

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΗΣ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Τεχνικές Εκπαίδευσης Βαθιών Νευρωνικών Δικτύων για την Αναγνώριση Ανθρώπινης Συμπεριφοράς από Οπτικά Δεδομένα

Ειρήνη Α. Μαθέ

ΚΕΡΚΥΡΑ

Μάιος 2024



IONIAN UNIVERSITY

FACULTY OF INFORMATION SCIENCE & INFORMATICS
DEPARTMENT OF INFORMATICS

PhD THESIS

Neural Network Training Techniques for Human Activity Recognition Using Visual Data

Eirini A. Mathe

CORFU

May 2024

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Τεχνικές Εκπαίδευσης Βαθιών Νευρωνικών Δικτύων για την Αναγνώριση Ανθρώπινης Συμπεριφοράς από Οπτικά Δεδομένα

Ειρήνη Α. Μαθέ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:

Φοίβος-Απόστολος Μυλωνάς, Αναπληρωτής Καθηγητής Πανεπιστήμιο Δυτικής Αττικής

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Φοίβος-Απόστολος Μυλωνάς, Αναπληρωτής Καθηγητής, Πανεπιστήμιο Δυτικής Αττικής

Σταύρος Περαντώνης, Ερευνητής Α', ΕΚΕΦΕ–"Δημόκριτος"

Κάτια-Λήδα Κερμανίδου, Αναπληρώτρια Καθηγήτρια, Ιόνιο Πανεπιστήμιο

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Φοίβος-Απόστολος Μυλωνάς
Αναπληρωτής Καθηγητής
Πανεπιστήμιο Δυτικής Αττικής

Σταύρος Περαντώνης
Ερευνητής Α'
ΕΚΕΦΕ–"Δημόκριτος"

Κάτια-Λήδα Κερμανίδου
Αναπληρώτρια Καθηγήτρια
Ιόνιο Πανεπιστήμιο

Θεμιστοκλής Έξαρχος
Αναπληρωτής Καθηγητής
Ιόνιο Πανεπιστήμιο

Παναγιώτης Κουρουθανάσης
Καθηγητής
Ιόνιο Πανεπιστήμιο

Στέργιος Παλαμάς
Επίκουρος Καθηγητής
Ιόνιο Πανεπιστήμιο

Σταύρος Καρκάνης
Καθηγητής
Πανεπιστήμιο Θεσσαλίας

Ημερομηνία Εξέτασης: Μάιος 2024

PhD THESIS

Neural Network Training Techniques for Human Activity Recognition Using Visual Data

Eirini A. Mathe

SUPERVISOR:

Phivos-Apostolos Mylonas, Associate Professor, University Of West Attica

THREE-MEMBER ADVISORY COMMITTEE:

Phivos-Apostolos Mylonas, Associate Professor, University Of West Attica

Stavros Perantonis, Research Director, NCSR–"Demokritos"

Katia-Lida Kermanidis, Associate Professor, Ionian University

SEVEN-MEMBER EXAMINATION COMMITTEE:

Phivos-Apostolos Mylonas
Associate Professor
University Of West Attica

Stavros Perantonis
Research Director
NCSR–"Demokritos"

