child models, which helps reducing the time of training each child model from scratch to convergence. State-of-the-art performance has been achieved by ENAS on well known public datasets. We encourage the readers to refer to the original paper (Pham et al., 2018a) for more details. FIGURE 5.3 illustrates the entire pipeline of our approach for the recognition stage.

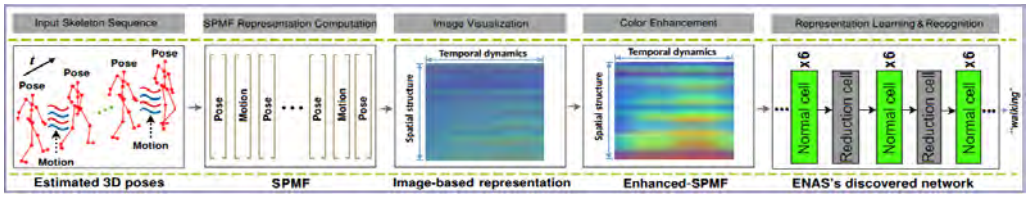


FIGURE 5.3: Illustration of the proposed approach for 3D human action recognition. Intead of using skeletal data provided by depth sensors as described in chapter 4, we exploit in this chapter the estimated 3D poses from RGB sensors.

5.4 Experiments

5.4.1 Datasets and settings

We evaluate the proposed method on three challenging datasets: Human3.6M, MSR Action3D and SBU Kinect Interaction. The Human3.6M is used for evaluating 3D pose estimation. Meanwhile, the other two datasets are used for validating action recognition. The characteristics of each dataset are as follows.

Human3.6M (Ionescu et al., 2014): This is a very large-scale dataset containing 3.6 million different 3D articulated poses captured from 11 actors for 17 actions, under 4 different viewpoints. For each subject, the dataset provides 32 body joints, from which only 17 joints are used for training and computing scores. In particular, 2D joint locations and 3D poses ground truth are available for evaluating supervised learning models.

MSR Action3D (Li, Zhang, and Liu, 2010): This dataset contains 20 actions, performed by 10 subjects. Our experiment was conducted on 557 video sequences of the MSR Action3D, in which the whole dataset is divided into three subsets: AS1, AS2, and AS3. There are 8 actions classes for each subset. Half of the data is selected for training and the rest is used for testing. Section 4.1 provides more details about the MSR Action3D.

SBU Kinect Interaction (Yun et al., 2012a): This dataset contains a total of 300 interactions, performed by 7 participants for 8 actions. This is a challenging dataset due to the fact that it contains pairs of actions that are difficult to distinguish such as *exchanging objects – shaking*

hands or *pushing – punching*. We randomly split the whole dataset into 5 folds, in which 4 folds are used for training and the remaining 1 fold is used for testing. More details about this dataset can be found in section 4.3.

5.4.2 Implementation details

The proposed networks were implemented in Python with Keras/TensorFlow backend. The two streams of the 3D pose estimator are trained separately with the same hyperparameters setting, in which we use mini-batches of 128 poses with 0.25 dropout rate. The weights are initialized by the He initialization (He et al., 2015). Adam optimizer (Kingma and Ba, 2014) is used with default parameters. The initial learning rate is set to 0.001 and is decreased by a factor of 0.5 after every 50 epochs. The network is trained for 300 epochs from scratch on the Human3.6M dataset (Ionescu et al., 2014). For action recognition task, we run OpenPose (Cao et al., 2017) to generate 2D detections on MSR Action3D (Li, Zhang, and Liu, 2010) and SBU Kinect Interaction (Yun et al., 2012a). The pre-trained 3D pose estimator on Human3.6M (Ionescu et al., 2014) is then used to provide 3D poses in which the input data are the 2D poses provided by the OpenPose. We use standard data pre-processing and augmentation techniques such as randomly cropping and flipping on these two datasets due to their small sizes. To discover optimal recognition networks, we use ENAS (Pham et al., 2018a) with the same parameter setting as the original work. Concretely, the shared parameters ω are trained with Nesterov accelerated gradient descent (Yurii, 1983) using Cosine learning rate (Ilya and Frank, 2016). The candidate architectures are initialized by He initialization (He et al., 2015) and trained by Adam optimizer (Kingma and Ba, 2014) with a learning rate of 0.00035. Additionally, each search is run for 200 epochs.

5.4.3 Experimental results and comparison

Evaluation on 3D pose estimation

We evaluate the effectiveness of the proposed 3D pose estimation network using the standard protocol of the Human3.6M dataset (Ionescu et al., 2014; Pavlakos et al., 2017; Martinez et al., 2017; Mehta et al., 2017b). Five subjects S1, S5, S6, S7, S8 are used for training and the rest two subjects S9, S11 are used for evaluation. Experimental results are reported by the average error in millimeters between the ground truth and the corresponding predictions over all joints. The results show that our method outperforms the previous best result from the literature (Martinez et al., 2017) by 3.1mm, corresponding to an error reduction of 6.8% even when combining the ground truth 2D locations with the 2D OpenPose detections. This result proves that our network design can learn to recover the 3D pose from the 2D joint locations with a low error rate, which to the best of our knowledge, has established a new state-of-the-art on 3D human pose estimation (see TABLE 5.1).

FIGURE 5.4: Visualization of 3D output of the estimation stage with some samples on the test set of Human3.6M (Ionescu et al., 2014). For each example, from left to right are 2D poses, 3D ground truths and our 3D predictions, respectively.

Evaluation on 3D action recognition

In this section, we present our experimental results on the task of action recognition. We compare the obtained results with several state-of-the-art approaches. At this step, the input of deep learning networks are the estimated 3D poses from RGB images, provided by

TABLE 5.1: Experimental results and comparison with previous state-of-the-art 3D pose estimation approaches on the Human3.6M dataset (Ionescu et al., 2014). Results are reported by the average error in millimeters between the ground truth and the corresponding predictions over all joints.

Method	Direct.	Disc.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg
Ionescu et al., 2014†	132.7	183.6	132.3	164.4	162.1	205.9	150.6	171.3	151.6	243.0	162.1	170.7	177.1	96.6	127.9	162.1
Du et al., 2016⋆	85.1	112.7	104.9	122.1	139.1	135.9	105.9	166.2	117.5	226.9	120.0	117.7	137.4	99.3	106.5	126.5
Tekin et al., 2016	102.4	147.2	88.8	125.3	118.0	182.7	112.4	129.2	138.9	224.9	118.4	138.8	126.3	55.1	65.8	125.0
Park, Hwang, and Kwak, 2016⋆	100.3	116.2	90.0	116.5	115.3	149.5	117.6	106.9	137.2	190.8	105.8	125.1	131.9	62.6	96.2	117.3
Zhou et al., 2016⋆	87.4	109.3	87.1	103.2	116.2	143.3	106.9	99.8	124.5	199.2	107.4	118.1	114.2	79.4	97.7	113.0
Xingyi et al., 2016⋆	91.8	102.4	96.7	98.8	113.4	125.2	90.0	93.8	132.2	159.0	107.0	94.4	126.0	79.0	99.0	107.3
Pavlakos et al., 2017	67.4	71.9	66.7	69.1	72.0	77.0	65.0	68.3	83.7	96.5	71.7	65.8	74.9	59.1	63.2	71.9
Mehta et al., 2017a⋆	67.4	71.9	66.7	69.1	71.9	65.0	68.3	83.7	120.0	66.0	79.8	63.9	48.9	76.8	53.7	68.6
Martinez et al., 2017⋆	51.8	56.2	58.1	59.0	69.5	55.2	58.1	74.0	94.6	62.3	78.4	59.1	49.5	65.1	52.4	62.9
Shuang, Xiao, and Yichen, 2018	52.8	54.2	54.3	61.8	53.1	53.6	71.7	86.7	61.5	53.4	67.2	54.8	53.4	47.1	61.6	59.1
Luvizon, Picard, and Tabia, 2018	49.2	51.6	47.6	50.5	51.8	48.5	51.7	61.5	70.9	53.7	60.3	48.9	44.4	57.9	48.9	53.2
Martinez et al., 2017†	37.7	44.4	40.3	42.1	48.2	54.9	44.4	42.1	54.6	58.0	45.1	46.4	47.6	36.4	40.4	45.5
Ours†,⋆	**36.6**	**43.2**	**38.1**	**40.8**	**44.4**	**51.8**	**43.7**	**38.4**	**50.8**	**52.0**	**42.1**	**42.2**	**44.0**	**32.3**	**35.9**	**42.4**

The symbol ⋆ denotes that a 2D detector was used and the symbol † denotes the ground truth 2D joint locations were used.

the proposed 3D estimator. For 3D action recognition evaluation, we followed the same protocol as described in chapter 4. The main aim here is to compare our proposed method to those of the state of the art. We report experimental results using recognition accuracy rate (%) on two datasets: the MSR Action3D dataset (Li, Zhang, and Liu, 2010) and the SBU Kinect Interaction dataset (Yun et al., 2012b). TABLE 5.2 shows results and comparisons with state-of-the-art methods on the MSR Action3D dataset (Li, Zhang, and Liu, 2010). The ENAS algorithm (Pham et al., 2018a) is able to explore a diversity of network architectures and the best design is identified based on its validation score. Thus, the final architecture achieved a total average accuracy of 97.98% over three subset AS1, AS2 and AS3. This result outperforms many previous studies (Li, Zhang, and Liu, 2010; Chen, Liu, and Kehtarnavaz, 2013; Vemulapalli, Arrate, and Chellappa, 2014; Du, Wang, and Wang, 2015; Liu et al., 2016b; Wang et al., 2016b; Weng, Weng, and Yuan, 2017; Xu et al., 2015a; Lee et al., 2017), and among them, many are depth sensor-based approaches. APPENDIX B3 provides a schematic diagram of the best cells and optimal architecture found by ENAS on the AS1 subset (Li, Zhang, and Liu, 2010). For the SBU Kinect Interaction dataset (Yun et al., 2012b), the best model achieved an accuracy of 96.30%, as shown in TABLE 5.3. We observe that by only using the 3D predicted poses, we are able to outperform previous works reported in Song et al., 2017; Liu et al., 2016b; Weng et al., 2018; Ke et al., 2017; Tas and Koniusz, 2018; Wang and Wang, 2017; Liu et al., 2018. The comparison with the Enhanced-SPMF DenseNet that was described in section 4.3 leads to a slight lower accuracy (97.98% versus 99.10%). This means that the estimated 3D pose provided by our method is comparable to 3D skeletal data provided by Kinect v2 sensor.

5.4.4 Computational efficiency evaluation

On a single GeForce GTX 1080Ti GPU with 11GB memory, the runtime of OpenPose (Cao et al., 2017) is less than 0.1*s* per frame on a image size of 800 $\times$ 450 pixels. On the Human3.6M dataset (Ionescu et al., 2014), the 3D pose estimation stage takes around 15*ms* to complete a pass (forward + backward) through each stream with a mini-batch of size 128. Each epoch was done within 3 minutes. For the action recognition stage, our implementation of ENAS algorithm takes about 2 hours to find the final architecture ($\sim$2.3M parameters) on each subset of MSR Action3D dataset (Li, Zhang, and Liu, 2010), whilst it takes around 3 hours

TABLE 5.2: Test accuracies (%) on the MSR Action3D dataset (Li, Zhang, and Liu, 2010).

Method	AS1	AS2	AS3	Aver.
Li, Zhang, and Liu, 2010	72.90	71.90	71.90	74.70
Chen, Liu, and Kehtarnavaz, 2013	96.20	83.20	92.00	90.47
Vemulapalli, Arrate, and Chellappa, 2014	95.29	83.87	98.22	92.46
Du, Wang, and Wang, 2015	99.33	94.64	95.50	94.49
Liu et al., 2016b	N/A	N/A	N/A	94.80
Wang et al., 2016b	93.60	95.50	95.10	94.80
Weng, Weng, and Yuan, 2017	91.50	95.60	97.30	94.80
Xu et al., 2015a	99.10	92.90	96.40	96.10
Lee et al., 2017	95.24	96.43	100.0	97.22
Enhanced-SPMF DenseNet (L=250, k=24)	**98.83**	**99.06**	**99.40**	**99.10**
Proposed method	**97.87**	**96.81**	**99.27**	**97.98**

TABLE 5.3: Test accuracies (%) on the SBU Kinect Interaction dataset (Yun et al., 2012b).

Method	Accuracy (%)
Song et al., 2017	91.51
Liu et al., 2016b	93.30
Weng et al., 2018	93.30
Ke et al., 2017	93.57
Tas and Koniusz, 2018	94.36
Wang and Wang, 2017	94.80
Liu et al., 2018	94.90
Zhang et al., 2019 (using VA-RNN)	95.70
Zhang et al., 2019 (using VA-CNN)	97.50
Enhanced-SPMF DenseNet (L=250,k=24)	**97.86**
Proposed method	**96.30**

on the SBU Kinect Interaction dataset (Yun et al., 2012b) to discover the best architecture ($\sim$3M parameters).

5.5 Conclusion

In this chapter, we presented a unified deep learning framework for joint 3D human pose estimation and action recognition from RGB video sequences. The proposed method first runs a state-of-the-art 2D pose detector to estimate 2D locations of body joints. A deep neural network is then designed and trained to learn a direct 2D-to-3D mapping and predict human poses in 3D space. Experimental results demonstrated that the 3D human poses can be effectively estimated by a simple network design and training methodology over 2D keypoints. We also introduced a novel action recognition approach based on a compact image-based representation and automated machine learning, in which an advanced neural architecture search algorithm is exploited to discover the best performing architecture for each recognition task. Our experiments on public and challenging action recognition datasets indicated

that the proposed framework is able to reach state-of-the-art performance, whilst requiring low computation time for training. Despite that, our method naturally depends on the quality of the output of the 2D detectors. Hence, a limitation is that it cannot estimate 3D poses in the case the 2D detector failure. For the time being, we do not know if the failure is frequently happening or not. This a perspective of work.

Chapter 6

Conclusions and Perspectives

Contents

Chapter overview: We summarize and discuss in this chapter the key findings of this thesis, tying together the various tasks carried and the obtained results. We then outline the limitations of the proposed approaches. Finally, we end the thesis by providing some research directions for future work.

6.1 Discussion

Our main goal in this thesis was to propose, develop and validate different deep learning-based approaches for determining which human actions occur from monocular RGB-D video sequences. To tackle this problem, we first reviewed the most prominent state-of-the-art deep learning algorithms applied to the recognition of human actions in videos (Chapter 3). We found that Deep Convolutional Neural Networks (D-CNNs) based approaches are among the best performing learning models to address this task. We then proposed a new approach for skeleton-based action recognition using D-CNNs from skeletal data provided by depth cameras (Chapter 4). Two key questions had been studied and addressed. First, how to efficiently represent the spatio-temporal patterns of skeletal data for fully exploiting the capacity in learning high-level representations of D-CNNs. Second, how to design a powerful D-CNN architecture that is able to learn discriminative features from the proposed representation for classification task. As a result, we introduced two new 3D motion representations called SPMF ("*Skeleton Posture-Motion Feature*") and Enhanced-SPMF that encode skeleton poses and their motions into color images. The proposed representations (called action maps) were then fed into state-of-the-art D-CNNs such as ResNet (Kaiming et al., 2016), Inception-ResNet-v2 (Szegedy et al., 2017), DenseNet (Huang et al., 2017) and ENAS (Pham et al., 2018a) for feature learning and recognition. Experimental results on various public and challenging human action recognition datasets including MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014), SBU Kinect Interaction (Yun et al., 2012a), and NTU-RGB+D (Shahroudy et al., 2016) showed the effectiveness of the proposed representations as well as the deep learning frameworks.

Our study also showed that the Enhanced-SPMF was able to capture the spatio-temporal

motion features better than the SPMF. Specifically, we carried out an ablation study by training and evaluating a DenseNet (Huang et al., 2017) on both the SPMF and Enhanced-SPMF provided by the SBU Kinect Interaction dataset (Yun et al., 2012a), in which the same hyper-parameters setting and training methodology were applied (Chapter 4). The experimental results indicated that learning on the Enhanced-SPMF led to a better recognition performance. This proved that the use of the AHE algorithm (Pizer et al., 1987) and the Savitzky-Golay smoothing filter (Savitzky and Golay, 1964) helped improving the accuracy.

For learning and classification tasks, we designed different D-CNN architectures based on the ResNet (Kaiming et al., 2016), Inception-ResNet-v2 (Szegedy et al., 2017), DenseNet (Huang et al., 2017) and Effective Neural Architecture Search (ENAS - Pham et al., 2018a) to extract robust features from color–coded images and classify them into classes. We pose the question how to identify which is the best performing architectures? It is difficult but interesting to try to answer this question. For example, TABLE 6.1 shows that the proposed DenseNet ($L = 250$, $k = 24$) got the best accuracies on the NTU-RGB+D dataset (Shahroudy et al., 2016). However, the DenseNet model performed worse than the proposed ResNet-44 on smaller dataset like the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010). As we have explained in Chapter 4, learning behaviors of deep neural networks are heavily dependent on the size and distribution of input data.

TABLE 6.1: Summary of the proposed models and their experimental results.

Model	Input	MSR Action3D (overall)	KARD (overall)	SBU Kinect (overall)	NTU-RGB+D (cross-subject)	NTU-RGB+D (cross-view)
ResNet-44	Image-coded	99.90%	99.98%	N/A	77.20%	84.80%
Inception-ResNet-222	SPMF	98.56%	N/A	N/A	78.89%	86.15%
DenseNet	Enhanced-SPMF	99.10%	N/A	96.67%	80.11%	86.82%
ENAS	Estimated 3D pose + Enhanced-SPMF	97.98%	N/A	96.30%	N/A	N/A

In this thesis, we also conducted research on the problem of 3D human pose estimation from monocular RGB video sequences and used the estimated 3D poses for action recognition task. As shown is TABLE 6.1, the ENAS algorithm (Pham et al., 2018a) was trained on the estimated 3D poses (with Enhanced-SPMF) and achieved an accuracy of 97.98% on the MSR Action3D dataset (Wanqing, Zhengyou, and Zicheng, 2010) and 96.30% on the SBU Kinect Interaction dataset (Yun et al., 2012b), respectively. Compared to the proposed DenseNet that was trained on skeleton sequences provided by Kinect cameras, the recognition accuracies provided by the ENAS (Pham et al., 2018a) on the estimated 3D poses are lower. However, these differences are quite small. In TABLE 6.1 some tests were not carried out for time reason and also for the size of the datasets. For example, for ENAS model – the NTU-RGB+D dataset was so big that it would have taken around several months to process them. Always for a question of time, we made the choice to use the SBU Kinect and MSR Action3D datasets to evaluate the effectiveness of DenseNet with Enhanced-SPMF and ENAS models.

In addition, we collected and introduced the CEMEST – a new RGB-D dataset depicting passenger behaviors in public transport. The dataset consists of 203 untrimmed real-world surveillance videos of realistic normal and anomalous events. We have made this dataset public[1] in order to encourage the development of the field. We achieved promising results on real conditions of the CEMEST with the proposed DenseNet-28 trained on the Enhanced-SPMF representation, in which some data augmentation and transfer learning techniques were exploited. Now, we are continuing to conduct new experiments with the proposed 3D pose estimation and recognition approach on this dataset. The obtained results will be reported in our next publication.

[1] https://sites.google.com/site/hhpham172/image-processing-and-computer-vision/tisseo-cerema-dataset.

6.2 Limitations

Although the effectiveness of the proposed methods have been proven in terms of accuracy and speed, some limitations are still remaining, which requires more research works to overcome.

Firstly, the proposed models are not suitable to online data (*i.e.* "*Online Action Recognition*" or OAR) that aims to detect and recognize actions from unsegmented streams in a continuous manner, where boundaries between different kinds of actions within the stream are unknown. For the time being, our study focused on recognizing actions from segmented sequences of input data, with each segment corresponding to one single action or interaction. A common approach for an OAR problem that we could consider is the "*sliding window-based methods*" (Kviatkovsky, Rivlin, and Shimshoni, 2014; Kulkarni et al., 2015; Zhu et al., 2016a). These approaches consider the temporal coherence within the window for prediction. This idea can also be applied to solve our problem. Specifically, the recognition system may have two main phases including offline training and online recognition. During the training phase, we train the proposed deep neural networks from segmented sequences in a supervised manner. The spatio-temporal features of actions will be learned and the learned weights will be stored as a pre-trained model. During the online recognition phase, we use a sliding window on the original skeleton sequences or on the original image-coded representation to predict actions by the pre-trained model. However, we understand that the performance of these methods are sensitive to the window size and a compromise has to be found. Ìf the window is too small or too large this could lead to a significant drop in performance.

Secondly, the proposed D-CNNs such as ResNets (Kaiming et al., 2016), Inception-ResNet-v2 (Szegedy et al., 2017), DenseNets (Huang et al., 2017) are very deep networks that contain millions of trainable parameters. Hence, exploiting these architectures on CPUs and mobile platforms is unrealistic.

Thirdly, the proposed method for 3D pose estimation from RGB video sequences naturally depends on the quality of the output of the 2D detectors. Therefore, a limitation is that it cannot accurately estimate 3D poses from a bad 2D pose estimator. This problem could be tackled by adding more visual information such as color silhouettes of people to the network in order to further gains in performance. In that case, the processing time will be increased.

Finally, we have collected and introduced the CEMEST dataset. We recognized that the CEMEST is a small dataset and training supervised learning algorithms such as D-CNNs could easily lead to overfitting.

6.3 Future work

There are many potential research directions that could be considered to expand the current approach. Here we outline some of the most promising ideas.

6.3.1 Recurrent Neural Networks with Long Short-Term Memory units

Recurrent Neural Networks and Long Short-Term Memory (RNN-LSTM) units are widely used for time series modeling and forecasting. This kind of network can completely be used for modeling the spatio-temporal features contained in the proposed SPMF and Enhanced-SPMF representations. An RNN-LSTM can be used to model the temporal dependencies between the 3D positional configurations of human body joints. In particular, some new gating mechanisms or trust gates for LSTM such as the works of (Veeriah, Zhuang, and Qi, 2015; Liu et al., 2016b) allow modeling the derivatives of the memory states and explore the salient action patterns. In fact, the SPMF and Enhanced-SPMF are motion maps that contain both spatial and temporal information of human actions. The elements of the SPMF and Enhanced-SPMF (*e.g.* rows) can be considered as temporal signals and their features can be modeled by an RNN-LSTM.

6.3.2 Temporal Convolutional Network

The work of Bai, Kolter, and Koltun, 2018 has proven that temporal convolutional architectures (see FIGURE 6.1) can outperform recurrent networks on tasks such as audio synthesis and machine translation. Given a new sequence modeling task or dataset, the authors of this research indicated that a simple convolutional architecture could outperform canonical recurrent networks such as RNN-LSTMs across a diverse range of tasks and datasets, while demonstrating longer effective memory.

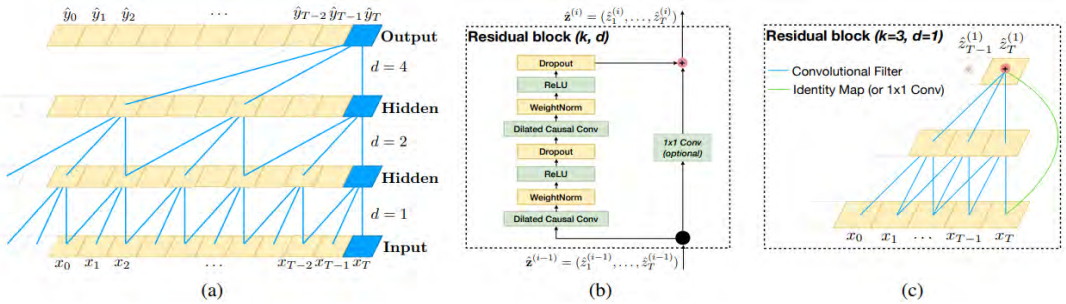


FIGURE 6.1: Illustration of a Temporal Convolutional Network (Bai, Kolter, and Koltun, 2018): **(a)** A dilated causal convolution with dilation factors d = 1, 2, 4 and filter size k = 3. The receptive field is able to cover all values from the input sequence. **(b)** TCN residual block. An 1x1 convolution is added when residual input and output have different dimensions. **(c)** An example of residual connection in a TCN. The blue lines are filters in the residual function, and the green lines are identity mappings.

6.3.3 Multi-Stream Deep Neural Networks

The proposed 3D motion representations (SPMF and Enhanced-SPMF) are constructed from two action features: static postures and temporal motions. The two features were combined into a unified color image representation and fed into D-CNNs for representation learning. This combination could make the representation learning process more complicated[2]. An alternative solution is to encode each type of feature into an image and build a two-stream deep neural network framework that accepts each channel as an input. The final layer of each stream will be fused later to improve the performance.

6.3.4 Attention Temporal Networks

The Attention Temporal Networks (ATNs – Zang et al., 2018; Li et al., 2019) are also a promising research direction to further boost the performance of human action recognition in videos. Instead of processing all sampled video frames equally, an ATN network (Zang et al., 2018) has an attention mechanism that allows to automatically focus more heavily on the semantically critical segments and could lead to reduce less important video frames as well as noise. This idea can be also applied for skeletal data provided by RGB+D sensors as explained in the previous chapters or 3D poses estimated from 2D poses coming from RGB sensor (Xie et al., 2018; Si et al., 2019). For instance, Si et al., 2019 proposed a deep network architecture called AGC-LSTM within an attention mechanism as shown in FIGURE 6.2. This architecture is able to capture discriminative features in spatio-temporal dynamics and explore the co-occurrence relationship between spatial and temporal domains. By this way, the AGC-LSTM has the ability to learn the high-level semantic representation by selecting discriminative spatial information from skeleton joints.

[2]Personal communication with Pablo Zegers from the University of the Andes, Santiago, Chile. Pablo recommended to me that I should split the SPMF and Enhanced-SPMF maps into two independent channels and building a two stream-CNN for feature learning and recognition tasks.

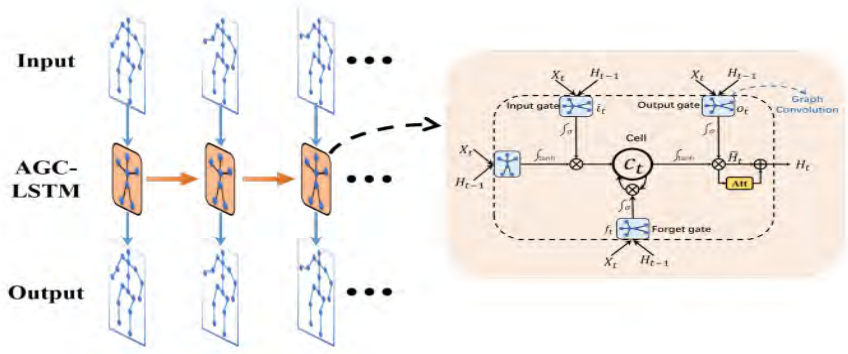


FIGURE 6.2: Structure of the AGC-LSTM layer (Si et al., 2019): AGC-LSTM layers are used to model spatial-temporal features of skeletal movements. The graph convolutional operator within AGC-LSTM can not only effectively capture discriminative features in spatio-temporal dynamics but also explore the co-occurrence relationship between spatial and temporal domains, which provides richer motion features for recognition task.

Appendix A

Datasets

A1. List of action classes from the NTU-RGB+D dataset

Below is the list of the action classes provided by the NTU-RGB+D dataset (Shahroudy et al., 2016). It contains 60 different actions captured by Kinect v2 sensors:

Drinking, eating, brushing teeth, brushing hair, dropping, picking up, throwing, sitting down, standing up, clapping, reading, writing, tearing up paper, wearing jacket, taking off jacket, wearing a shoe, taking off a shoe, wearing on glasses, taking off glasses, puting on a hat/cap, taking off a hat/cap, cheering up, hand waving, kicking something, reaching into self pocket, hopping, jumping up, making/answering a phone call, playing with phone, typing, pointing to something, taking selfie, checking time, rubbing two hands together, bowing, shaking head, wiping face, saluting, putting palms together, crossing hands in front, sneezing/coughing, staggering, falling down, touching head, touching chest, touching back, touching neck, vomiting, fanning self, punching/slapping other person, kicking other person, pushing other person, patting other's back, pointing to the other person, hugging, giving something to other person, touching other person's pocket, handshaking, walking towards each other, and walking apart from each other.

A2. List of action classes of the combination dataset

To improve the learning performance of the proposed deep networks on the CEMEST dataset, we prepared a dataset as the combination of the public action datasets and exploited transfer learning. Specifically, the following action classes from the MSR Action 3D (Wanqing, Zhengyou, and Zicheng, 2010) and NTU-RGB+D (Shahroudy et al., 2016) datasets were used for training the proposed DenseNets before fine-tuning on the CEMEST:

Walking, bend, jogging, jumping up, forward punch, high arm wave, hand clap, dropping, picking up, sitting down, standing up, hand waving, pointing to something, staggering, falling down, punching/slapping other person, kicking other person, pushing other person, patting others back, giving something to other person, touching other persons pocket.

To ensure the number of samples in each action class is balanced, we augmented samples in the MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010) to match the size of the larger dataset. The network configurations were kept the same as the previous experiments. More specifically, three configurations including DenseNet-16, DenseNet-28, and DenseNet-40 were trained with a learning rate of $3e-4$, a batch size of 64 and training for 250 epochs.

Appendix B

Network Architectures

B1. ResNets

This section describes the network architectures in detail. To build 20-layer, 32-layer, 44-layer, 56-layer, and 110-layer networks, we stack the proposed ResNet building units as following:

Baseline 20-layer ResNet architecture
3x3 Conv., 16 filters, BN, ReLU
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Global mean pooling
FC layer with n units where n is equal to the number of action class
Softmax layer

Baseline 32-layer ResNet architecture
3x3 Conv., 16 filters, BN, ReLU
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual block: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Global mean pooling
FC layer with n units where n is equal to the number of action class
Softmax layer

Baseline 44-layer ResNet architecture
3x3 Conv., 16 filters, BN, ReLU
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Global mean pooling
FC layer with n units where n is equal to the number of action class
Softmax layer

Baseline 56-layer ResNet architecture
3x3 Conv., 16 filters, BN, ReLU
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Global mean pooling
FC layer with n units where n is equal to the number of action class
Softmax layer

Baseline 110-layer ResNet architecture

3x3 Conv., 16 filters, BN, ReLU
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,16 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,32 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Residual unit: BN-ReLU-Conv.-BN-ReLU-Dropout-Conv.,64 filters
Global mean pooling
FC layer with n units where n is equal to the number of action class
Softmax layer

B2. Inception-ResNet-v2

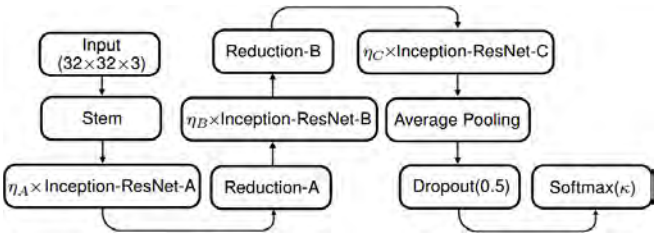


FIGURE B.1: The proposed Inception-ResNet architectures. The symbols η_A, η_B and η_C denote the number of building blocks Inception-ResNet-A, Inception-ResNet-B, and Inception-ResNet-C, respectively. The symbol κ indicates the number of neurons in Softmax layer, corresponding to the number of action classes. The details about blocks are provided in FIGURE B.2, FIGURE B.3, FIGURE B.4, FIGURE B.5, and FIGURE B.6.

TABLE B.1 below describes in details the network configurations SPMF Inception-ResNet-121, SPMF Inception-ResNet-222, and SPMF Inception-ResNet-242. To measure the complexity of networks, we report the number of network parameters.

TABLE B.1: The proposed network configurations and their complexities (M = $\times$ 10^6).

Network architecture	η_A	η_B	η_C	# param.
SPMF Inception-ResNet-121	1	2	1	1.35M
SPMF Inception-ResNet-222	2	2	2	1.59M
SPMF Inception-ResNet-242	2	4	2	1.86M

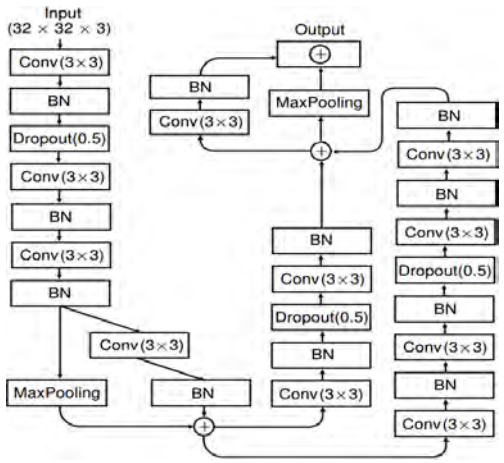


FIGURE B.2: The schema for STEM block. The symbol $\oplus$ denotes the concatenation operator.

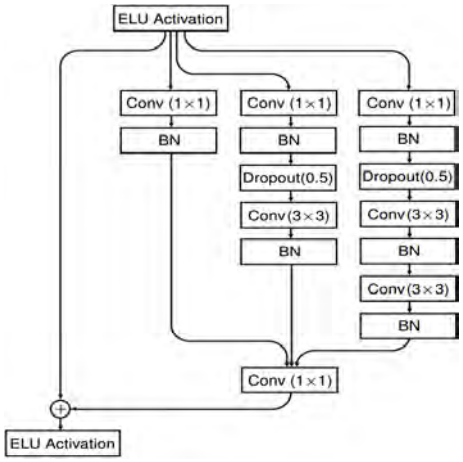


FIGURE B.3: The schema for Inception-ResNet-A block. The symbol ⊕ denotes the concatenation operator.

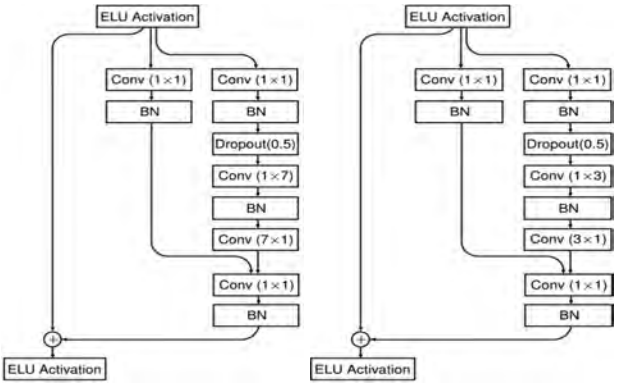


FIGURE B.4: The schemas for Inception-ResNet-B (**left**) and Inception-ResNet-C (**right**) blocks. The symbol ⊕ denotes the concatenation operator.

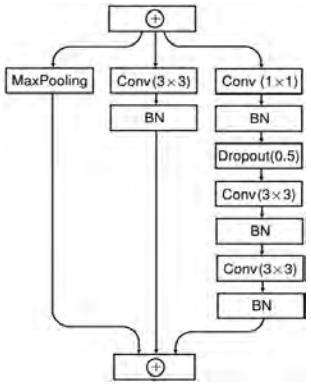


FIGURE B.5: The schema for Reduction-A block. The symbol ⊕ denotes the concatenation operator.

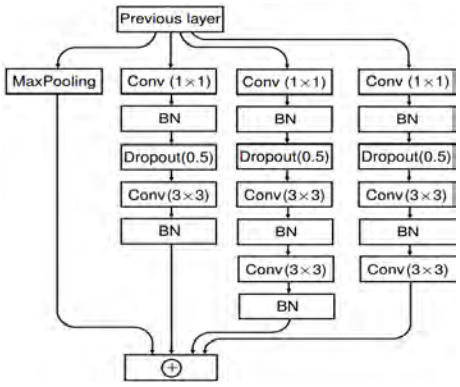
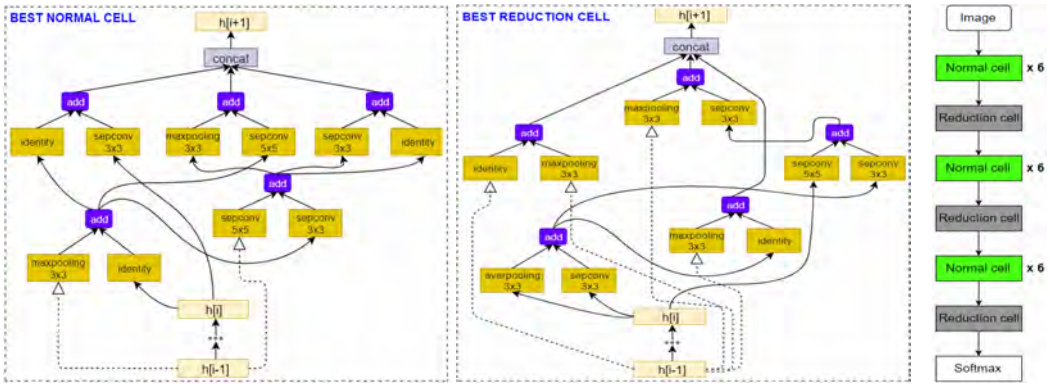


FIGURE B.6: The schema for Reduction-B block. The symbol ⊕ denotes the concatenation operator.