Katia-Lida Kermanidis
Associate Professor
Ionian University

Themistoklis Exarchos
Associate Professor
Ionian University

Panagiotis Kourouthanassis
Professor
Ionian University

Stergios Palamas,
Assistant Professor
Ionian University

Stavros Karkanis
Professor
University of Thessaly

Examination Date: May 2024

ΠΕΡΙΛΗΨΗ

Ο στόχος αυτής της Διατριβής είναι να εξετάσει προσεγγίσεις και στρατηγικές για την εκπαίδευση αρχιτεκτονικών βαθιών δικτύων με σκοπό την Αναγνώριση Ανθρώπινης Δραστηριότητας (Human Activity Recognition – HAR). Στην προσπάθεια να βελτιωθούν τα συστήματα αυτά, η Διατριβή ξεκινά με μια εκτενή μελέτη διαφόρων πρόσφατων ερευνητικών προσπαθειών στον τομέα της αναγνώρισης δραστηριοτήτων. Οι ερευνητικές προσπάθειες αρχίζουν με την εισαγωγή μιας νέας προσέγγισης για την διδιάστατη αναπαράσταση της τρισδιάστατης κίνησης των σκελετικών αρθρώσεων, βασισμένη σε γνωστούς 2Δ φασματικούς μετασχηματισμούς εικόνας (DFT, FFT, DCT, DST), αποδεικνύοντας την αποτελεσματικότητά της στην αναγνώριση. Ωστόσο, σε περιπτώσεις όπου το υποκείμενο μπορεί να παρατηρηθεί υπό διαφορετικές οπτικές γωνίες σε σχέση με αυτές που έχει εκπαιδευθεί το μοντέλο, παρατηρείται μια σημαντική πτώση της απόδοσης, οδηγώντας την έρευνα προς τεχνικές και στρατηγικές που θα μπορούσαν να προσφέρουν προοπτικές βελτίωσης.

Επομένως, αναγνωρίζοντας αυτήν την πρόκληση, η έρευνα στράφηκε στο να σχεδιαστεί μια προηγμένη τεχνική επαύξησης δεδομένων, χρησιμοποιώντας τεχνητά δεδομένα που προκύπτουν έπειτα από την εφαρμογή γεωμετρικής περιστροφής στα πραγματικά δείγματα και επιβάλλοντας, επίσης, ένα βήμα ευθυγράμμισης της οπτικής γωνίας παρατήρησης. Με αυτόν τον τρόπο αποδείχθηκε πειραματικά ότι η προτεινόμενη μεθοδολογία μπορεί να εμφανίσει ισχυρή απόδοση και σε συνθήκες διαφορετικής οπτικής γωνίας παρατήρησης, η οποία κατέστη πλεόν συγκρίσιμη με τις τιμές άλλων συνθηκών αξιολόγησης. Μέσω διεξοδικών πειραμάτων, αυτή η μείωση της απόδοσης αντισταθμίστηκε, επιδεικνύοντας την προσαρμοστικότητα της προτεινόμενης προσέγγισης σε διαφορετικές οπτικές γωνίες και βοηθώντας την εφαρμογή της σε πραγματικές συνθήκες.

Το επόμενο βήμα της έρευνας ήταν η εξέταση τεχνικών προσαρμογής πεδίου, όπου η ανταγωνιστική εκπαίδευση χρησιμοποιήθηκε για να γεφυρώσει τα κενά απόδοσης. Ενώ αυτές οι προσπάθειες έφεραν ελπιδοφόρα αποτελέσματα υπό συγκεκριμένες συνθήκες, αναγνωρίστηκε επίσης ότι αυξημένη πολυπλοκότητα συνδέεται με αυτήν την προσέγγιση, καθιστώντας την ανεφάρμοστη για εφαρμογές στην πραγματική ζωή. Έτσι, το επόμενο βήμα ήταν να εξεταστούν προσεγγίσεις σύντηξης, που συνδύαζαν διάφορες αναπαραστάσεις κίνησης, δηλαδή RGB και σκελετικά δεδομένα. Αυτές οι τεχνικές σύντηξης όχι μόνο υπερέβησαν την απόδοση των πρόσφατων ερευνητικών μεθόδων, αλλά και υπογράμμισαν την ευελιξία και την ανθεκτικότητα της προτεινόμενης μεθοδολογίας στην αναγνώριση ανθρώπινης δραστηριότητας σε διάφορα σενάρια.

Τέλος, αυτή η Διατριβή ολοκληρώνεται με μια νέα μέθοδο επαύξησης δεδομένων με βάση την παρεμπόδιση της ανθρώπινης κίνησης. Αυτή η προσέγγιση όχι μόνο υπερέβη την αποτελεσματικότητα της αρχικής μεθοδολογίας, αλλά συνέβαλε σημαντικά και στη συνολική βελτίωση της απόδοσης στο πρόβλημα της αναγνώρισης. Αυτή η τεχνική επαύξησης επιβεβαίωσε επιπλέον την πρακτική εφαρμοσιμότητα των ερευνητικών ευρημάτων. Συνο-

πτικά, η πορεία αυτής της Διατριβής μέσω πειραματισμών στις περιοχές της προσαρμογής πεδίου, της σύντηξης και της επαύξησης δεδομένων προτείνει μια ολιστική προσέγγιση στην προαγωγή των συστημάτων αναγνώρισης, ανοίγοντας το δρόμο για πιο ανθεκτικές και ευέλικτες τεχνολογίες για την αναγνώριση ανθρώπινων δραστηριοτήτων σε διάφορα περιβάλλοντα.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αναγνώριση Ανθρώπινης Συμπεριφοράς

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Αναγνώριση Ανθρώπινης Συμπεριφοράς, Βαθιά Μάθηση, Προσαρμογή Πεδίου, Επαύξηση Δεδομένων, Πολυτροπική Σύντηξη

ABSTRACT

The goal of this Thesis is to investigate approaches and training strategies for training deep networks architectures with the goal of Human Activity Recognition (HAR). In the pursuit to enhance (HAR) systems, this Thesis begins with a comprehensive exploration of various recent research efforts on the field of HAR. The research efforts begin with the introduction of a novel approach for 2D representation of 3D motion of skeletal joints, based on well-known 2D spectral image transformations (DFT, FFT, DCT, DST), demonstrating its efficacy in HAR tasks. However, as this approach transitioned to cross-view setups, a notable decline in performance emerged, driving research towards a thorough examination, aiming to uncover nuanced details that could shed light on potential avenues for improvement.

Therefore, recognizing this challenge, research turned into devising an advanced data augmentation approach using artificial data which resulted upon applying geometric rotation transformation to real samples and also imposed a view alignment step. This way it was demonstrated that cross-view scenarios may also exhibit strong performance, comparable to other evaluation setups. Through meticulous experimentation, this performance drop was mitigated, showcasing the adaptability of the proposed approach across different viewing perspectives and reinforcing its relevance in real-world scenarios.

The next step of research was the investigation of domain adaptation techniques, where adversarial training was leveraged to bridge performance gaps. While these endeavors yielded promising results under specific circumstances, it was also acknowledged that heightened complexity was associated with this approach, rendering it impractical for real-life applications. Therefore, the next step was to explore fusion approaches that combined various motion representations, i.e., RGB and skeletal data. These fusion techniques not only surpassed the performance of state-of-the-art methods but also underscored the versatility and robustness of the proposed methodology in capturing nuanced human actions across diverse contexts.

Finally, this Thesis concludes with a novel occlusion-based data augmentation method. This approach not only surpassed the efficacy of the initial work but also contributed significantly to overall HAR performance enhancement. By addressing occlusion challenges inherent in real-world scenarios, this augmentation technique further validated the practical applicability of the research findings. In essence, the journey of this Thesis through experimentation, adaptation, fusion, and augmentation underscores a holistic approach to advancing HAR systems, paving the way for more robust and versatile human activity recognition technologies in diverse environments.

SUBJECT AREA: Human Activity Recognition

KEYWORDS: Human Activity Recognition, Deep Learning, Domain Adaptation, Data Augmentation, Multimodal Fusion

ACKNOWLEDGEMENTS

I stand at the culmination of this incredible journey, and it is with deep gratitude that I acknowledge those who have made this feat possible. First and foremost, my sincerest thanks go to my advisor, Phivos Mylonas, whose wisdom and steadfast support have been my guiding lights. Phivos, your patience and belief in my abilities have transformed this challenging process into a rewarding experience.

To my colleagues Dimitrios Koutrintzes, Alexandros Mitsou, Antonios Papadakis , George Pikramenos, Evaggelos Spyrou, Ioannis Vernikos and others —you have not only been co-researchers but great friends. The camaraderie and support we shared made all the late nights and seemingly endless experiments bearable, and even enjoyable.

A special thanks to my family for their unwavering love and encouragement. You have been my emotional backbone, continually inspiring me to persevere even when the end seemed elusive.

Lastly, I extend my gratitude to all those who provided support in any form, be it as brief words of encouragement or long sessions of discussion and debate. Each of you has contributed to this work in your own unique way, and for that, I am eternally grateful.