B3. Deep learning architecture discovered by ENAS algorithm



(a) (b) (c)

FIGURE B.7: Diagram of the top performing *normal cell* (a) and *reduction cell* (b) discovered by ENAS (Pham et al., 2018a) on AS1 subset (Li, Zhang, and Liu, 2010). They were then used to construct the final network architecture (c). We recommend the interested readers to Pham et al., 2018a to better understand this procedure.

Appendix C

Savitzky-Golay Smoothing Filter

Savitzky-Golay (S-G) filter is a *low-pass* filter based on local least-squares polynomial approximation that is often used to smooth noisy data. The 3D skeleton joints obtained from depth cameras can be considered as a series of equally spaced data in the time domain, applying S-G filter on raw skeletal data helps reduce the level of noise while maintaining the 3D geometric characteristics of the input sequences.
Considering a sequence of $N = 2M + 1$ input data points $x[n]$ centered at $n = 0$, given by

$$\mathbf{x} = [x_{-M}, ..., x_{-1}, x_0, x_1, ..., x_M]^T. \tag{C.1}$$

The N data samples of $\mathbf{x}$ can be fitted by a polynomial

$$p(n) = \sum_{k=0}^{N} c_k n^k. \tag{C.2}$$

To best fit the given data $\mathbf{x}$, Savitzky and Golay, 1964 proposed a method of data smoothing by finding the vector of polynomial coefficients $\mathbf{c} = [c_0, c_1, ..., c_N]^T$ that minimizes the mean-squares approximation error

$$\mathcal{E}_N = \sum_{n=-M}^{M} \left(\sum_{k=0}^{N} c_k n^k - x[n] \right)^2. \tag{C.3}$$

To this end, one solution is to determine a set of coefficients that satisfies the partial derivative equation is equal to zero

$$\frac{\partial \mathcal{E}_N}{\partial a_i} = \sum_{n=-M}^{M} 2n^i \left(\sum_{k=0}^{N} c_k n^k - x[n] \right) = 0 \text{ with } i = 0, 1, ..., N. \tag{C.4}$$

Eq. (C.4) is equivalent to

$$\sum_{k=0}^{N} \left(\sum_{n=-M}^{M} n^{i+k} \right) c_k = \sum_{n=-M}^{M} n^i x[n]. \tag{C.5}$$

Defining a matrix $\mathbf{A} = \{\alpha_{n,i}\}$ as the matrix with elements

$$\alpha_{n,i} = n^i \tag{C.6}$$

where $-M \leq n \leq M$ and $i = 0, 1, ..., N$. The matrix $\mathbf{A}$ is called the design matrix for the polynomial approximation problem. Note that, the transpose of $\mathbf{A}$ is $\mathbf{A}^T = \{\alpha_{i,n}\}$ and the product matrix $\mathbf{B} = \mathbf{A}^T\mathbf{A}$ is a symmetric matrix with elements

$$\beta_{i,k} = \sum_{n=-M}^{M} \alpha_{i,n}\alpha_{n,k} = \sum_{n=-M}^{M} n^{i+k} \tag{C.7}$$

Therefore, Eq. (C.5) can be rewritten in matrix form as

$$\mathbf{Bc} = \mathbf{A}^T\mathbf{Ac} = \mathbf{A}^T\mathbf{x}. \tag{C.8}$$

The polynomial coefficients can be determined as

$$\mathbf{c} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{x}\,). \tag{C.9}$$

For example, for smoothing by a 5-point quadratic polynomial with $N = 5, M = -2, -1, 0, 1, 2$, the tth filtering result, y_t is given by

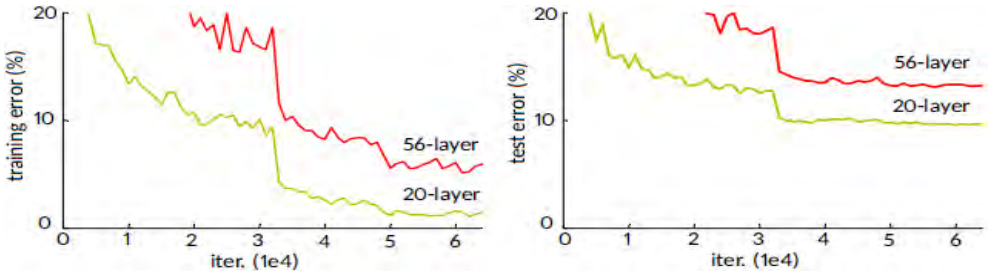
$$y_t = \frac{-3x_{t-2} + 12x_{t-1} + 17x_t + 12x_{t+1} - 3x_{t+2}}{35}. \tag{C.10}$$

Eq. (C.10) above was used in our experiments to reduce the effect of noise on the raw skeleton data.

Appendix D

Degradation phenomenon in training very deep neural networks

Very deep neural networks demonstrate to have a high performance on many visual-related tasks (Simonyan and Zisserman, 2014b; Szegedy et al., 2015b; Kaiming et al., 2016; Telgarsky, 2016). However, they are very difficult to optimize. One of the main challenges for training deeper networks is the vanishing and exploding gradient problems (Glorot and Bengio, 2010). Specifically, when the network is deep enough, the supervision signals from the output layer can be completely attenuated or exploded on their way back towards the previous layers. Therefore, the network cannot learn the parameters effectively. These obstacles can be solved by recent advanced techniques in deep learning such as Normalized Initialization (LeCun et al., 1998a) or Batch Normalization (Ioffe and Szegedy, 2015). When the deep networks start converging, a degradation phenomenon occurs. Due to this, the training and test errors increase if more layers are added to a deep architecture. This phenomenon is called by the degradation phenomenon. FIGURE D.1 shows an experimental result (Kaiming et al., 2016) related to this phenomenon.



(a) (b)

FIGURE D.1: Degradation phenomenon during training D-CNNs. (**a**) Training error and (**b**) test error on CIFAR-10 (Krizhevsky, 2009) with 20-layer and 56-layer CNNs reported by Kaiming et al., 2016. The deeper network has higher error for both training and test phases.

Appendix E

Version française résumée

Chapitre 1

Introduction

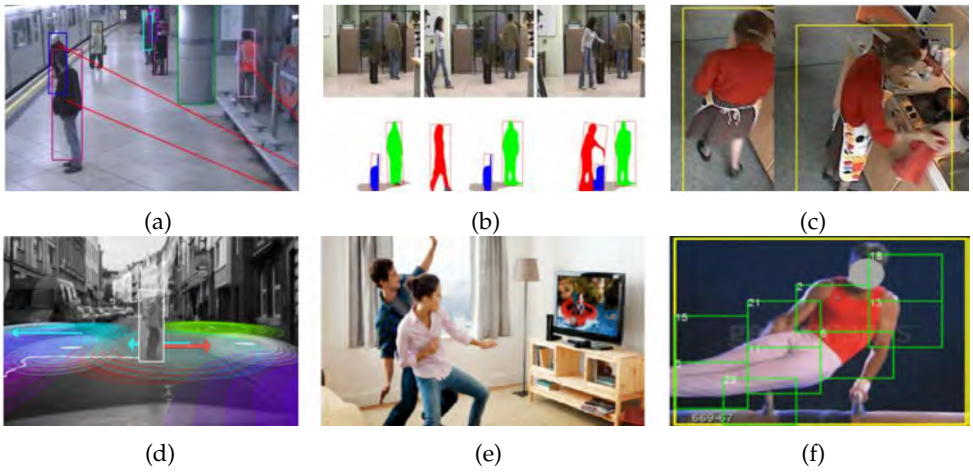
La reconnaissance des actions humaines joue un rôle important dans plusieurs systèmes intelligents d'analyse vidéo. Son objectif principal consiste à analyser automatiquement les flux vidéo fournis par les capteurs optiques afin de reconnaître les actions qui se produisent dans la scène observée. Ce sujet de recherche a été initialement motivé par le domaine des représentations artistiques, la biomécanique et la perception du mouvement (Ivan, 2012). Ensuite, il a été élargi à de nombreuses applications (Ranasinghe, Al Machot, and Mayr, 2016) comme les systèmes de surveillance intelligents (Wei Niu et al., 2004; Valera and Velastin, 2005; Weiyao Lin et al., 2008), l'interation homme-machine (Pickering, Burnham, and Richardson, 2007; Sonwalkar et al., 2015), la santé (Zouba et al., 2009), ou la réalité virtuelle (Maqueda et al., 2015). La FIGURE E.1 montre des exemples d'applications spécifiques dans lesquelles la reconnaissance des actions joue un rôle clé.

Bien que des progrès importants aient été réalisés au cours de ces deux dernières décennies, le développement d'un système de reconnaissance des actions rapide et précis est une tâche particulièrement difficile à cause d'un certain nombre de contraintes liées à l'acquisition des vidéos comme les conditions d'éclairage, la position, l'orientation et le champ de vue de la caméra (Poppe, 2010), ainsi que par les contraintes liées à la variabilité de la réalisation des actions, notamment leur vitesse d'exécution.

Les approches traditionnelles de vision par ordinateur considèrent un système de reconnaissance des actions comme un processus hiérarchique, dans lequel les niveaux inférieurs correspondent à la détection et à la segmentation des personnes et les niveaux supérieurs permettent l'extraction de caractéritiques qui vont être utilisées pour reconnaître les actions (voir la FIGURE E.2). Bien que les méthodes traditionnelles montrent leur efficacité dans de nombreux cas, elles restent limitées car elles sont fortement dépendantes des données et nécessitent beaucoup de descripteurs spatio-temporels complexes. Par conséquent, l'un des principaux défis de la reconnaissance des actions humaines dans les vidéos est de trouver une représentation robuste et assez discriminante pour que les modèles d'apprentissage soient capables de reconnaître de manière fiable plusieurs actions différentes.

Outre les difficultés mentionnées précédemment, les chercheurs et les ingénieurs dans ce domaine sont également confrontés à de nouveaux défis. Par exemple, la complexité dans les bases de données à grande échelle pose un nouveau problème: la reconnaissance des « *actions et comportements complexes dans les vidéos* ». De plus, comment construire des « *systèmes de reconnaissance des actions humaines en temps réel* » est également un problème important, en particulier dans le cas où ces systèmes sont construits sur des modèles chronophages tels que l'apprentissage de réseaux de neurones profonds.

Dans cette thèse, nous abordons le problème de la reconnaissance des actions humaines dans des vidéos RGB-D monoculaires. Nous exploitons l'apprentissage automatique afin



(a) (b) (c)

(d) (e) (f)

FIGURE E.1: Quelques applications importantes sur la reconnaissance d'actions humaines par vidéo : **(a)** reconnaissance et suivi d'actions humaines dans les systèmes de transport intelligents (Ryoo and Aggarwal, 2008); **(b)** détection de vol (Ryoo and Aggarwal, 2007); **(c)** service de surveillance à distance de personnes âgées avec détection de chute (Zouba et al., 2009); **(d)** détection de piétons devant des véhicules autonomes (Kooij, Schneider, and Gavrila, 2014); **(e)** reconnaissance d'actions avec des capteurs de profondeur pour l'industrie du jeu (Zhang, 2012); **(f)** localisation et analyse d'actions sportives (Tian, Rahul, and Shah, 2013).

de reconnaître les actions humaines à partir de séquences de squelettes et en utilisant des réseaux de neurones convolutifs (CNNs). L'objectif dans le cadre de ce travail est de proposer de nouvelles représentations du mouvement à partir des données RGB-D et des modèles d'apprentissage profonds performants qui permettent la reconnaissance des actions humaines précisément et rapidement.

Les objectifs suivants sont principalement visés par cette thèse:

• Identifier l'état actuel de la recherche, les défis, les avantages et les inconvénients des approches basées sur l'apprentissage profond pour la reconnaissance des actions humaines dans les vidéos, en décrivant les architectures profondes les plus couramment utilisées pour l'apprentissage des caractéristiques du mouvement humain.

• Étudier et proposer de nouvelles représentations du mouvement 3D et des architectures d'apprentissage profond pour la reconnaissance des actions humaines à partir d'un capteur RGB-D. L'approche proposée devrait être capable de reconnaître des actions humaines à partir de vidéos réalistes en combinant précision et rapidité.

• Collecter et introduire une nouvelle base de données RGB-D dans les transports publics afin d'évaluer les approches proposées dans des conditions réelles.

Les principales contributions de ces travaux de thèse peuvent être résumées comme suit. Tout d'abord, nous introduisons le problème de la reconnaissance des actions humaines dans des vidéos au travers d'une étude approfondie de plusieurs méthodes de reconnaissance des actions humaines basées sur l'apprentissage profond. À l'aide d'environ 250 publications, nous identifions l'état actuel et les futurs défis concernant ce sujet (chapitre 3). Ensuite, nous présentons de nouvelles représentations du mouvement 3D pour la reconnaissance des actions à partir de séquences de squelettes fournies par des capteurs de profondeur permettant d'utiliser des réseaux de neurones convolutifs (CNNs). Les représentations proposées, que nous appelons SPMF et Enhanced-SPMF, sont capables de capturer la dynamique spatio-temporelle des mouvements du squelette en les transformant en une structure 2D sous la

FIGURE E.2: Un système typique de reconnaissance des actions humaines basées sur la vision par ordinateur. Les régions d'intérêts (ROIs) correspondant aux mouvements humains sont identifiées. Leurs caractéristiques ou descripteurs spatio-temporels, par exemple SIFT (Lowe, 2004), HOG/HOF (Dalal and Triggs, 2005; Laptev et al., 2008) ou HOG-3D (Klaser, Marszałek, and Schmid, 2008) sont ensuite calculés et donnés à un classificateur chargé de reconnaître les actions.

forme d'une image RGB. Cette représentation est bien adaptée à l'apprentissage des réseaux CNNs. La méthode proposée apprend directement la relation entre les séquences du squelette et l'action exécutée via SPMF ou Enhanced-SPMF et montre une amélioration significative des performances par rapport aux approches existantes à l'aide de quatre bases de données de référence très difficiles. L'évaluation de l'efficacité des phases d'apprentissage et d'inférence montre en particulier que la méthode proposée permet d'atteindre un niveau de performance élevé tout en exigeant un temps de calcul réduit. Par ailleurs, nous présentons également CEMEST, une nouvelle base de données RGB-D décrivant le comportement de passagers dans les transports en commun. Elle contient 203 vidéos de vidéosurveillance présentant des événements réalistes normaux et anormaux dans une station de métro à Toulouse en France. Nous obtenons des résultats prometteurs dans les conditions réelles de ces bases de données grâce aux techniques d'augmentation des données et de transfert d'apprentissage. Cela permet d'envisager le développement d'applications réelles basées sur l'apprentissage profond pour améliorer la surveillance et la sécurité des transports publics (chapitre 4).

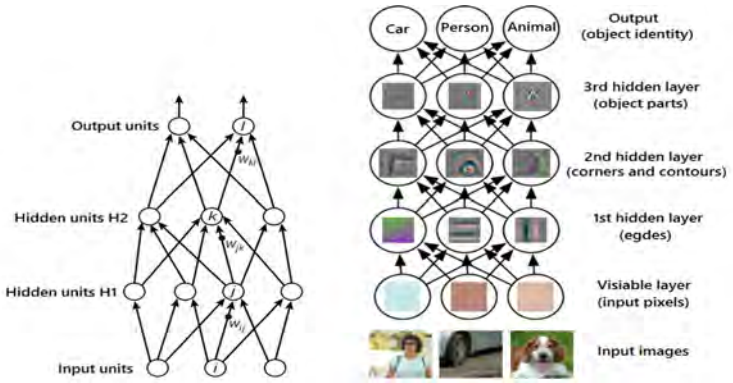
Enfin, nous proposons une méthode d'apprentissage profond unifiée pour l'estimation de poses humaines 3D et la reconnaissance des actions à partir de ces poses. Ce système utilise un détecteur de squelette 2D appelé OpenPose (Cao et al., 2017) pour produire des poses humaines en 2D à partir d'images RGB. Ensuite, il intègre un réseau de neurones profond afin d'apprendre la relation 2D vers 3D entre des poses en 2D et des poses en 3D. Les squelettes 3D obtenus sont ensuite exploités pour la tâche de reconnaissance des actions (chapitre 5). Nous montrons que la méthode d'apprentissage profond proposée est capable de résoudre ces deux tâches (l'estimation des poses 3D et la reconnaissance des actions) de manière efficace.

La thèse est structurée de la manière suivante. Le chapitre 2 est une introduction générale à l'apprentissage profond. Nous présentons des connaissances de base sur les algorithmes d'apprentissage automatique et d'apprentissage profond, ainsi que les modèles d'apprentissage profond inportants. Le chapitre 3 propose une étude complète sur les techniques d'apprentissage profond appliquées à la reconnaissance des actions humaines à partir de vidéos RGB-D. Une description détaillée des approches proposées pour la reconnaissance des actions utilisant des séquences de squelettes fournies par les capteurs de profondeur est faite au chapitre 4. Le chapitre 5 décrit notre nouvelle approche basée sur d'apprentissage profond pour la reconstruction de squelettes en 3D et la reconnaissance des actions à partir de caméras RGB. Enfin, dans le chapitre 6, nous résumons et analysons les principaux résultats du travail réalisé. Nous soulignons ensuite les limites de nos approches et achevons cette thèse en fournissant des orientations prometteuses pour les travaux futurs.