CONTENTS

1 Human Activity Recognition (HAR) 29

1.1 Introduction 29

1.2 Basic terminology of motion-based HAR 31

1.3 Datasets 33

1.3.1 Segmented Datasets 33

1.3.2 Continuous/Online Datasets 42

1.4 Recent Approaches on HAR 43

1.4.1 RGB-based Motion Recognition using Deep Networks 43

1.4.1.1 Segmented RGB-based Motion Recognition 44

1.4.1.2 Continuous RGB-based Motion Recognition 46

1.4.2 Depth-based Motion Recognition with Deep Learning 47

1.4.2.1 Segmented Depth-Based Motion Recognition 48

1.4.2.2 Continuous Depth-Based Motion Recognition 48

1.4.3 Skeleton-based Motion Recognition with Deep Learning 49

1.4.3.1 Segmented Skeleton-Based Motion Recognition 49

1.4.3.2 Continuous Skeleton-Based Motion Recognition 50

1.4.4 RGB+D-based Motion Recognition with Deep Learning 50

1.4.4.1 Segmented RGB+D-Based Motion Recognition 51

1.4.4.2 Continuous RGB+D-Based Motion Recognition 51

1.5 Research Questions 52

1.6 Contributions of this Thesis 52

2 Theoretical and Experimental Background 55

2.1 Artificial Intelligence 55

2.1.1 Machine Learning 59

2.1.2 Neural Networks 60

2.1.3 Deep Learning 62

2.1.4 Convolutional Neural Networks 64

2.2 Neural Network Training Techniques . . . 66
2.2.1 Data augmentation . . . 67
2.2.2 Domain Adaptation . . . 69
2.2.3 Adversarial Discriminative Domain Adaptation (ADDA) . . . 71
2.3 Acquisition of Human Motion Data . . . 73
2.3.1 The Microsoft Kinect SDK . . . 73
2.4 OpenPose and PoseNet Libraries . . . 80
2.4.1 Datasets . . . 81
2.4.1.1 PKU-MMD Data Set . . . 82
2.4.1.2 SYSU 3D Human-Object Interaction (HOI) Set . . . 83
2.4.1.3 UTKinect-Action3D Dataset . . . 86
2.4.2 Typical HAR Evaluation Protocols . . . 87
2.4.3 Evaluation Metrics . . . 88
2.4.4 Implementation and Libraries . . . 91

3 Human Activity Recognition using CNNs 93
3.1 Introduction . . . 93
3.1.1 Initial experiment . . . 94
3.1.1.1 Skeletal Data Extraction . . . 95
3.1.1.2 Activity Images . . . 97
3.1.1.3 Gesture Classification . . . 98
3.1.1.4 Experiments . . . 98
3.1.1.5 Results of the Initial Experiments . . . 99
3.1.1.6 Preliminary Conclusions . . . 100
3.2 Related Work on HAR based on 2D Representations of 3D Skeletal Motion 101
3.2.1 Visual Skeletal Representations . . . 101
3.3 Methodology . . . 103
3.3.1 Signal Images . . . 104
3.3.2 Activity Images . . . 105
3.3.3 Network Architecture and Training . . . 106
3.4 Experimental Protocol and Results . . . 107
3.5 Conclusions . . . 109

4 Geometric Data Augmentation for HAR **111**
4.1 Introduction . . . 111
4.2 Related Work . . . 112
4.3 Methodology . . . 112
4.3.1 Geometric Transformations on 3D skeletons . . . 112
4.3.2 Skeleton Alignment and Data Augmentation . . . 114
4.3.3 CNN Architecture . . . 114
4.4 Experimental Protocol and Results . . . 116
4.5 Conclusions . . . 119

5 Domain Adaptation for HAR **121**
5.1 Introduction . . . 121
5.2 Related Work . . . 122
5.2.1 Domain Adaptation . . . 122
5.2.2 Transfer Learning for HAR . . . 123
5.3 Methodology . . . 124
5.4 Experimental Protocol and Results . . . 127
5.5 Conclusions . . . 130

6 Data Fusion Approaches for HAR **133**
6.1 Introduction . . . 133
6.2 Related Work . . . 134
6.2.1 Hand-crafted Features . . . 134
6.2.2 Multimodal Fusion Methods . . . 134
6.2.3 Fusion of handcrafted and deep features . . . 136
6.3 Methodology . . . 136
6.3.1 Data Modalities . . . 136
6.3.1.1 Pseudo-colored Image Representation of Skeletal Data . . . 136
6.3.2 Dynamic Images for Activity Representation . . . 138
6.3.3 Handcrafted Features . . . 138
6.3.4 Fusion and Classification . . . 139
6.4 Experimental Protocol and Results . . . 142

6.4.1 Experimental Setup and Implementation Details 144
6.4.2 Experimental Results 145
6.5 Conclusions 147

7 Data Augmentation using Occlusion for HAR 149
7.1 Introduction 149
7.2 Related Work 150
7.2.1 Occlusion in HAR 150
7.2.2 Data Augmentation for HAR 151
7.3 Methodology 152
7.3.1 Occlusion of Skeletal Data 152
7.3.2 CNN Architecture 153
7.4 Experimental Protocol and Results 155
7.5 Conclusions 157

8 Conclusions and Future Research Directions 159
8.1 Conclusions 159
8.2 Future Research Directions 162

ABBREVIATIONS - ACRONYMS 163

A Short CV 167

B List of Publications 169
B.1 Journals 169
B.2 Conferences 169

REFERENCES 199

LIST OF FIGURES

1.1 A generic HAR pipeline incorporating sensors, model training, evaluation, and interpretation. 31

1.2 Capture process using Mocap technology in case of CMU Mocap dataset. (a) and (b): positions of markers on the subject's body. (c): visualization of the subject's body – round dots correspond to markers, while colored lines to the body structure. 34

1.3 Data collection process in case of Mocap Database HDM05. (a) the markers on the body of the human subject; (b) the camera setup used for capturing; (c) comparison between a C3D file depicting a point cloud (left) and an AMC file depicting a skeleton (right). Figure taken from [190]. 35

1.4 Depth maps from the MSR-Action3D dataset. (a) draw tick; (b) tennis serve. Figure taken from [157]. 36

1.5 Example actions of the MSRC-12 Dataset. Left: "draw tick;" right: "tennis serve." Figure taken from [65]. 36

1.6 Example actions of the MSRC-12 Dataset. Left: "start music/raise volume (of music);" right: "navigate to next menu." Figure taken from [65]. 36

1.7 Sample frames of the DailyActivity3D dataset. Figure taken from [291]. . . 37

1.8 Samples from G3D dataset: (a) color video frame; (b) the corresponding depth map; and (c) the extracted skeleton from the human subject. Figure taken from [21]. 37

1.9 Sample activities of the SBU Kinect Interaction Dataset: (a) "approaching;" (b) "departing." Each row contains a color image, a depth map, and extracted skeletons at the first, 25%, 50%, 75%, and the last frame of the entire sequence for each interaction. Red skeleton corresponds to the active subject, blue skeleton corresponds to inactive subject. Figure taken from [322]. 38

1.10 The experimental setup that has been used for the acquisition of the Berkeley MHAD Database. Figure taken from [194]. 39

1.11 Snapshots from all the actions available in the Berkeley MHAD Database, displayed along with the corresponding point clouds obtained from the Kinect depth data. Depicted actions from left to right: "jumping," "jumping jacks," "bending," "punching," "waving two hands," "waving one hand," "clapping," "throwing," "sit down/stand up," "sit down," "stand up." Figure taken from [194]. 40

1.12 Throwing action from the Berkeley MHAD Database as captured by the reference camera in each camera cluster, as well as from the two Kinect cameras. Figure taken from [194]. 40

1.13 Sample frames of Multiview Action3D dataset and specifically from the cross environment setting. Figure taken from [292]. 41

1.14 Samples from ChaLearn LAP IsoG Dataset: (a) Colorized sample depth maps; (b) body part annotations of head, shoulder, elbow and hand positions, with circles denoting occluded body parts. 41

1.15 Sample frames from activities of the two datasets NTU RGB+D datasets. Figure taken from [236]. 42

1.16 A brief taxonomy of RGB+D based motion recognition approaches. Figure taken from [302]. 44

2.1 A Venn diagram, displaying how deep learning is a subset of machine learning and how machine learning is a subset of artificial intelligence. Figure taken from [222] 55

2.2 Several modern applications of AI, including (a) spam filtering; (b) search engines; (c) visual recognition; (d) stock price prediction; (e) semantic segmentation; (f) face detection; (g) home assistants; (h) autonomous driving. . 58