Chapitre 2

Introduction à l'apprentissage profond

L'apprentissage profond est une classe de techniques d'apprentissage automatique. Cette technique est devenue une avancée majeure dans le domaine de la vision par ordinateur et de l'intelligence artificielle après qu'AlexNet (Krizhevsky, Sutskever, and Hinton, 2012a) a réalisé une performance record sur la base de données ImageNet (Rahmani and Mian, 2016). De manière générale, les méthodes d'apprentissage profond sont des méthodes d'apprentissage automatique utilisées pour modéliser des abstractions de haut niveau sur les données à l'aide de réseaux de neurones artificiels, composés de multiples transformations non linéaires. La FIGURE E.3 illustre un réseau multicouche et le processus d'apprentissage des représentations de haut niveau avec des images comme données d'entrée. Plusieurs architectures



(a) (b)

FIGURE E.3: **(a)** Illustration d'un réseau de réseau multicouche (LeCun, Bengio, and Hinton, 2015) et de son processus d'entraînement. Ce modèle permet de déterminer automatiquement des représentations nécessaires aux tâches de prédiction. La première couche appelée « couche visible » qui contient des données sous leur forme brute. À partir de la couche visible, une série de couches cachées est construite en extrayant des entités de plus en plus abstraites à partir des niveaux inférieurs. La couche supérieure contient des informations utiles pour prédire le contenu des données d'entrée. **(b)** Un exemple de modèle d'apprentissage profond pour une tâche de la classification (Zeiler and Fergus, 2014; Goodfellow, Bengio, and Courville, 2016). Étant données certaines images, la première couche est constituée d'un tableau de valeurs des pixels. La première couche cachée représente la présence de bords. Ensuite, la deuxième couche cachée identifie les coins et les contours à partir des bords fournis par la première couche. En reliant les coins et les contours, la troisième couche représente des parries d'objets.

d'apprentissage profond ont été proposées au fil des ans (voir TABLE E.1). Il a été montré que certains d'entre eux atteignent des performances de pointe dans de nombreuses tâches de reconnaissance visuelle.

Dans ce chapitre, nous décrivons les architectures d'apprentissage profond les plus importantes pour la reconnaissance des actions humaines, y compris les réseaux de neurones convolutifs (CNNs) (Fukushima, 1980; Rumelhart, Hinton, and Williams, 1986; LeCun et al., 1989a; Krizhevsky, Sutskever, and Hinton, 2012a), les réseaux de neurones récurrents à mémoire court-terme persistante (RNN-LSTMs) Hochreiter and Schmidhuber, 1997), les réseaux de croyances profonds (DBNs) (Hinton, Osindero, and Teh, 2006), les auto-encodeurs débruiteurs empilés (SDA) (Vincent et al., 2008), et les réseaux antagonistes génératifs (GANs)

). Pour chaque classe d'algorithme, ce chapitre présente leurs concepts de base au travers de leur idée clé et de leur modèle mathématique.

TABLE E.1: Quelques architectures d'apprentissage profond populaires pour les tâches de reconnaissance visuelle.

Architecture	**Article original**
CNNs	Fukushima, 1980;
	Rumelhart, Hinton, and Williams, 1986;
	LeCun et al., 1989a;
	Krizhevsky, Sutskever, and Hinton, 2012a;
	Szegedy et al., 2015a;
	Simonyan and Zisserman, 2014b;
	Kaiming et al., 2016.
RNN-LSTMs	Hochreiter and Schmidhuber, 1997.
DBNs	Hinton, Osindero, and Teh, 2006.
DBMs	Salakhutdinov and Hinton, 2009.
Sparse Coding	Olshausen and Field, 1996;
	Lee et al., 2006.
SDAs	Vincent et al., 2008.
GANs	Goodfellow et al., 2014.

Chapitre 3

L'apprentissage profond pour la reconnaissance des actions humaines : état de l'art

La reconnaissance des actions humaines à partir de vidéos RGB-D est devenue un sujet très populaire dans le domaine de la vision par ordinateur. La capacité à détecter et à prédire correctement des actions dans des vidéos inconnues permet de construire de nombreuses applications importantes dans des domaines tels que la surveillance intelligente, l'interaction homme-machine et la robotique. Au cours des dernières années, les approches basées sur l'apprentissage profond ont montré des performances impressionnantes et un grand potentiel dans l'analyse et la reconnaissance des actions humaines dans des vidéos. De nombreuses architectures profondes différentes ont été proposées pour la reconnaissance des actions et ont permis de faire progresser l'état de l'art dans ce domaine. Ce chapitre décrit l'état de l'art de la reconnaissance des actions humaines à partir de séquences vidéo RGB-D en utilisant l'apprentissage profond. Plus précisément, nous décrivons les architectures profondes les plus couramment utilisées pour apprendre des caractéristiques du mouvement humain et nous montrons comment elles pourraient être appliquées pour relever les défis de la reconnaissance des actions et identifier leurs avantages et leurs limites. En particulier, grâce à des analyses quantitatives des résultats obtenus sur trois grandes bases de données de référence, HMDB-51 (Kuehne et al., 2011), UCF-101 (Soomro, Zamir, and Shah, 2012) et NTU-RGB+D (Shahroudy et al., 2016), nous identifions les architectures profondes les plus performantes (voir les FIGURES E.4 and E.5) qui ont été appliquées avec succès pour la reconnaissance des actions, puis nous fournissons les tendances actuelles et les problèmes qui restent ouverts pour les travaux futurs. De nombreuses bases de données publiques pour la reconnaissance des actions dans des vidéos sont également décrites.

La reconnaissance des actions humaines a rapidement progressé, passant de la reconnaissance des actions dans un environnement simple, contrôlé, avec des bases de données réduites, à la reconnaissance des actions dans des vidéos réalistes à très grande échelle. Sur ce thème, les algorithmes d'apprentissage profond ont joué un rôle important. Dans ce chapitre, nous réalisons une analyse détaillée des travaux existants sur la reconnaissance des actions humaines à partir des vidéos RGB-D utilisant l'apprentissage profond. Cette étude a permis de confirmer que les modèles CNNs sont les plus utilisés pour l'apprentissage des caractéristiques spatio-temporelles du mouvement humain dans les vidéos. Les idées clés qui sous-tendent les CNNs leur permettent de travailler directement à partir des images et d'obtenir des caractéristiques de haut niveau en composant des structures de niveau inférieur. En plus de fonctionner comme une solution de bout en bout, les CNNs ont également été utilisés comme des extracteurs de caractéristiques et constituent l'un des éléments d'un système plus grand, notamment avec les réseaux RNN-LSTMs. Bien que les CNNs obtiennent d'excellentes performances dans plusieurs tâches de reconnaissance des actions, ils nécessitent d'utiliser une immense base de données d'apprentissage. De plus, l'apprentissage d'une architecture CNNs très profonde nécessite beaucoup de calculs. Par conséquent, appliquer des CNNs très profonds à des tâches de reconnaissance d'action en temps réel reste un grand défi.

Les réseaux de neurones récurrents à mémoire court-terme persistante (RNN-LSTMs) ont été conçus pour traiter les séries chronologiques. Ils ont été utilisés avec succès dans la modélisation des informations de contexte à long terme des séquences de mouvement, en particulier avec des données sous forme de squelettes comme dans les travaux de Du, Wang, and Wang, 2015, Song et al., 2017, Zhu et al., 2016b, ou Liu et al., 2016b. Mais la plupart des modèles basés sur les RNN-LSTMs ne peuvent pas fonctionner directement sur des données brutes. Par exemple, les données du squelette doivent être prétraitées avant d'être introduites dans des RNN-LSTMs. La combinaison de CNNs et de LSTMs est un excellent exemple de la façon dont nous pouvons construire des modèles d'apprentissage profond

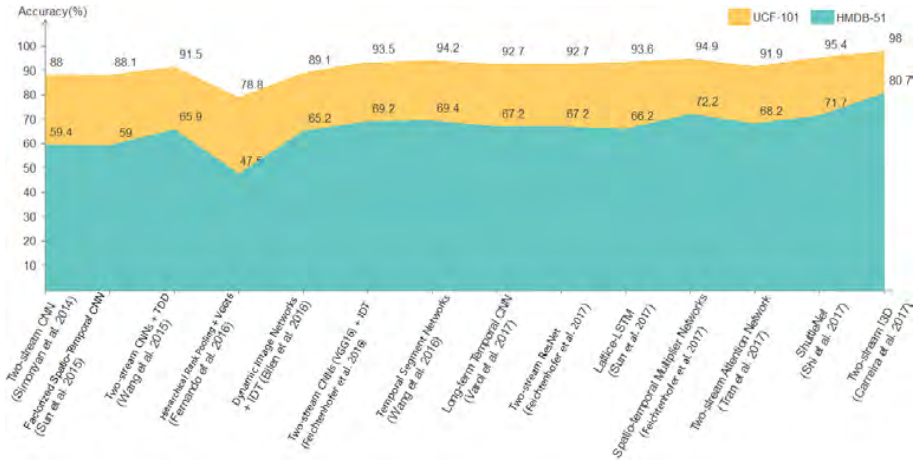


FIGURE E.4: Comparaison de la précision (%) de différentes approches basées sur des algorithmes d'apprentissage profond sur les bases de données HMDB-51 (Kuehne et al., 2011) et UCF-101 (Soomro, Zamir, and Shah, 2012).

plus puissants en tirant parti des avantages de différentes architectures. Dans ce cas, les CNNs ont été utilisés pour extraire des caractéristiques de mouvement de haut niveau à partir des vidéos, tandis que les LSTMs ont été exploités pour l'apprentissage et la prédiction de séquences.

Les réseaux de croyances profonds (DBNs – Hinton, Osindero, and Teh, 2006) et les autoencodeurs (SDAs – Vincent et al., 2008) sont également des choix prometteurs pour les tâches de reconnaissance des actions humaines. L'apprentissage des réseaux DBNs peut être effectué de manière semi-supervisée, avec moins de données étiquetées. Une des limitations des DBNs est qu'ils requièrent des descripteurs prédéfinis (« *handcrafted features* ») (Foggia et al., 2014) ou la conversion des données d'entrée en une forme appropriée (Ali and Wang, 2014). Par ailleurs, les SDAs peuvent apprendre les caractéristiques de mouvement de manière non supervisée et sont capables de générer des caractéristiques robustes. Cependant, ils ont plusieurs inconvénients liés à leur processus d'optimisation. En outre, les approches de reconnaissance des actions basées sur les réseaux antagonistes génératifs (GANs – Goodfellow et al., 2014) récemment introduits ont également montré de grandes possibilités d'apprentissage et de reconnaissance des actions humaines de manière semi-supervisée, bien qu'ils soient difficiles à optimiser.

Dans le futur, les algorithmes d'apprentissage profond continueront d'attirer beaucoup d'attention pour la reconnaissance des actions humaines. Quelques directions de recherche potentielles incluront des modèles d'apprentissage non-supervisés, des réseaux de neurones plus profonds, la combinaison de différentes architectures (par exemple CNNs avec RNN-LSTMs). Par ailleurs, le transfert d'apprentissage et des nouvelles représentations du mouvement 3D pour la reconnaissance des actions sont également très prometteurs.

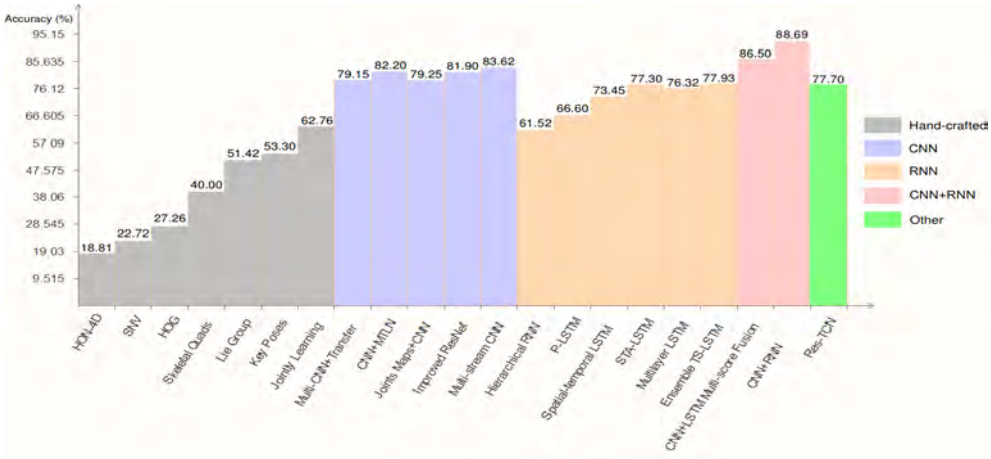


FIGURE E.5: Comparaison de la précision (%) de différentes approches basées sur l'apprentissage profond sur la base de données NTU-RGB+D (Shahroudy et al., 2016). Figure plus lisible en couleur.

Chapitre 4

Méthodes proposées à base d'apprentissage profond pour la reconnaissance des actions humaines 3D à partir de squelettes fournis par des capteurs RGB-D

Les capteurs de profondeur peuvent fournir des informations détaillées sur la structure 3D des mouvements humains à l'aide d'algorithmes d'estimation du squelette en temps réel. Cette source de données est une représentation de haut niveau permettant de décrire des actions humaines de manière précise, efficace et adaptée au problème de l'analyse et de la reconnaissance des actions dans les vidéos. Cependant, concevoir des représentations de mouvement pour la tâche de reconnaissance des actions à partir de séquences de squelettes reste une tâche compliquée. Cette représentation doit être robuste au bruit, invariante aux changements de point de vue de la caméra et donner de bonnes performances de reconnaissance. Les deux principaux défis de cette tâche sont de représenter efficacement les motifs spatio-temporels des mouvements du squelette et de bien apprendre leurs caractéristiques discriminantes pour la tâche de classification. Dans ce chapitre, nous proposons de nouvelles représentations basées sur le squelette pour la reconnaissance des actions dans les vidéos en utilisant des réseaux de neurones convolutifs profonds (D-CNNs). Deux questions clés sont abordées : tout d'abord, comment construire une représentation robuste qui décrit facilement les évolutions spatio-temporelles des mouvements à partir de séquences de squelettes ? Ensuite, comment concevoir des réseaux D-CNNs capables d'apprendre de manière efficace les caractéristiques discriminantes à partir de la nouvelle représentation proposée ? Pour répondre à ces questions, nous proposons de coder les coordonnées 3D des articulations du corps humain représentées dans les séquences de squelettes par une structure spatio-temporelle représentée par une image couleur. Ces images sont capables de représenter les évolutions spatio-temporelles des squelettes et peuvent être apprises efficacement par les D-CNNs. Nous proposons ensuite une architecture d'apprentissage profond basée sur ResNets (Kaiming et al., 2016) pour apprendre les caractéristiques des représentations obtenues et les classer en actions (voir FIGURE E.6). Les résultats expérimentaux sur trois bases de données répresentant une grande diversité d'actions humaines, MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014) et NTU-RGB+D (Shahroudy et al., 2016), montrent que notre méthode permet d'atteindre des performances supérieures à celles de l'état de l'art.

Par la suite, nous proposons deux nouvelles représentations 3D basées sur le squelette, appelées SPMF (Skeleton Pose-Motion Feature) et Enhanced-SPMF. Les SPMF et Enhanced-SPMF sont des représentations compactes sous forme d'images construites à partir des positions des squelettes et de leurs mouvements. Enhanced-SPMF (voir FIGURE E.7) est une extension de SPMF dans laquelle un filtre de lissage et un algorithme d'égalisation d'histogramme adaptative (Adaptive Histogram Equalization – Pizer et al., 1987) ont été appliqués pour réduire l'effet du bruit sur les squelettes et mettre en valeur les motifs locaux de la représentation afin de la rendre plus discriminante. Pour les tâches d'apprentissage et de classification, nous exploitons des réseaux D-CNNs de l'état de l'art, tels que Inception-ResNet-v2 (Szegedy et al., 2017), DenseNet (Huang et al., 2017) et Effective Neural Architecture Search (ENAS - Pham et al., 2018a), afin d'apprendre directement la relation directe entre des séquences de squelettes en entrée et les action correspondantes en sortie via les représentations proposées. Notre méthode est évaluée sur quatre bases de données de référence complexes, comprenant des actions individuelles dans la base MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014), des interactions dans la base SBU Kinect Interaction (Yun et al., 2012a)) et des données multivues à

grande échelle dans la base NTU-RGB+D (Shahroudy et al., 2016). Les résultats expérimentaux montrent que l'approche proposée est plus performante que celles de l'état de lart quel que soit le type d'action (actions individuelles, interactions, etc.).

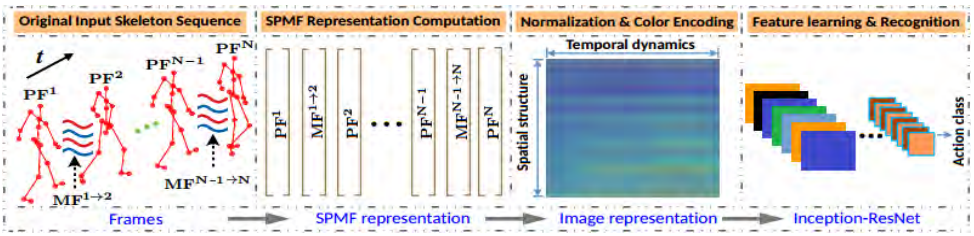


FIGURE E.6: Vue d'ensemble schématique de notre méthode. Chaque séquence de squelettes est codée sous la forme d'une image couleur via une représentation appelée SPMF. Chaque SPMF est construit à partir de vecteurs de positions (PFs) et vecteurs de mouvement (MFs). Ils sont ensuite placés en entrée d'un D-CNN, conçu sur la base de la combinaison de ResNet (Kaiming et al., 2016) et Inception (Szegedy et al., 2016) pour apprendre les caractéristiques discriminantes des SPMF et pouvoir effectuer la classification des actions.