2.3 The layers of a feed-forward neural network. Figure taken from [232] 61

2.4 A visual example of convolution. Figure taken from [226]. 64

2.6 A visual example of Max Pooling. Figure taken from [180]. 65

2.5 Example filters that have been learned by a CNN and corresponding feature maps. Figure taken from [6]. 65

2.7 A convolutional neural network, comprising convolutional, pooling and fully-connected layers. Figure taken from [238]. 66

2.8 An example of deactivation of neurons using the dropout technique. Figure taken from [95]. 67

2.9 Several data augmentation techniques applied to the problem of image classification. Figure taken from [197] 68

2.10 Typical network structure/architecture of deep neural networks implementing adversarial-based methods for domain adaptation. The generator(s) are optional – used only to create synthetic data if required. Domain Discriminator attempts to discriminate (classify) source and target domains, while task-specific (e.g., classification) is performed by task-specific network. Figure taken from [250] 72

2.11 An overview Adversarial Discriminative Domain Adaptation (ADDA) approach. A source encoder CNN is firstly trained using labeled source image examples. Next, adversarial adaptation is performed by learning a target encoder CNN such that a discriminator that sees encoded source and target examples cannot reliably predict their domain label. During testing, target images are mapped with the target encoder to the shared feature space and classified by the source classifier. Dashed lines indicate fixed network parameters. Figure taken from [278] 73

2.12 The two versions of the Kinect sensor. Left: Kinect v1; Right: Kinect v2. . . 74

2.13 User interaction using the Kinect sensor. 74

2.14 The components of the Kinect sensor include: RGB-D camera, 3D Depth camera, IR emitter and array of microphones. 75

2.15 Left column: RGB video frames extracted of sequences of a person performing some activities; Right column: the corresponding depth frames with overlaid skeleton sequences. 76

2.16 Extracted human skeleton using the Kinect v2. Yellow circles indicate joints, while green lines indicate their parent-child relationships within the human body. 77

2.17 Extracted human skeletons using the Kinect v1 (left) and v2 (right). In both cases joints have been grouped to form body parts, i.e., left arm (green), right arm (red), torso (blue), left leg (orange) and right leg (magenta). Figure taken from [287]. 77

2.18 Left: depth comparisons between extracted features within the body part classification process. Right: randomized decision forests used for classification of image pixels to body parts. Figure taken from [244]. 78

2.19 Synthetic and real poses used for training and evaluation of the Kinect skeleton extraction algorithm. Figure taken from [244]. 78

2.20 A sequence of images demonstrating the transformation from depth images to body parts and then to 3D joint proposals, ultimately leading to the identification of joints. Figure taken from [244]. 78

2.21 The Azure Kinect DK sensor. 79

2.22 Several consumer RGB-D sensors. Top row: Asus Xtion, Creative Senz3d, Stereolab Zed; Bottom row: Intel Realsense, Orbec Astra Pro, Mynt Eye. . 79

2.23 The output from OpenPose involves real-time detection of keypoints for the body, feet, hands, and face. Results contain viewpoint and appearance variation, occlusion, crowding, contact, and other common imaging artifacts. Figure taken from [27]. 81

2.24 Camera setup of the PKU-MMD dataset. Three cameras are faced in front of the subject, capturing her/him under three viewpoints, under angles $\theta = \pm 45^{\circ}$ Cameras $\#1$, $\#2$, $\#3$ will be denoted as L, M, R, respectively. 83

2.25 A brief visual summary of the properties of the PKU-MMD dataset, consisting of 4 modalities, 3 camera views, 60 distinct human subjects and 51 daily and interaction activities. Figure taken from [163]. 84

2.26 A set of 3 consecutive frames from the PKU-MMD, from each of the four modalities, i.e., per row and from top to bottom: RGB, depth, skeleton and IR. Figure taken from [163]. 84

2.27 A set of extracted frames from the PKU-MMD, illustrating a subset of the collected 51 actions for single person activities and person-to-person interaction activities. Figure taken from [163]. 85

2.28 Examples of one sample per class from the SYSU 3D HOI dataset. Each row displays an RGB sample (phone, chair, bag, wallet, mop and besom) at the top and the corresponding depth channel with superimposed skeleton data is illustrated below. Figure taken from [102]. 86