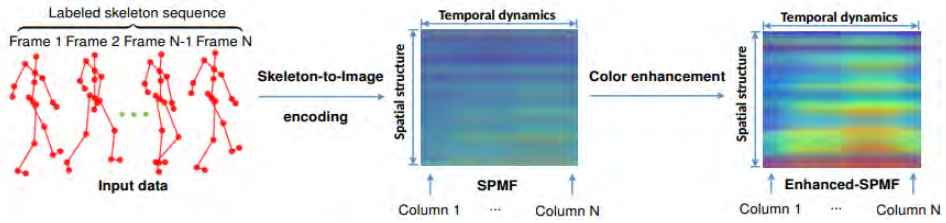


FIGURE E.7: Vue d'ensemble de la représentation proposée Enhanced-SPMF. Chaque séquence de squelettes est transformée en une image couleur RGB qui est une carte de mouvement appelée SPMF. Une technique d'ajustement de contraste (Pizer et al., 1987) est ensuite utilisée pour mettre en évidence les éléments présents dans la carte de mouvement et former Enhanced-SPMF, qui sera appris et classé par un réseau de neurones profond. Avant de calculer la répresentation SPMF, un filtre de lissage est appliqué pour réduire l'effet du bruit sur les positions des points des squelettes.

Chapitre 5

Un système d'apprentissage profond unifié pour l'estimation conjointe de la pose 3D et la reconnaissance des actions humaines à partir de vidéos couleur monoculaires

Ce chapitre représente un travail supplémentaire qui n'était pas initialement prévu dans le programme de la thèse. Les capteurs RGB-D conviennent parfaitement dans un contexte d'application bien défini : en intérieur, en milieu confiné comme dans les autobus et avec une portée de détection située entre 50 cm et environ 5m. Lors de la prise de mesure dans la station de métro à Toulouse, nous avons atteint les limites d'utilisation des capteurs RGB-D. Nous avons alors pensé qu'il pourrait être intéressant, en guise de travail supplémentaire, d'exploiter des capteurs RGB pour extraire des squelettes 3D comme ceux issus des capteurs RGB-D. Notre objectif dans ce chapitre est donc de proposer une approche de reconnaissance des actions humaines basée sur des squelettes 3D obtenus à partir de données vidéo RGB. Plus précisément, nous présentons un système multitâche basé sur des algorithmes d'apprentissage profond pour l'estimation conjointe de poses humaines en 3D et la reconnaissance des actions à partir de séquences vidéo RGB. La méthode est divisée en deux parties. Tout d'abord, nous utilisons le détecteur de pose humaine 2D existant OpenPose (Cao et al., 2017) pour déterminer l'emplacement précis dans chaque image des points clés du corps. Nous avons ensuite conçu un réseau de neurones à deux flux pour produire les positions 3D des points clés 2D. Ensuite, nous avons utilisé l'algorithme ENAS (Efficient Neural Architecture Search – Pham et al., 2018a) pour trouver une architecture de réseau optimale pour modéliser l'évolution spatio-temporelle des poses 3D estimées via une représentation intermédiaire sous la forme d'une image RGB et effectuer la tâche de reconnaissance des actions. Les évaluations sur les bases de données Human3.6M (Ionescu et al., 2014), MSR Action3D (Li, Zhang, and Liu, 2010) et SBU Kinect Interaction (Yun et al., 2012a) vérifient l'efficacité de la méthode proposée pour le passage 2D à 3D et pour la reconnaissance d'actions. Les FIGURES E.8, E.9 et E.10 fournissent des illustrations du fonctionnement de notre méthode.

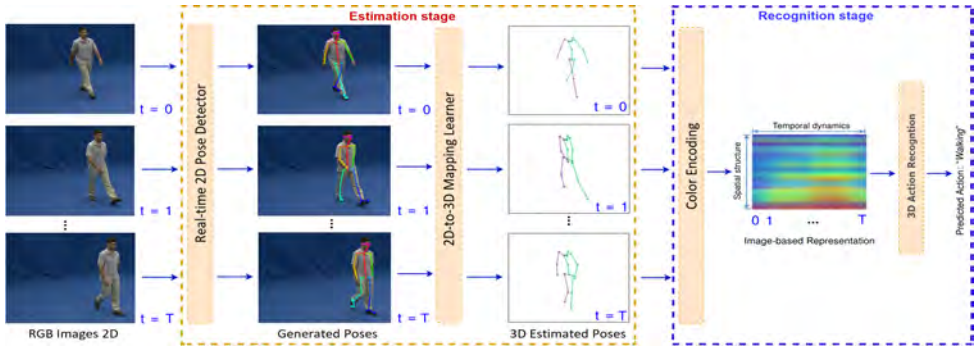


FIGURE E.8: Vue d'ensemble de la méthode proposée. Avant le niveau de l'estimation, nous avons d'abord exécuté OpenPose (Cao et al., 2017) – un détecteur de pose 2D multi-personnes pour générer des points-clés 2D du corps humain en temps réel. Un réseau de neurones profond est ensuite utilisé pour produire des poses 3D à partir des détections 2D. Au niveau de la reconnaissance d'actions, les poses estimées en 3D sont codées dans une représentation compacte à base d'image et finalement introduites dans un réseau de neurones convolutif profond pour la tâche de classification supervisée, qui est automatiquement recherchée par l'algorithme ENAS (Pham et al., 2018a).

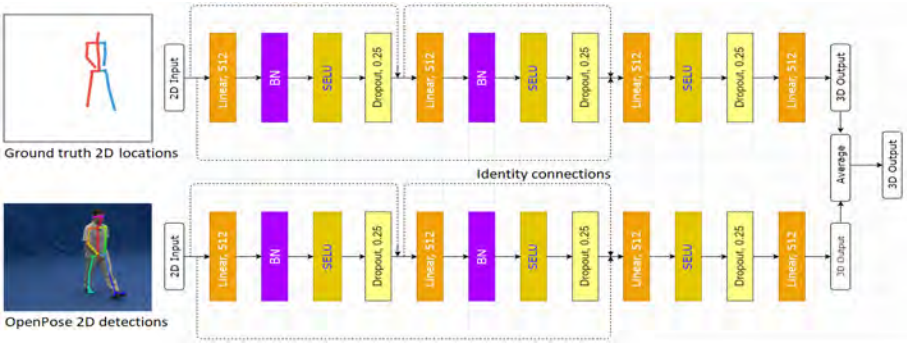


FIGURE E.9: Schéma du réseau à deux flux proposé pour l'entraînement de notre estimateur de pose 3D à partir des vidéos RGB.

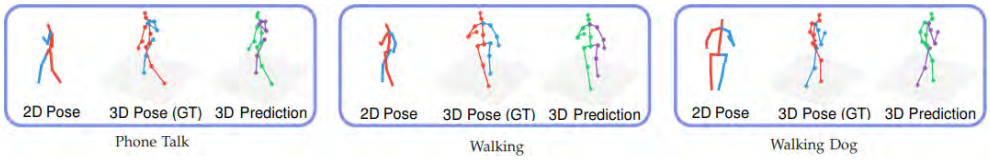


FIGURE E.10: Trois visualisations de la sortie 3D de l'étape d'estimation sur la base de données Human3.6M (Ionescu et al., 2014). Dans chaque exemple, on trouvera de gauche à droite, la pose 2D, la vérité terrain 3D et notre prédiction 3D, respectivement.

Chapitre 6

Conclusions et perspectives

L'objectif principal dans cette thèse a été de proposer, développer et valider un système basé sur l'apprentissage profond pour reconnaître les actions humaines dans des séquences vidéo monoculaires RGB-D. Pour résoudre ce problème, nous avons d'abord analysé les algorithmes d'apprentissage profond les plus importants, appliqués à la reconnaissance des actions humaines dans les vidéos (chapitre 3). Nous avons constaté que les approches basées sur les réseaux de neurones convolutifs profonds (D-CNNs) étaient parmi les modèles les plus performants pour cette tâche. Nous avons ensuite proposé une nouvelle approche basée sur les D-CNNs pour la reconnaissance des actions humaines à partir de squelettes fournis par des caméras de profondeur (chapitre 4). Deux questions clés ont été étudiées et traitées. Tout d'abord, comment représenter efficacement les séquences de squelettes afin d'exploiter pleinement la capacité d'apprentissage des représentations de haut niveau des réseaux D-CNNs ? Ensuite, comment concevoir une architecture de D-CNN capable d'apprendre des caractéristiques discriminantes à partir de la représentation proposée afin de reconnaître les actions ? En conséquence, nous avons introduit deux nouvelles représentations du mouvement 3D appelées SPMF et Enhanced-SPMF, qui représentent les poses et les mouvements des squelettes sous la forme d'images couleur. Les résultats expérimentaux obtenus sur quatre bases de données publiques, MSR Action3D (Wanqing, Zhengyou, and Zicheng, 2010), KARD (Gaglio, Re, and Morana, 2014), SBU Kinect Interaction (Yun et al., 2012a) et NTU-RGB+D (Shahroudy et al., 2016), ont montré l'efficacité des représentations proposées ainsi que notre système d'apprentissage profond proposé.

Notre étude a également montré que la représentation Enhanced-SPMF permet de mieux apprendre les caractéristiques spatio-temporelles du mouvement que la représentation SPMF. Plus précisément, nous avons réalisé une étude d'ablation en entraînant et en évaluant un réseau DenseNet (Huang et al., 2017) en utilisant SPMF et Enhanced-SPMF. La base utilisée était SBU Kinect Interaction (Yun et al., 2012a). Les mêmes valeurs des hyperparamètres et la même stratégie d'apprentissage ont été utilisées dans les deux cas (chapitre 4). Les résultats expérimentaux indiquent que l'apprentissage sur Enhanced-SPMF a conduit à de meilleures performances de reconnaissance. Cela montre que l'utilisation de l'algorithme AHE (Pizer et al., 1987) et du filtre de lissage Savitzky-Golay (Savitzky and Golay, 1964) contribue à améliorer la précision. Pour les tâches d'apprentissage et de classification, nous avons conçu différentes architectures D-CNN basées sur ResNet (Kaiming et al., 2016), Inception-ResNet-v2 (Szegedy et al., 2017), DenseNet (Huang et al., 2017) et Effective Neural Architecture Search (ENAS - Pham et al., 2018a) pour extraire des caractéristiques robustes à partir des représentations sous forme d'images couleur et effectuer le classement. Nous constatons que les comportements d'apprentissage des réseaux de neurones profonds dépendent fortement de la taille et de la distribution des données d'entrée.

Dans cette thèse, nous avons également mené des recherches sur le problème de l'estimation de poses humaines en 3D à partir de séquences vidéo RGB monoculaires et utilisé les poses 3D estimées pour la reconnaissance des actions. Les expérimentations sur les bases de données Human3.6M (Ionescu et al., 2014), MSR Action3D (Li, Zhang, and Liu, 2010) et SBU Kinect Interaction (Yun et al., 2012a) ont montré l'efficacité de la méthode proposée sur les deux tâches. De plus, nous avons collecté et présenté la nouvelle base de données RGB-D CEMEST décrivant le comportement de passagers dans les transports publics. Cette base de données contient un total de 203 vidéos de surveillance réelles, présentant des événements réalistes « normaux » et « anormaux ». Nous avons rendue publique[1] cette base de données, sous forme de squelettes, à des fins de recherche et développement dans ce domaine.

[1]La base de données CEMEST et sa description sont disponibles à l'adresse : https://sites.google.com/site/hhpham172/image-processing-and-computer-vision/tisseo-cerema-dataset.

Nous avons obtenu des résultats prometteurs dans les conditions réelles de CEMEST avec le réseau DenseNet (Huang et al., 2017) et la représentation Enhanced-SPMF, en exploitant des techniques d'augmentation de données et de transfert d'apprentissage. Nous continuons actuellement de mener de nouvelles expérimentations de l'estimation de la pose 3D à partir de vidéos RGB sur CEMEST. Les résultats obtenus seront rapportés dans notre prochaine publication.

Bien que l'efficacité des différentes méthodes que l'on a proposées ait été montrée en termes de précision, certaines limites subsistent. Cela nécessite davantage de travaux de recherche. Tout d'abord, le modèle proposé ne peut pas gérer les données en ligne (ou *Online Action Recognition – OAR*) qui visent à détecter et à reconnaître des actions de manière continue à partir de flux non segmentés, lorsque les frontières entre les différentes actions dans le flux sont inconnues. Notre étude s'est concentrée jusqu'à maintenant sur la reconnaissance des actions à partir de séquences segmentées, chaque segment correspondant à une seule action ou à une interaction. Une approche courante de l'OAR que nous pouvons envisager est la technique de la « fenêtre glissante » (Kviatkovsky, Rivlin, and Shimshoni, 2014; Kulkarni et al., 2015; Zhu et al., 2016a). Cette approche considère la cohérence temporelle dans une fenêtre de prédiction. Cette idée peut également être appliquée pour résoudre notre problème. Cependant, nous comprenons que les performances de ces méthodes sont sensibles à la taille de la fenêtre et qu'une taille de fenêtre trop grande ou trop petite peut entraîner une baisse significative des performances. Ensuite, les réseaux D-CNNs proposés tels que ResNets (Kaiming et al., 2016) ou DenseNets (Huang et al., 2017) sont des réseaux très profonds contenant des millions de paramètres devant être appris. Il est donc irréaliste d'exploiter ces architectures sur des CPUs ou des plates-formes mobiles. Enfin, la méthode proposée pour l'estimation de la pose 3D à partir de séquences vidéo RGB dépend naturellement de la qualité de la sortie des détecteurs 2D. Par conséquent, une limitation est qu'il n'est pas possible de récupérer les poses 3D à partir d'une sortie 2D de mauvaise qualité. Ce problème pourrait être résolu en fournissant davantage d'informations visuelles au réseau, comme les silhouettes couleur des personnes, afin d'accroître les performances. Enfin, nous avons collecté et proposé le jeu de données CEMEST. Nous somme conscients du fait que CEMEST est une petite base de données et que l'entraînement d'algorithmes d'apprentissage supervisés tels que les réseaux D-CNNs pourrait facilement conduire à un surajustement.

De nombreuses directions de recherche potentielles devraient être considérées pour élargir l'approche actuelle. Nous décrivons ici certaines des idées les plus prometteuses. Par exemple, les réseaux de neurones récurrents à mémoire court-terme persistante (RNN-LSTMs – Hochreiter and Schmidhuber, 1997) sont largement utilisés pour la modélisation et la prévision de séries chronologiques. Ce type de réseau peut être utilisé pour modéliser les caractéristiques spatio-temporelles contenues dans les représentations SPMF et Enhanced-SPMF proposées. Les réseaux convolutifs temporels (TCNs – Bai, Kolter, and Koltun, 2018) ont montré que les architectures de convolution peuvent dépasser les réseaux récurrents sur des tâches telles que la synthèse audio et la traduction automatique. Les auteurs ont montré que, pour des problèmes de modélisation de séquences, une architecture convolutive simple est meilleure que les réseaux récurrents tels que les RNN-LSTMs, à travers un large éventail de tâches et de bases de données, tout en montrant une mémoire à long terme. Les représentations de mouvement 3D proposées (SPMF et Enhanced-SPMF) sont construites à partir de deux fonctions d'action : les postures statiques et les mouvements temporels. Les deux caractéristiques ont été combinées dans une image couleur unique et mises en entrée des D-CNNs pour un apprentissage de la représentation. Une autre solution consiste à coder chaque type de caractéristique dans une image et à créer un modèle de réseau neuronal profond à deux flux qui accepte les deux images en tant qu'entrées. Les dernières couches des deux flux seront fusionnées ultérieurement pour améliorer les performances. Par ailleurs, les réseaux temporels reposant sur le mécanisme d'attention (« *Attention Temporal Network* » ou ATN – Zang et al., 2018; Li et al., 2019) peuvent encore améliorer la performance de la reconnaissance des actions humaines dans les vidéos. Au lieu de traiter toutes les images vidéo échantillonnées sur un pied d'égalité, un réseau ATN dispose d'un mécanisme d'attention, ce qui permet de se concentrer automatiquement davantage sur les segments sémantiquement critiques. Cette idée peut également être appliquée aux squelettes (Xie et al., 2018;

Si et al., 2019). Par exemple, Si et al., 2019 ont proposé une architecture profonde appelée AGC-LSTM qui dispose d'un mécanisme d'attention. L'AGC-LSTM est capable d'extraire des caractéristiques discriminantes dans la dynamique spatio-temporelle et d'explorer la relation de cooccurrence entre le domaine spatial et le domaine temporel. Cela permet à cette architecture d'accroître la capacité d'apprendre la représentation sémantique de haut niveau en sélectionnant des informations spatiales discriminantes à partir de points caractéristiques d'un squelette.

Appendix F

Curriculum Vitæ

Huy Hieu PHAM (https://huyhieupham.github.io/) was born on the first of January, 1992 in Nam Dinh, Vietnam. He is currently pursuing a Ph.D. degree in Computer Science at Informatics Research Institute of Toulouse (IRIT), Université Toulouse III - Paul Sabatier and CEREMA Research Center, Toulouse, France. His research focuses on Computer Vision, Machine Learning, and Deep Learning. More specifically, he designs and optimizes high-performance deep learning networks such as Deep Convolutional Neural Networks (D-CNNs) for solving the security problems in public transport, including human action recognition and behavior analysis. From November 2015 to May 2016, he has done a research internship at ICA laboratory, Ecole des Mines d'Albi, France. His work is part of a dynamic multi-partner robotized airplane inspection project, called Air-Cobot , lead by AKKA Technologies and AIRBUS group. Previously, Hieu was a Computer Engineering undergraduate student at the Center for Training of Excellent Students, Hanoi University of Science and Technology (HUST). He has done my graduate internship at International Research Institute MICA, Hanoi, Vietnam and AGIM laboratory, Université Grenoble Alpes, Grenoble, France. The main goal of the graduate project is to develop a computer vision algorithm based on electrode matrix and mobile Kinect for detecting obstacles and warning to visually impaired people.

Hieu will back to Hanoi, Vietnam and work as a Research Scientist in Computer Vision and Artificial Intelligence at the Vingroup Big Data Institute (VinBDI) starting from October 2019.

Bibliography

Abu-El-Haija, Sami et al. (2016). "Youtube-8M: A large-scale video classification benchmark". In: *arXiv preprint arXiv:1609.08675*.

Aggarwal, Jake K and Quin Cai (1999). "Human motion analysis: A review". In: *Computer Vision and Image Understanding* 73, pp. 428–440.

Aggarwal, Jake K and Lu Xia (2014). "Human activity recognition from 3D data: A review". In: *Pattern Recognition Letters* 48, pp. 70–80.

Ahsan, Unaiza, Chen Sun, and Irfan Essa (2018). "DiscrimNet: Semi-supervised action recognition from videos using generative adversarial networks". In: *arXiv preprint arXiv:1801.07230*.

Alfaro, Anali, Domingo Mery, and Alvaro Soto (2016). "Action recognition in video using sparse coding and relative features". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2688–2697.

Ali, K. H. and T. Wang (2014). "Learning features for action recognition and identity with deep belief networks". In: *International Conference on Audio, Language and Image Processing (ICALIP)*, pp. 129–132.

Aliakbarian, Mohammad Sadegh et al. (2017). "Encouraging LSTMs to anticipate actions very early". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 280–289.

Altman, Naomi S (1992). "An introduction to kernel and nearest-neighbor nonparametric regression". In: *The American Statistician* 46.3, pp. 175–185.

ASUS (2018). *ASUS Xtion Pro depth sensor*. https://www.asus.com/3D-Sensor/Xtion_PRO/. Accessed: 2018-01-09.

Baccouche, Moez et al. (2011). "Sequential deep learning for human action recognition". In: *International Workshop on Human Behavior Understanding (HBU)*, pp. 29–39.

Baek, Seungryul et al. (2016). "Kinematic-layout-aware random forests for depth-based action recognition". In: *arXiv preprint arXiv:1607.06972*.

Bai, Shaojie, J Zico Kolter, and Vladlen Koltun (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Ballan, Lamberto et al. (2012). "Effective codebooks for human action representation and classification in unconstrained videos". In: *IEEE Transactions on Multimedia* 14.4, pp. 1234–1245.

Barret, Z. and V. L. Quoc (2017). "Neural architecture search with reinforcement learning". In: *arXiv preprint arXiv:1611.01578*.

Beaudry, Cyrille, Renaud Péteri, and Laurent Mascarilla (2016). "An efficient and sparse approach for large scale human action recognition in videos". In: *Machine Vision and Applications* 27, pp. 529–543.

Bengio, Y., P. Simard, and P. Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE Transactions on Neural Networks* 5.2, pp. 157–166. ISSN: 1045-9227. DOI: 10.1109/72.279181.

Bilen, Hakan et al. (2016). "Dynamic image networks for action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3034–3042.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent dirichlet allocation". In: *Journal of Machine Learning Research (JMLR)*. Vol. 3, pp. 993–1022.

Bottou, Léon (2010). "Large-scale machine learning with stochastic gradient descent". In: *International Conference on Computational Statistics (COMPSTAT)*, pp. 177–186.

Brahnam, Sheryl and Loris Nanni (2009). "High performance set of features for human action classification". In: *International Conference on Image Processing, Computer Vision, & Pattern Recognition (IPCV)*, pp. 980–984.

Cao, Congqi et al. (July, 2016). "Action recognition with joints-pooled 3D deep convolutional descriptors". In: *International Joint Conference on Artificial Intelligence (IJCAI)*. Vol. 1, p. 3.

Cao, Zhe et al. (2017). "Realtime multi-person 2D pose estimation using part affinity fields". In: *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Carreira, Joao and Andrew Zisserman (2017). "Quo vadis, action recognition? a new model and the kinetics dataset". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 4724–4733.

Castro, Daniel et al. (2015). "Predicting daily activities from egocentric images using deep learning". In: *The ACM International Symposium on Wearable Computers (ISWC)*, pp. 75–82.

Chaaraoui, Alexandros Andre, Jose Ramon Padilla-Lopez, and Francisco Florez-Revuelta (2013). "Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices". In: *International Conference on Computer Vision (ICCV)*, pp. 91–97.

Chatfield, Ken et al. (2014). "Return of the devil in the details: Delving deep into convolutional nets". In: *arXiv preprint arXiv:1405.3531*.

Chaudhry, Rizwan et al. (2013). "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 471–478.

Chen, Chen, Roozbeh Jafari, and Nasser Kehtarnavaz (2015). "Action recognition from depth sequences using depth motion maps-based local binary patterns". In: *IEEE Winter Conference on Applications of Computer Vision (WCAV)*, pp. 1092–1099.

Chen, Chen, Kui Liu, and Nasser Kehtarnavaz (2013). "Real-time human action recognition based on depth motion maps". In: *J. Real-Time Image Processing* 12.

Chen, Minmin et al. (2012). "Marginalized stacked denoising autoencoders". In: *Proceedings of the Learning Workshop*. Vol. 36, pp. 7–15.

Cheng, Guangchun et al. (2015). "Advances in human action recognition: a survey". In: *arXiv preprint arXiv:1501.05964*.

Chéron, Guilhem, Ivan Laptev, and Cordelia Schmid (2015). "P-CNN: Pose-based CNN features for action recognition". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 3218–3226.

Cippitelli, E. et al. (2016a). "Evaluation of a skeleton-based method for human activity recognition on a large-scale RGB-D dataset". In: *IET International Conference on Technologies for Active and Assisted Living (TechAAL 2016)*, pp. 1–6.

Cippitelli, Enea et al. (2016b). "A human activity recognition system using skeleton data from RGB-D sensors". In: *Computational Intelligence and Neuroscience* 2016, p. 21.

Clevert, D., T. Unterthiner, and S. Hochreiter (2015). "Fast and accurate deep network learning by exponential linear units (ELUs)". In: *arXiv preprint arXiv:1511.07289*.

Cortes, Corinna and Vladimir Vapnik (1995). "Support-vector networks". In: *Machine learning* 20.3, pp. 273–297.

Cruz, Leandro, Djalma Lucio, and Luiz Velho (2012). "Kinect and RGB-D images: Challenges and applications". In: *SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pp. 36–49.

Dalal, N. and B. Triggs (2005). "Histograms of oriented gradients for human detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

Devanne, Maxime et al. (2013). "Space-time pose representation for 3D human action recognition". In: *International Conference on Image Analysis and Processing (ICIAP)*, pp. 456–464.

Ding, Wenwen et al. (2016). "Profile HMMs for skeleton-based human action recognition". In: *Signal Processing: Image Communication* 42, pp. 109–119.

Dobhal, Tushar et al. (2015). "Human activity recognition using binary motion image and deep learning". In: *Procedia Computer Science* 58, pp. 178–185.

Dollár, Piotr et al. (2005). "Behavior recognition via sparse spatio-temporal features". In: *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS)*, pp. 65–72.

Donahue, Jeffrey et al. (2015). "Long-term recurrent convolutional networks for visual recognition and description". In: *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2625–2634.

Du, Wenbin, Yali Wang, and Yu Qiao (2017). "RPAN: An end-to-end recurrent pose-attention network for action recognition in videos". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3725–3734.

Du, Y. et al. (2016). "Marker-less 3D human motion capture with monocular image sequence and height-maps". In: *European Conference on Computer Vision (ECCV)*, pp. 20–36.

Du, Yong, Wei Wang, and Liang Wang (2015). "Hierarchical recurrent neural network for skeleton based action recognition". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1110–1118.

Eitel, Andreas et al. (2015). "Multimodal deep learning for robust RGB-D object recognition". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, pp. 681–687.

Evangelidis, Georgios, Gurkirt Singh, and Radu Horaud (2014). "Skeletal quads: Human action recognition using joint quadruples". In: *International Conference on Pattern Recognition (ICPR)*, pp. 4513–4518.

Fan, Jialue et al. (2010). "Human tracking using convolutional neural networks". In: *IEEE Transactions on Neural Networks* 21, pp. 1610–1623.

Feichtenhofer, Christoph, Axel Pinz, and Richard Wildes (2016). "Spatiotemporal residual networks for video action recognition". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3468–3476.

Feichtenhofer, Christoph, Axel Pinz, and Richard P Wildes (2017). "Spatiotemporal multiplier networks for video action recognition". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 7445–7454.

Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman (2016). "Convolutional two-stream network fusion for video action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941.

Fernando, Basura et al. (2016). "Discriminative hierarchical rank pooling for activity recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1924–1932.

Fisher, Y. and K. Vladlen (2015). "Multi-scale context aggregation by dilated convolutions". In: *arXiv preprint arXiv:1511.07122*.

Foggia, P. et al. (2014). "Exploiting the deep learning paradigm for recognizing human actions". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 93–98.

Foggia, Pasquale et al. (2013). "Recognizing human actions by a bag of visual words". In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2910–2915.

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological Cybernetics*, pp. 267–285.

Gaglio, Salvatore, Giuseppe Lo Re, and Marco Morana (2014). "Human activity recognition process using 3-D posture data". In: *IEEE Transactions on Human-Machine Systems* 45.5, pp. 586–597.

Gan, Chuang et al. (2015). "Devnet: A deep event network for multimedia event detection and evidence recounting". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2568–2577.

Gao, Zan et al. (2014). "Human action recognition via multi-modality information". In: *Journal of Electrical Engineering and Technology* 9.2, pp. 739–748.

Giese, Martin A and Tomaso Poggio (2003). "Cognitive neuroscience: Neural mechanisms for the recognition of biological movements". In: *Nature Reviews Neuroscience* 4, p. 179.

Gkioxari, Georgia et al. (2014). "R-CNNs for pose estimation and action detection". In: *arXiv preprint arXiv:1406.5212*.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 249–256.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (2011). "Deep sparse rectifier neural networks". In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 315–323.

Gong, Dian, Gérard G. Medioni, and Xuemei Zhao (2014). "Structured time series analysis for human action segmentation and recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 36, pp. 1414–1427.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). "Deep learning". MIT Press. URL: http://www.deeplearningbook.org.

Goodfellow, Ian et al. (2014). "Generative adversarial nets". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 2672–2680.

Gorban, A. et al. (2015). *THUMOS Challenge: Action recognition with a large number of classes*. http://www.thumos.info/. Accessed: 2019-04-21.

Gorelick, L. et al. (2007). "Actions as space-time shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 29, pp. 2247–2253.

Gowayyed, M.A. et al. (2013). "Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition". In: *IJCAI*.

Graves, Alex (2008). "Supervised sequence labelling with recurrent neural networks". In: *Studies in Computational Intelligence*.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013). "Speech recognition with deep recurrent neural networks". In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pp. 6645–6649.

Gu, Feng et al. (2015). "Marginalised stacked denoising Autoencoders for robust representation of real-time multi-View action recognition". In: *Sensors* 15, pp. 17209–17231.

Guha, Tanaya and Rabab K Ward (2012). "Learning sparse representations for human action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 34, pp. 1576–1588.

Guo, Guodong and Alice Lai (2014). "A survey on still image based human action recognition". In: *Pattern Recognition* 47, pp. 3343–3361.

Han, Jungong et al. (2013). "Enhanced computer vision with Microsoft Kinect sensor: A review". In: *IEEE Transactions on Cybernetics* 43.5, pp. 1318–1334.

Hasan, Mahmudul and Amit K Roy-Chowdhury (2014). "Continuous learning of human activity models using deep nets". In: *European Conference on Computer Vision (ECCV)*, pp. 705–720.

He, Kaiming and Jian Sun (2015). "Convolutional neural networks at constrained time cost". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5353–5360.

He, Kaiming et al. (2015). "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034.

Heckbert, Paul (1995). "Fourier transforms and the fast Fourier transform (FFT) algorithm". In: *Computer Graphics* 2, pp. 15–463.

Heilbron, F. C. et al. (2015). "ActivityNet: A large-scale video benchmark for human activity understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970. DOI: 10.1109/CVPR.2015.7298698.

Hieu Pham, Huy et al. (2018). "Skeletal movement to color map: A novel representation for 3D action recognition with Inception residual networks". In: *IEEE International Conference on Image Processing (ICIP)*, pp. 3483–3487.

Hinton, Geoffrey (2010). "A practical guide to training restricted Boltzmann machines". In: *Momentum*, pp. 599–619.

Hinton, Geoffrey E (2002). "Training products of experts by minimizing contrastive divergence". In: *Neural Computation* 14, pp. 1771–1800.

Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural Computation* 18, pp. 1527–1554.

Hinton, Geoffrey E, Terrence J Sejnowski, and David H Ackley (1984). *Boltzmann machines: Constraint satisfaction networks that learn*. Carnegie-Mellon University, Department of Computer Science Pittsburgh, PA.

Hinton, Geoffrey E et al. (2012). "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580*.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: http://dx.doi.org/10.1162/neco.1997.9.8.1735.

Hou, Y. et al. (2017). "Skeleton optical spectra based action recognition using convolutional neural networks". In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* PP.99, pp. 1–1. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2016.2628339.

Hsu, Eugene, Kari Pulli, and Jovan Popović (2005). "Style translation for human motion". In: *ACM Transactions on Graphics (TOG)*. Vol. 24, pp. 1082–1089.

Hu, Jian-Fang et al. (2015a). "Jointly learning heterogeneous features for RGB-D activity recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5344–5352.

Hu, Jianfang et al. (2015b). "Jointly learning heterogeneous features for RGB-D activity recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5344–5352.

Huang, Fu Jie, Y-Lan Boureau, and Yann LeCun (2007). "Unsupervised learning of invariant feature hierarchies with applications to object recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Huang, G. et al. (2017). "Densely connected convolutional networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

Hubel, David H and Torsten N Wiesel (1962). "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex". In: *The Journal of Physiology* 160, pp. 106–154.

Huber, P. J. (1992). "Robust estimation of a location parameter". In: *Breakthroughs in Statistics*. Springer, pp. 492–518.

Hussein, Mohamed E. et al. (2013). "Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations". In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI '13. Beijing, China: AAAI Press, pp. 2466–2472. ISBN: 978-1-57735-633-2. URL: http://dl.acm.org/citation.cfm?id=2540128.2540483.

Ibrahim, Mostafa S. et al. (2016a). "A hierarchical deep temporal model for group activity recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1971–1980.

Ibrahim, Mostafa S et al. (2016b). "A hierarchical deep temporal model for group activity recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1971–1980.

Ikizler, Nazlı and Pınar Duygulu (2007). "Human action recognition using distribution of oriented rectangular patches". In: *Human Motion–Understanding, Modeling, Capture and Animation*, pp. 271–284.

Ilya, L. and H. Frank (2016). "SGDR: Stochastic gradient descent with warm restarts". In: *arXiv preprint arXiv:1608.03983*.

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *International Conference on Machine Learning (ICML)*, pp. 448–456.

Ionescu, C. et al. (2014). "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 36.7, pp. 1325–1339.

Isola, Phillip et al. (2017). "Image-to-image translation with conditional adversarial networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1125–1134.

Ivan, Laptev (2012). "Action recognition using rate-invariant analysis of skeletal shape trajectories". In: *The INRIA Computer Vision and Machine Learning Summer School Grenoble*.

Jaeyong Sung et al. (2012). "Unstructured human activity detection from RGB-D images". In: *2012 IEEE International Conference on Robotics and Automation*, pp. 842–849. DOI: 10.1109/ICRA.2012.6224591.

Jain, Arjun et al. (2013). "Learning human pose estimation features with convolutional networks". In: *arXiv preprint arXiv:1312.7302*.

Jain, Arjun et al. (2014). "Modeep: A deep learning framework using motion features for human pose estimation". In: *Asian Conference on Computer Vision (ACCV)*, pp. 302–315.

Jhuang, Hueihan (2007). "A biologically inspired system for action recognition". PhD thesis. PhD Thesis - Massachusetts Institute of Technology.

Ji, Shuiwang et al. (2013). "3D convolutional neural networks for human action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 35, pp. 221–231.

Ji, Yanli, Guo Ye, and Hong Cheng (2014). "Interactive body part contrast mining for human interaction recognition". In: *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pp. 1–6.

Jiang, Y.-G. et al. (2014). *THUMOS challenge: Action recognition with a large number of classes*. http://crcv.ucf.edu/THUMOS14/. Accessed: 2019-04-21.

Jin, Ke et al. (2017). "Action recognition using vague division depth motion maps". In: *The Journal of Engineering* 1.1.

Kaiming, He et al. (2016). "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Kang, Soo Min and Richard P Wildes (2016). "Review of action recognition and detection methods". In: *arXiv preprint arXiv:1610.06906*.

Karpathy, A. et al. (2014). "Large-scale video classification with convolutional neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732. DOI: 10.1109/CVPR.2014.223.

Karpathy, Andrej et al. (2014). "Large-scale video classification with convolutional neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732.

Katircioglu, I. et al. (2018). "Learning latent representations of 3D human pose with deep neural networks". In: *International Journal of Computer Vision (IJCV)* 126.12, pp. 1326–1341.

Ke, Qiuhong et al. (2017). "A new representation of skeleton sequences for 3d action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4570–4579.

Ke, Shian-Ru et al. (2013). "A review on video-based human activity recognition". In: *Computers* 2, pp. 88–131.

Kim, Ho-Joon, Joseph S Lee, and Hyun-Seung Yang (2007). "Human action recognition using a modified convolutional neural network". In: *International Symposium on Neural Networks (ISNN)*, pp. 715–723.

Kim, Ho-Joon, Juho Lee, and Hyun-Seung Yang (2006). "A weighted FMM neural network and its application to face detection". In: *International Conference on Neural Information Processing (ICONIP)*, pp. 177–186.

Kim, Tae Soo and Austin Reiter (2017). "Interpretable 3D human action analysis with temporal convolutional networks". In: *arXiv preprint arXiv:1704.04516*.

Kingma, D. and J. Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Klambauer, G. et al. (2017). "Self-normalizing neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 971–980.

Klaser, Alexander, Marcin Marszałek, and Cordelia Schmid (2008). "A spatio-temporal descriptor based on 3D-gradients". In: *British Machine Vision Conference (BMVC)*, pp. 275–1.

Kooij, J. F. P., N. Schneider, and D. M. Gavrila (2014). "Analysis of pedestrian dynamics from a vehicle perspective". In: *IEEE Intelligent Vehicles Symposium Proceedings (IVSP)*, pp. 1445–1450. DOI: 10.1109/IVS.2014.6856505.

Koppula, Hema Swetha, Rudhir Gupta, and Ashutosh Saxena (2013). "Learning human activities and object affordances from RGB-D videos". In: *The International Journal of Robotics Research* 32, pp. 951–970.

Koppula, Hema Swetha and Ashutosh Saxena (2013). "Learning spatio-temporal structure from RGB-D videos for human activity detection and anticipation". In: *International Conference on Machine Learning (ICML)*, pp. 792–800.

Krizhevsky, Alex (2009). "Learning multiple layers of features from tiny images". In: *Tech Report*.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012a). "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton (2012b). "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 1097–1105.