2.29 10 activities of the UTKinect dataset with RGB and depth frames. First row: walk, stand up, sit down, pick up, carry. Second row: throw, push, pull, wave hands, clap hands. Figure taken from [311]. 87

2.30 A toy example of a confusion matrix for the binary classification task cats vs. dogs. 90

3.1 A graphic illustration of the proposed methodology for arm gesture recognition. Specifically, 8 skeleton joints are tracked during a gesture. Then, a signal image is constructed. Upon application of the Discrete Cosine Transform, the activity image is constructed and given as input to the CNN which classifies it to the depicted class, i.e., *clapping*. 95

3.2 The 8 selected skeleton joints that have been captured using Kinect v1 and have been used in the initial experiment. 96

3.3 (a) a sample signal image; and (b) the corresponding activity image from gesture “clapping.” Figure best viewed in color. 97

3.4 A sample signal image and the corresponding activity image from all 10 gestures recognized throughout our experiments. Upper row: signal images; lower row: activity images. Figure best viewed in color. 98

3.5 The CNN architecture that has been used in our initial experiment for arm gesture classification. 99

3.6 Examples of gestures taken from the data that has been created for the arm gesture experiments. For each example skeleton sequences overlaid to RGB data and single are illustrated. From top to bottom: *clapping*, *two hands raise* and *two hands circle*. 100

3.7 (a) A signal image; activity image resulting upon (b) DFT; (c) FFT; (d) DCT; (e); DST. Action is *playing with phone/tablet*. DFT and FFT images have been processed with log transformation for visualization purposes. Figure best viewed in color. 103

3.8 A graphic illustration of the proposed methodology for activity recognition. Specifically, 25 skeleton joints are tracked during a gesture. Then, a signal image is constructed. Upon application of the Discrete Cosine Transform, the activity image is constructed and given as input to the CNN which classifies it to the depicted class, i.e., *falling*. 104

3.9 The proposed CNN architecture for recognition of human activities based on the 2D activity representation presented in this Chapter. 106

3.10 Examples of activity images from 11 classes and for the 4 transforms used. 1st row: DFT; 2nd row: FFT; 3rd row: DCT; 4th row: DST. a) eat meal/snack; b) falling; c) handshaking; d) hugging other person; e) make a phone call/answer phone; f) playing with phone/tablet; g) reading; h) sitting down; i) standing up; j) typing on a keyboard; k) wear jacket. DFT and FFT images have been processed with log transformation for visualization purposes. Figure best viewed in color. 108

4.1 Camera setup of the PKU-MMD dataset. Three cameras are placed in front of the subject, capturing her/him under three viewpoints, under angles $\theta = 0^\circ \pm 45^\circ$ Cameras $\#1$, $\#2$, $\#3$ will be denoted as L, M, R, respectively.(This Figure is the same as Fig. 2.24, and is herein used for convenience.) . . . 113

4.2 A visual overview of the proposed data alignment methodology. 115

4.3 A visual overview of the proposed data augmentation methodology. 115

4.4 skeleton rotated by angle θ: a) $\theta = 90^\circ$, b) $\theta = 45^\circ$, c) $\theta = 0^\circ$ (raw skeleton), d) $\theta = -45^\circ$, e) $\theta = -90^\circ$. For illustrative purposes, depth information, i.e., z-coordinate has been discarded. 116

4.5 A visual overview of the heuristically modified methodology. In this artificial example we consider an imaginary testing sample, 10 possible classes and the 5 alignments presented in subsection 4.4. All classification confidences are first estimated (i.e., for all 10 classes). Then, we consider as classification output the class with the maximum confidence score, among all, i.e., in our example the imaginary testing sample is classified as “class 1” with 0.89 confidence. 118

5.1 Visual illustration of the effects of domain adaptation. On top the adapted model is equal to the trained model. Because source and target data have different distributions, the resulting distributions in latent space are different and the domain discriminator can be trained to classify instances in latent space. After the adaptation procedure is finished, the adapted model has been trained so that the distribution of its output, when its input is distributed as the test data, is equal to the source data distribution in latent space. The domain discriminator can no longer discriminate the domains. 125