Kuehne, H. et al. (2011). "HMDB: A large video database for human motion recognition". In: *International Conference on Computer Vision (ICCV)*, pp. 2556–2563. DOI: 10.1109/ICCV.2011.6126543.

Kuehne, Hildegard et al. (2011). "HMDB: a large video database for human motion recognition". In: *International Conference on Computer Vision (ICCV)*, pp. 2556–2563.

Kulkarni, Kaustubh et al. (2015). "Continuous action recognition based on sequence alignment". In: *International Journal of Computer Vision* 112.1, pp. 90–114.

Kviatkovsky, Igor, Ehud Rivlin, and Ilan Shimshoni (2014). "Online action recognition using covariance of shape and motion". In: *Computer Vision and Image Understanding* 129, pp. 15–26.

Laptev, I. et al. (2008). "Learning realistic human actions from movies". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. DOI: 10.1109/CVPR.2008.4587756.

Laptev, Ivan (2005). "On space-time interest points". In: *International Journal of Computer Vision* 64, pp. 107–123.

Laptev, Ivan et al. (2008). "Learning realistic human actions from movies". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Lea, Colin et al. (2017). "Temporal convolutional networks for action segmentation and detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 156–165.

Lecun, Y. et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). "Deep learning". In: *Nature* 521, p. 436.

LeCun, Yann et al. (1989a). "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1, pp. 541–551.

LeCun, Yann et al. (1989b). "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1, pp. 541–551.

LeCun, Yann et al. (1998a). "Efficient backprop". In: *Neural networks: Tricks of the trade*. Springer, pp. 9–50.

LeCun, Yann et al. (1998b). "Effiicient backProp". In: *Neural networks: Tricks of the trade*. London, UK, UK: Springer-Verlag, pp. 9–50. ISBN: 3-540-65311-2. URL: http://dl.acm.org/citation.cfm?id=645754.668382.

Ledig, Christian et al. (2017). "Photo-realistic single image super-resolution using a generative adversarial network". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4681–4690.

Lee, Honglak et al. (2006). "Efficient sparse coding algorithms". In: *Proceedings of the 19th International Conference on Neural Information Processing Systems*. NIPS'06. Vancouver, Canada: MIT Press Cambridge, MA, USA, pp. 801–808. URL: http://dl.acm.org/citation.cfm?id=2976456.2976557.

Lee, Honglak et al. (2009). "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations". In: *International Conference on Machine Learning (ICML)*, pp. 609–616.

Lee, Inwoong et al. (2017). "Ensemble deep learning for skeleton-based action recognition using temporal sliding LSTM networks". In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1012–1020.

Li, Chuankun et al. (2017a). "Joint distance maps based action recognition with convolutional neural networks". In: *IEEE Signal Processing Letters* 24, pp. 624–628.

Li, Chuankun et al. (2017b). "Skeleton-based action recognition using LSTM and CNN". In: *IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 585–590.

Li, Dong et al. (2019). "Unified spatio-temporal attention networks for action recognition in videos". In: *IEEE Transactions on Multimedia* 21.2, pp. 416–428.

Li, Qing et al. (2016a). "Action recognition by learning deep multi-granular spatio-temporal video representation". In: *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval (ACM)*, pp. 159–166.

Li, S. and A. B. Chan (2014). "3D human pose estimation from monocular images with deep convolutional neural network". In: *Asian Conference on Computer Vision (ACCV)*, pp. 332–347.

Li, Wanqing, Zhengyou Zhang, and Zicheng Liu (2010). "Action recognition based on a bag of 3D points". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9–14.

Li, Wenbo et al. (2015). "Category-blind human action recognition: A practical recognition system". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 4444–4452.

Li, Xinyu et al. (2017c). "Region-based activity recognition using conditional GAN". In: *Proceedings of the 2017 ACM on Multimedia Conference*. ACM, pp. 1059–1067.

Li, Yanghao et al. (2016b). "Online human action detection using joint classification-regression recurrent neural networks". In: *European Conference on Computer Vision (ECCV)*, pp. 203–220.

Liang, B. and L. Zheng (2013). "Three dimensional motion trail model for gesture recognition". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 684–691.

Ling, Jiaxu, Lihua Tian, and Chen Li (2016). "3D human activity recognition using skeletal data from RGB-D sensors". In: *International Symposium on Visual Computing (ISVC)*, pp. 133–142.

Liu, A. A. et al. (2017a). "Benchmarking a multimodal and multiview and interactive dataset for human action recognition". In: *IEEE Transactions on Cybernetics* 47.7, pp. 1781–1794. ISSN: 2168-2267. DOI: 10.1109/TCYB.2016.2582918.

Liu, Anan et al. (2017b). "Hierarchical clustering multi-task learning for joint human action grouping and recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39, pp. 102–114.

Liu, Cong et al. (2015). "Learning motion and content-dependent features with convolutions for action recognition". In: *Multimedia Tools and Applications* 75, pp. 13023–13039.

Liu, Fang et al. (2016a). "Simple to complex transfer learning for action recognition". In: *IEEE Transactions on Image Processing* 25, pp. 949–960.

Liu, J., Jiebo Luo, and M. Shah (2009). "Recognizing realistic actions from videos "in the wild"". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1996–2003. DOI: 10.1109/CVPR.2009.5206744.

Liu, Jun et al. (2016b). "Spatio-temporal LSTM with trust gates for 3D human action recognition". In: *European Conference on Computer Vision (ECCV)*, pp. 816–833.

Liu, Jun et al. (2017c). "Global context-aware attention LSTM networks for 3d action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4674–4683.

Liu, Jun et al. (2018). "Skeleton-based human action recognition with global context-aware attention LSTM networks". In: *IEEE Transactions on Image Processing* 27.4, pp. 1586–1599.

Liu, Li, Ling Shao, and Peter Rockett (2012). "Genetic programming-evolved spatio-temporal descriptor for human action recognition". In: *British Machine Vision Conference (BMVC)*, pp. 1–12.

Liu, Mengyuan, Hong Liu, and Chen Chen (2017). "Enhanced skeleton visualization for view invariant human action recognition". In: *Pattern Recognition*, pp. 346–362.

Lowe, David G (2004). "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision (IJCV)* 60.2, pp. 91–110.

Lu, Zhiwu and Yuxin Peng (2013). "Latent semantic learning with structured sparse representation for human action recognition". In: *Pattern Recognition* 46, pp. 1799–1809.

Luo, Jiajia, Wei Wang, and Hairong Qi (2013). "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps". In: *International Conference on Computer Vision (ICCV)*, pp. 1809–1816.

Luo, Zelun et al. (2017). "Unsupervised learning of long-term motion dynamics for videos". In: pp. 2203–2212.

Luong, Minh-Thang, Hieu Pham, and Christopher D Manning (2015). "Effective approaches to attention-based neural machine translation". In: *arXiv preprint arXiv:1508.04025*.

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

Lv, Fengjun and Ramakant Nevatia (2006). "Recognition and segmentation of 3D human action using HMM and multi-class Adaboost". In: *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*. ECCV'06, pp. 359–372.

Mahasseni, Behrooz and Sinisa Todorovic (2016). "Regularizing long short term memory with 3D human-skeleton sequences for action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3054–3062.

Maqueda, A. I. et al. (2015). "Human-action recognition module for the new generation of augmented reality applications". In: *International Symposium on Consumer Electronics (ISCE)*, pp. 262–264. DOI: 10.1109/ISCE.2015.7177833.

Marszalek, M., I. Laptev, and C. Schmid (2009). "Actions in context". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2929–2936. DOI: 10.1109/CVPR.2009.5206557.

Martinez, J. et al. (2017). "A simple yet effective baseline for 3D human pose estimation". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 2640–2649.

Mehta, D. et al. (2017a). "Monocular 3D human pose estimation in the wild using improved CNN supervision". In: *International Conference on 3D Vision (3DV)*, pp. 506–516.

Mehta, D. et al. (2017b). "VNect: Real-time 3D human pose estimation with a single RGB camera". In: *ACM Transactions on Graphics (TOG)* 36.4, p. 44.

Michael, Ryoo and Aggarwal Jake (2009). "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities." In: *ICCV*. Vol. 1, p. 2.

Microsoft (2014). *Kinect for Windows - Human interface guidelines v2.0*. Tech. rep.

Mirza, Mehdi and Simon Osindero (2014). "Conditional generative adversarial nets". In: *arXiv preprint arXiv:1411.1784*.

Misra, Ishan, C Lawrence Zitnick, and Martial Hebert (2016). "Shuffle and learn: unsupervised learning using temporal order verification". In: *European Conference on Computer Vision (ECCV)*, pp. 527–544.

Mnih, Volodymyr, Nicolas Heess, Alex Graves, et al. (2014). "Recurrent models of visual attention". In: *Advances in Neural Information Processing Systems*, pp. 2204–2212.

Mo, Lingfei et al. (2016). "Human physical activity recognition based on computer vision with deep learning model". In: *IEEE International on Instrumentation and Measurement Technology Conference Proceedings (I2MTC)*, pp. 1–6.

Moeslund, Thomas B and Erik Granum (2001). "A survey of computer vision-based human motion capture". In: *Computer Vision and Image Understanding* 81, pp. 231–268.

Moeslund, Thomas B, Adrian Hilton, and Volker Krüger (2006). "A survey of advances in vision-based human motion capture and analysis". In: *Computer Vision and Image Understanding* 104, pp. 90–126.

Moez, Baccouche et al. (2012). "Spatio-temporal convolutional sparse Autoencoder for sequence classification". In: *British Machine Vision Conference (BMVC)*, pp. 1–12.

Nair, Vinod and Geoffrey E. Hinton (2009). "3D object recognition with deep belief nets". In: *Proceedings of the 22Nd International Conference on Neural Information Processing Systems*. NIPS'09. Vancouver, Canada: Curran Associates Inc., pp. 1339–1347. ISBN: 978-1-61567-911-9. URL: http://dl.acm.org/citation.cfm?id=2984093.2984244.

Nair, Vinod and Geoffrey E Hinton (2010). "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 807–814.

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

Ng, Joe Yue-Hei et al. (2015). "Beyond short snippets: Deep networks for video classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702.

Nie, B. X., C. Xiong, and S. Zhu (2015). "Joint action recognition and pose estimation from video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1293–1301.

Niebles, Juan Carlos, Chih-Wei Chen, and Li Fei-Fei (2010). "Modeling temporal structure of decomposable motion segments for activity classification". In: *European Conference on Computer Vision (ECCV)*, pp. 392–405.

Nowlan, Steven J. and John C. Platt (1994). "A convolutional neural network hand tracker". In: *Proceedings of the 7th International Conference on Neural Information Processing Systems*. NIPS'94. Denver, Colorado: MIT Press Cambridge, MA, USA, pp. 901–903. URL: http://dl.acm.org/citation.cfm?id=2998687.2998799.

Ofli, F. et al. (2013). "Berkeley MHAD: A comprehensive multimodal human action database". In: *IEEE Workshop on Applications of Computer Vision (WACV)*, pp. 53–60.

Oh, S. et al. (2011). "AVSS 2011 demo session: A large-scale benchmark dataset for event recognition in surveillance video". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pp. 527–528.

Ohn-Bar, Eshed and Mohan Trivedi (2013). "Joint angles similarities and HOG2 for action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 465–470.

Olshausen, B A and D J Field (1996). "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381, p. 607.

Oreifej, Omar and Zicheng Liu (2013). "HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 716–723.

Park, S., J. Hwang, and N. Kwak (2016). "3D human pose estimation using convolutional neural networks with 2D pose information". In: *European Conference on Computer Vision (ECCV)*, pp. 156–169.

Pavlakos, G. et al. (2017). "Coarse-to-fine volumetric prediction for single-image 3D human pose". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7025–7034.

Pavllo, D. et al. (2018). "3D human pose estimation in video with temporal convolutions and semi-supervised training". In: *arXiv preprint arXiv:1811.11742*.

Peng, Xiaojiang et al. (2016). "Bag of visual words and fusion methods for action recognition: comprehensive study and good practice". In: *Computer Vision and Image Understanding (CVIU)* 150, pp. 109–125.

Pham, H. et al. (2018a). "Efficient neural architecture search via parameters sharing". In: *International Conference on Machine Learning (ICML)*, pp. 4095–4104.

Pham, Huy-Hieu et al. (2018b). "Exploiting deep residual networks for human action recognition from skeletal data". In: *Computer Vision and Image Understanding* 170, pp. 51–66.

Phung, Son Lam and Abdesselam Bouzerdoum (2007). "A pyramidal neural network for visual pattern recognition". In: *IEEE Transactions on Neural Networks* 18, pp. 329–343.

Pickering, C. A., K. J. Burnham, and M. J. Richardson (2007). "A research study of hand gesture recognition technologies and applications for human vehicle interaction". In: *The 3rd Conference on Automotive Electronics - Institution of Engineering and Technology*, pp. 1–15.

Pizer, Stephen M et al. (1987). "Adaptive histogram equalization and its variations". In: *Computer Vision, Graphics, and Image Processing* 39.3, pp. 355–368.

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

Poppe, Ronald (2010). "A survey on vision-based human action recognition". In: *Image and Vision Computing* 28.6, pp. 976–990.

Presti, Liliana Lo and Marco La Cascia (2016). "3D skeleton-based human action classification: A survey". In: *Pattern Recognition* 53, pp. 130–147.

Qin, Shuxin, Yiping Yang, and Yongshi Jiang (2013). "Gesture recognition from depth images using motion and shape features". In: *International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, pp. 172–175.

Radford, Alec, Luke Metz, and Soumith Chintala (2015). "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434*.

Rahmani, Hossein and Mohammed Bennamoun (2017). "Learning action recognition model from depth and skeleton videos". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 5832–5841.

Rahmani, Hossein and Ajmal Mian (2016). "3D action recognition from novel viewpoints". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1506–1515.

Rahmani, Hossein, Ajmal Mian, and Mubarak Shah (2018). "Learning a deep model for human action recognition from novel viewpoints". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40, pp. 667–681.

Rahmani, Hossein et al. (2016). "Histogram of oriented principal components for cross-view action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38, pp. 2430–2443.

Raina, Rajat et al. (2007). "Self-taught learning: transfer learning from unlabeled data". In: *International Conference on Machine Learning (ICML)*, pp. 759–766.

Ramakrishna, V., T. Kanade, and Y. Sheikh (2012). "Reconstructing 3D human pose from 2D image landmarks". In: *European Conference on Computer Vision (ECCV)*, pp. 573–586.

Ranasinghe, Suneth, Fadi Al Machot, and Heinrich C Mayr (2016). "A review on applications of activity recognition systems with regard to performance and evaluation". In: *International Journal of Distributed Sensor Networks* 12.8.

Reddy, Kishore K. and Mubarak Shah (2013). "Recognizing 50 human action categories of web videos". In: *Machine Vision and Applications* 24, pp. 971–981.

Reed, Scott et al. (2016). "Generative adversarial text to image synthesis". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, pp. 1060–1069. URL: http://dl.acm.org/citation.cfm?id=3045390.3045503.

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

Rodríguez, Natalia Díaz et al. (2014). "A survey on ontologies for human behavior recognition". In: *ACM Computing Surveys*, p. 43.

Ruck, Dennis W, Steven K Rogers, and Matthew Kabrisky (1990). "Feature selection using a multilayer perceptron". In: *Journal of Neural Network Computing* 2, pp. 40–48.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams (1986). "Learning representations by back-propagating errors". In: *Cognitive Modeling* 323, 533—536.

Russakovsky, Olga et al. (2015). "ImageNet large scale visual recognition challenge". In: *International Journal of Computer Vision (IJCV)* 115, pp. 211–252.

Ryoo, M. S. and J. K. Aggarwal (2007). "Hierarchical recognition of human activities interacting with objects". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. DOI: 10.1109/CVPR.2007.383487.

Ryoo, Michael S and Jake K Aggarwal (2008). "Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8.

Sabour, Sara, Nicholas Frosst, and Geoffrey E Hinton (2017). "Dynamic routing between capsules". In: *Advances in Neural Information Processing Systems (NIPS)*, pp. 3859–3869.

Salakhutdinov, Ruslan and Geoffrey E Hinton (2009). "Deep Boltzmann machines". In: *Artificial Intelligence and Statistics Conference (AISTATS)*, pp. 448–455.

Sargano, Allah Bux et al. (2017). "Human action recognition using transfer learning with deep representations". In: *Neural Networks (IJCNN), 2017 International Joint Conference on*, pp. 463–469.

Savitzky, Abraham and Marcel JE Golay (1964). "Smoothing and differentiation of data by simplified least squares procedures." In: *Analytical Chemistry* 36.8, pp. 1627–1639.

Schuldt, C., I. Laptev, and B. Caputo (2004). "Recognizing human actions: a local SVM approach". In: *IEEE International Conference on Pattern Recognition (ICPR)*. Vol. 3, pp. 32–36. DOI: 10.1109/ICPR.2004.1334462.

Schuster, Mike and Kuldip K. Paliwal (1997). "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45, pp. 2673–2681.

Sermanet, P. and Y. LeCun (2011). "Traffic sign recognition with multi-scale convolutional networks". In: *International Joint Conference on Neural Networks (IJCNN)*, pp. 2809–2813.

Sermanet, Pierre et al. (2013). "Pedestrian detection with unsupervised multi-stage feature learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3626–3633.

Shahroudy, Amir et al. (2016). "NTU RGB+D: A large scale dataset for 3D human activity analysis". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1010–1019.

Shahroudy, Amir et al. (2017). "Deep multimodal feature analysis for action recognition in RGB+ D videos". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40.5, pp. 1045–1058.

Shao, Jing et al. (2015). "Deeply learned attributes for crowded scene understanding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4657–4666.

Sharma, Shikhar, Ryan Kiros, and Ruslan Salakhutdinov (2015). "Action recognition using visual attention". In: *arXiv preprint arXiv:1511.04119*.

Shi, Yemin et al. (2017). "Learning long-term dependencies for action recognition with a biologically-inspired deep network". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 716–725.

Shi, Zhiyuan and Tae-Kyun Kim (2017). "Learning and refining of privileged information-based RNNs for action recognition from depth sequences". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4684–4693.

Shotton, Jamie et al. (2011). "Real-time human pose recognition in parts from single depth images". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 3.

Shuang, L., S. Xiao, and W. Yichen (2018). "Compositional human pose regression". In: *Computer Vision and Image Understanding* 176-177, pp. 1 –8.

Si, Chenyang et al. (2019). "An attention enhanced graph convolutional LSTM network for skeleton-based action recognition". In: *arXiv preprint arXiv:1902.09130*.

Sigala, Rodrigo et al. (2005). "Learning features of intermediate complexity for the recognition of biological motion". In: *International Conference on Artificial Neural Networks (ICANN)*, pp. 241–246.

Sigurdsson, Gunnar A et al. (2016). "Hollywood in homes: Crowdsourcing data collection for activity understanding". In: *European Conference on Computer Vision (ECCV)*. Springer, pp. 510–526.

Simonyan, Karen and Andrew Zisserman (2014a). "Two-stream convolutional networks for action recognition in videos". In: *Advances in Neural Information Processing Systems (NIPS)*.

— (2014b). "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556*.

Singh, Amarjot, Devendra Patil, and SN Omkar (2018). "Eye in the sky: Real-time drone surveillance system (DSS) for violent individuals identification using scatterNet hybrid deep learning network". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1629–1637.

Singh, Bharat et al. (2016). "A multi-stream bi-directional recurrent neural network for fine-grained action detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1961–1970.

Singh, Sanchit, Sergio A Velastin, and Hossein Ragheb (2010). "Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods". In: *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, pp. 48–55.

Singh, Suriya, Chetan Arora, and CV Jawahar (2016). "First person action recognition using deep learned descriptors". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2620–2628.

Sminchisescu, C. (2006). "3D human motion analysis in monocular video techniques and challenges". In: *IEEE International Conference on Video and Signal Based Surveillance (ICVSBS)*, pp. 76–76.

Song, Sijie et al. (2017). "An end-to-end spatio-temporal attention model for human action recognition from skeleton data". In: *Thirty-first AAAI conference on Artificial Intelligence (AAAI)*, pp. 4263–4270.

Sonwalkar, Poonam et al. (2015). "Hand gesture recognition for real time human machine interaction system". In: *International Journal of Engineering Trends and Technology (IJETT)* 19.5.

Soomro, Khurram, Amir Roshan Zamir, and Mubarak Shah (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *arXiv preprint arXiv:1212.0402*.

Srivastava, Nitish, Elman Mansimov, and Ruslan Salakhudinov (2015). "Unsupervised learning of video representations using LSTMs". In: *International Conference on Machine Learning (ICML)*, pp. 843–852.

Srivastava, Nitish et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958.

Subetha, T. and S. Chitrakala (2016). "A survey on human activity recognition from videos". In: *International Conference on Information Communication and Embedded Systems (ICICES)*, pp. 1–7. DOI: 10.1109/ICICES.2016.7518920.

Sun, Lin et al. (2015). "Human action recognition using factorized spatio-temporal convolutional networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 4597–4605.

Sun, Lin et al. (2017). "Lattice long short-term memory for human action recognition". In: pp. 2147–2156.

Sung, Jaeyong et al. (2011). "Human activity detection from RGB-D images". In: *Plan, Activity, and Intent Recognition* 64.

Szegedy, C. et al. (2015a). "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

Szegedy, Christian et al. (2015b). "Going deeper with convolutions". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9.

Szegedy, Christian et al. (2016). "Rethinking the Inception architecture for computer vision". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826.

Szegedy, Christian et al. (2017). "Inception-v4, Inception-ResNet and the impact of residual connections on learning". In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. AAAI'17. San Francisco, California, USA: AAAI Press, pp. 4278–4284. URL: http://dl.acm.org/citation.cfm?id=3298023.3298188.

Tanfous, Amor Ben, Hassen Drira, and Boulbaba Ben Amor (2018). "Coding Kendall's shape trajectories for 3D action recognition". In: *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 2840–2849.

Tang, Huixuan (2008). "A comparative evaluation of deep belief nets in semi-supervised learning". In: Department of Computer Science University of Toronto.

Tas, Yusuf and Piotr Koniusz (2018). "CNN-based action recognition and supervised domain adaptation on 3D body skeletons via kernel feature maps". In: *British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*. BMVA Press, p. 158. URL: http://bmvc2018.org/contents/papers/0753.pdf.

Taylor, Graham W., Geoffrey E Hinton, and Sam T. Roweis (2007). "Modeling human motion using binary latent variables". In: *Advances in Neural Information Processing Systems 19*. Ed. by B. Schölkopf, J. C. Platt, and T. Hoffman. MIT Press, pp. 1345–1352.

Tekin, B. et al. (2016). "Direct prediction of 3D body poses from motion compensated sequences". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 991–1000.

Telgarsky, Matus (2016). "Benefits of depth in neural networks". In: *arXiv preprint arXiv:1602.04485*.

The Local (2015). *SNCF increases fines for ticket dodgers*. https://bit.ly/2mYaJwW. Published 20 February 2015. Accessed 10 July 2018.

Theodorakopoulos, Ilias et al. (2014). "Pose-based human action recognition via sparse representation in dissimilarity space". In: *Journal of Visual Communication and Image Representation* 25.1, pp. 12–23.

Tian, Yicong, S. Rahul, and Mubarak Shah (2013). "Spatiotemporal deformable part models for action detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2642–2649.

Tompson, Jonathan et al. (2014). "Real-time continuous pose recovery of human hands using convolutional networks". In: *ACM Transactions on Graphics (TOG)* 33, p. 169.

Tran, An and Loong-Fah Cheong (2017). "Two-stream flow-guided convolutional attention networks for action recognition". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 3110–3119.

Tran, Du et al. (2015). "Learning spatiotemporal features with 3D convolutional networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 4489–4497.

Turaga, P. et al. (2008). "Machine recognition of human activities: A survey". In: *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)* 18, pp. 1473–1488.

Ullah, Ihsan and Alfredo Petrosino (2015). "A strict pyramidal deep neural network for action recognition". In: *International Conference on Image Analysis and Processing (ICIP)*, pp. 236–245.

— (2016). "Spatiotemporal features learning with 3DPyraNet". In: *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 638–647.

Valera, M. and S. A. Velastin (2005). "Intelligent distributed surveillance systems: a review". In: *IEE Proceedings - Vision, Image and Signal Processing* 152.2, pp. 192–204. ISSN: 1350-245X. DOI: 10.1049/ip-vis:20041147.

Valstar, Michel F et al. (2011). "The first facial expression recognition and analysis challenge". In: *IEEE International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 921–926.

Varol, G., I. Laptev, and C. Schmid (2018). "Long-term temporal convolutions for action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6, pp. 1510–1517. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2712608.

Vedaldi, Andrea and Karel Lenc (2015). "Matconvnet: Convolutional neural networks for matlab". In: *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, pp. 689–692.

Veeriah, Vivek, Naifan Zhuang, and Guo-Jun Qi (2015). "Differential recurrent neural networks for action recognition". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 4041–4049.

Vemulapalli, Raviteja, Felipe Arrate, and Rama Chellappa (2014). "Human action recognition by representing 3D skeletons as points in a lie group". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 588–595.

Vieira, A. et al. (2014). "On the improvement of human action recognition from depth map sequences using Space-Time Occupancy Patterns". In: *Pattern Recognition Letters (PRL)* 36, pp. 221–227.

Vieira, Antonio W et al. (2012). "Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences". In: *Iberoamerican Congress on Pattern Recognition (ICPR)*, pp. 252–259.

Vincent, Pascal et al. (2008). "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th International Conference on Machine Learning (ICML)*. ICML'08. Helsinki, Finland: ACM, pp. 1096–1103. ISBN: 978-1-60558-205-4. DOI: 10.1145/1390156.1390294. URL: http://doi.acm.org/10.1145/1390156.1390294.

Vondrick, Carl, Hamed Pirsiavash, and Antonio Torralba (2016). "Generating videos with scene dynamics". In: *Advances In Neural Information Processing Systems (NIPS)*, pp. 613–621.

Vrigkas, Michalis, Christophoros Nikou, and Ioannis A Kakadiaris (2015). "A review of human activity recognition methods". In: *Frontiers in Robotics and AI* 2, p. 28.

Wang, H. et al. (2011). "Action recognition by dense trajectories". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176.

Wang, Heng and Cordelia Schmid (2013). "Action recognition with improved trajectories". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 3551–3558.

Wang, Hongsong and Liang Wang (2017). "Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 499–508.

Wang, J. et al. (2012). "Mining actionlet ensemble for action recognition with depth cameras". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1297.

Wang, J. et al. (2014). "Cross-view action modeling, learning, and recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2649–2656. DOI: 10.1109/CVPR.2014.339.

Wang, Jiang, Zicheng Liu, and Ying Wu (2014). *Human action recognition with depth cameras*. Springer Briefs in Computer Science.

Wang, Keze et al. (2014). "3D human activity recognition with reconfigurable convolutional neural networks". In: *Proceedings of the ACM International Conference on Multimedia (ACM Multimedia)*, pp. 97–106.

Wang, Liang, Weiming Hu, and Tieniu Tan (2003). "Recent developments in human motion analysis". In: *Pattern Recognition* 36, pp. 585–601.

Wang, Lijun et al. (2015a). "Visual tracking with fully convolutional networks". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 3119–3127.

Wang, Limin, Yu Qiao, and Xiaoou Tang (2015). "Action recognition with trajectory-pooled deep-convolutional descriptors". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4305–4314.

Wang, Limin et al. (2015b). "CUHK&SIAT submission for THUMOS'15 action recognition challenge". In: *THUMOS'15 Action Recognition Challenge*, pp. 1–3.

Wang, Limin et al. (2015c). "Towards good practices for very deep two-stream convnets". In: *arXiv preprint arXiv:1507.02159*.

Wang, Limin et al. (2016a). "Temporal segment networks: towards good practices for deep action recognition". In: *European Conference on Computer Vision (ECCV)*, pp. 20–36.

Wang, Limin et al. (2017a). "Untrimmednets for weakly supervised action recognition and detection". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4325–4334.

Wang, M., B. Ni, and X. Yang (2017). "Recurrent modeling of interaction context for collective activity recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3048–3056.

Wang, P. et al. (2016b). "Graph based skeleton motion representation and similarity measurement for action recognition". In: *European Conference on Computer Vision (ECCV)*, pp. 370–385.

Wang, Pichao et al. (2015d). "ConvNets-based action recognition from depth maps through virtual cameras and pseudocoloring". In: *Proceedings of the ACM International Conference on Multimedia (ACM)*, pp. 1119–1122.

Wang, Pichao et al. (2015e). "Deep convolutional neural networks for action recognition using depth map sequences". In: *arXiv preprint arXiv:1501.04686*.

Wang, Pichao et al. (2016c). "Action recognition based on joint trajectory maps using convolutional neural networks". In: *ACM Multimedia*, pp. 102–106.

Wang, Pichao et al. (2017b). "Scene flow to action map: A new representation for RGB-D based action recognition with convolutional neural networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 595–604.

Wang, Xiaolong, Ali Farhadi, and Abhinav Gupta (2016). "Actions transformations". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2658–2667.

Wang, Yifan et al. (2016d). "Two-stream SR-CNNs for action recognition in videos". In: vol. 108. British Machine Vision Conference (BMVC), pp. 1–12.

Wang, Yilin et al. (2016e). "Hierarchical attention network for action recognition in videos". In: *arXiv preprint arXiv:1607.06416*.

Wanqing, Li, Zhang Zhengyou, and Liu Zicheng (2010). "Action recognition based on a bag of 3D points". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9–14.

Wei Niu et al. (2004). "Human activity detection and recognition for video surveillance". In: *IEEE International Conference on Multimedia and Expo (ICME)*. Vol. 1, pp. 719–722. DOI: 10.1109/ICME.2004.1394293.

Weinland, Daniel, Remi Ronfard, and Edmond Boyer (2006). "Free viewpoint action recognition using motion history volumes". In: *Computer Vision and Image Understanding* 104, pp. 249–257.

Weinland, Daniel, Rémi Ronfard, and Edmond Boyer (2011). "A survey of vision-based methods for action representation, segmentation and recognition". In: *Computer Vision and Image Understanding* 115, pp. 224–241.

Weiyao Lin et al. (2008). "Human activity recognition for video surveillance". In: *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 2737–2740. DOI: 10.1109/ISCAS.2008.4542023.

Weng, J., C. Weng, and J. Yuan (2017). "Spatio-temporal naive-bayes nearest-neighbor (ST-NBNN) for skeleton-based action recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4171–4180.

Weng, J. et al. (2018). "Discriminative spatio-temporal pattern discovery for 3D action recognition". In: *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1.

Wolf, Christian et al. (2014). "Evaluation of video activity localizations integrating quality and quantity measurements". In: *Computer Vision and Image Understanding* 127, pp. 14–30.

Wu, Daoxi et al. (2014). "An adaptive stacked denoising auto-encoder architecture for human action recognition". In: *Applied Mechanics & Materials* 631, pp. 403–409.

Wu, Jialin et al. (2016). "Action recognition with joint attention on multi-level deep features". In: *arXiv preprint arXiv:1607.02556*.

Xia, L., C. Chen, and JK Aggarwal (2012a). "View invariant human action recognition using histograms of 3D joints". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20–27.

Xia, Lu, Chia-Chih Chen, and Jake K. Aggarwal (2012b). "View invariant human action recognition using histograms of 3D joints". In: *IEEE Conference on Computer Vision and Pattern Recognitio (CVPR)*, pp. 20–27.

Xie, Chunyu et al. (2018). "Memory attention networks for skeleton-based action recognition". In: *arXiv preprint arXiv:1804.08254*.

Xie, Lidong et al. (2014). "A pyramidal deep learning architecture for human action recognition". In: *International Journal of Modelling, Identification and Control* 21, pp. 139–146.

Xingyi, Z. et al. (2016). "Deep kinematic pose regression". In: *European Conference on Computer Vision (ECCV)*, pp. 186–201.

Xiong, Yuanjun et al. (2015). "Recognize complex events from static images by fusing deep channels". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1600–1609.

Xiong, Yuanjun et al. (2016). "CUHK & ETHZ & SIAT submission to ActivityNet challenge 2016". In: *arXiv preprint arXiv:1608.00797*.

Xu, Haining et al. (2015a). "Spatio-temporal pyramid model based on depth maps for action recognition". In: *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6. DOI: 10.1109/MMSP.2015.7340806.

Xu, Kelvin et al. (2015b). "Show, attend and tell: Neural image caption generation with visual attention". In: *International Conference on Machine Learning (ICML)*, pp. 2048–2057.

Xu, Tiantian et al. (2016). "Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition". In: *Image and Vision Computing* 55, pp. 127–137.

Yang, J., K. Yu, and T. Huang (2010). "Supervised translation-invariant sparse coding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3517–3524.

Yang, Jianchao et al. (2009). "Linear spatial pyramid matching using sparse coding for image classification". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 6.

Yang, Xiaodong and YingLi Tian (2014). "Super normal vector for activity recognition using depth sequences". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 804–811.

Yang, Yang, Imran Saleemi, and Mubarak Shah (2013). "Discovering motion primitives for unsupervised grouping and one-shot learning of human actions, gestures, and expressions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 35, pp. 1635–1648.

Yao, B. and L. Fei-Fei (2010). "Modeling mutual context of object and human pose in human-object interaction activities". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17–24.

Ye, Mao and Ruigang Yang (2014). "Real-time simultaneous pose and shape estimation for articulated objects using a single depth camera". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2345–2352.

Yeung, Serena et al. (2016). "End-to-end learning of action detection from frame glimpses in videos". In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2678–2687.

Yu, Kai, Yuanqing Lin, and John Lafferty (2011). "Learning image representations from the pixel level via hierarchical sparse coding". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1713–1720.

Yun, K. et al. (2012a). "Two-person interaction detection using body-pose features and multiple instance learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28–35. DOI: 10.1109/CVPRW.2012.6239234.

— (2012b). "Two-person interaction detection using body-pose features and multiple instance learning". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28–35.

Yurii, N. (1983). "A method for solving a convex programming problem with convergence rate $O(1/K2)$". In: *Soviet Mathematics Doklady*, pp. 372–367.

Zang, Jinliang et al. (2018). "Attention-based temporal weighted convolutional neural network for action recognition". In: *International Conference on Artificial Intelligence Applications and Innovations (IFIP)*, pp. 97–108.

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *European Conference on Computer Vision (ECCV)*, pp. 818–833.

Zhang, H. et al. (2014). "Real-time action recognition based on a modified deep belief network model". In: *IEEE International Conference on Information and Automation (ICIA)*, pp. 225–228.

Zhang, Jing et al. (2016). "RGB-D-based action recognition datasets: A survey". In: *Pattern Recognition* 60, pp. 86–105.

Zhang, P. et al. (2019). "View adaptive neural networks for high performance skeleton-based human action recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 1, pp. 1–1.

Zhang, Songyang, Xiaoming Liu, and Jun Xiao (2017). "On geometric features for skeleton-based action recognition using multilayer lstm networks". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 148–157.

Zhang, Zhengyou (2012). "Microsoft Kinect sensor and its effect". In: *IEEE Multimedia* 19, pp. 4–10.

Zhao, R., H. Ali, and P. van der Smagt (2017). "Two-stream RNN/CNN for action recognition in 3D videos". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4260–4267. DOI: 10.1109/IROS.2017.8206288.

Zhou, X. et al. (2016). "Sparseness meets deepness: 3D human pose estimation from monocular video". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4966–4975.

Zhu, Guangming et al. (2016a). "An online continuous human action recognition algorithm based on the Kinect sensor". In: *Sensors* 16.2, p. 161.

Zhu, H., R. Vial, and S. Lu (2017). "TORNADO: A spatio-temporal convolutional regression network for video action proposal". In: *IEEE International Conference on Computer Vision (ICCV)*, pp. 5813–5821.

Zhu, Wentao et al. (2016b). "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, pp. 3697–3703. URL: http://dl.acm.org/citation.cfm?id=3016387.3016423.

— (2016c). "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, pp. 3697–3703. URL: http://dl.acm.org/citation.cfm?id=3016387.3016423.

Zhu, Yan et al. (2010). "Sparse coding on local spatial-temporal volumes for human action recognition". In: *Asian Conference on Computer Vision (ACCV)*, pp. 660–671.

Zouba, Nadia et al. (2009). "Assessing computer systems for monitoring elderly people living at home". In: *Proceedings of the World Congress of Gerontology and Geriatrics (IAGG)*, pp. 5–9.