5.2 A visual overview of the proposed domain adaptation approach. 127

6.1 Sequences of actors performing the activities *handwaving* (top) and *kicking something* (bottom). Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the PKU-MMD dataset [163] and have been trimmed for illustration purposes. 137

6.2 The pseudo-colored images for the activities *handwaving* (top) and *kicking something* (bottom) that are illustrated in Fig. 6.1. 137

6.3 The dynamic images for the activities *handwaving* (left) and *kicking something* that are illustrated in Fig. 6.1. 139

6.4 A visual overview of the approach of fusion between handcrafted features, pseudocolored and dynamic images. 141

6.5 A visual overview of the proposed approach of fusion between activity, pseudocolored and dynamic images. 141

6.6 The deep convolutional network architectures that have been used in this work and for the cases of (a) pseudo-colored images; and (b) dynamic images. 143

7.1 A visual overview of the proposed data augmentation methodology using occluded samples. 153

7.2 Example skeleton sequences of the activity *make phone call/answer phone* from the PKU-MMD dataset, captured by Microsoft Kinect v2. First row: original skeletons, including all 25 joints (i.e., without any occlusion); Second row: joints corresponding to both arms have been discarded (i.e., the skeleton is partially occluded). 154

7.3 A full skeleton and the 8 cases of occlusion. First row: right arm, both arms, right leg, left arm and left leg; Second row: left arm, both legs, left leg, right arm and right leg. 154

LIST OF TABLES

2.1 The 41 activity classes comprising the PKU MMD Phase #1 dataset. 85

2.2 The 12 activity classes comprising the SYSU 3D-HOI dataset. 85

2.3 The 10 activity classes comprising the UTKinect-Action3D dataset. 87

3.1 Results of the initial arm gesture classification experiment and comparison to the approaches presented in Paraskevopoulos et al. [205], using extremely randomized trees (ERT) and support vector machines (SVM). Numbers denote accuracy. Best result indicated in bold. 101

3.2 Experimental results denoting accuracy of the proposed approach in the 11 selected classes of the PKU-MMD dataset. M, L and R denote the middle, left and right camera angles, respectively. Bold denotes best result. 109

3.3 Experimental results denoting accuracy of the proposed approach in the 51 classes of the PKU-MMD dataset. M, L and R denote the middle, left and right camera angles, respectively. Bold denotes best result. 110

4.1 Cross-view experimental results of the geometric data augmentation approach presented in this Chapter. In all cases, results denote classification accuracy. By B, VA1 and VA2 we denote the baseline approach (i.e., the approach of Chapter 3), the data augmentation and view alignment approach and its modification wherein the output of the most confident classifier is considered, respectively. Differences (Diff.) are calculated as $\%$ percentages. Note that in B the original samples have been used for training and testing, while in VA1 and VA2 the augmented samples have been used for training and testing. 118

4.2 Cross-subject experimental results of the geometric data augmentation approach presented in this Chapter. In all cases, results denote classification accuracy. By B and VA1 we denote the baseline approach (i.e., the approach of Chapter 3) and the data augmentation and view alignment approach, respectively. Differences (Diff.) are calculated as $\%$ percentages. Note that in B the original samples have been used for training and testing, while in VA1 the augmented samples have been used for training and testing. 119

5.1 Experimental results. VA and DA denote view adaptation and domain adaptation approaches, respectively. Figures represent average accuracy percentages over ten runs. In each setup best accuracy achieved is indicated with bold. For each viewpoint adaptation scenario, e.g. “Left to Middle” (indicated $L \rightarrow M$) the four numbers (0, 1, 5 and 10) indicate the percentage of target data instances which are labelled. S is the source model, built by a simple cross-view experiment. T_{rand} and T_{w_S} denote models whose initial weights have been taken randomly and from S, respectively. Note that with no labeled data in the target domain (i.e. 0) these models cannot be trained and hence these columns are omitted. 129

6.1 Proposed features, extracted from the skeletal joints. For features marked with *, respective angles a_{pc}, b_{pc}, c_{pc} are calculated as: $a_{pc}^2 = \left(v_x^J - v_x^{J_c}\right)^2 + \left(v_y^J - v_y^{J_c}\right)^2$, $b_{pc} = v_x^J$, $c_{pc}^2 = \left(v_x^{J_p}\right)^2 + \left(v_y^J - v_y^{J_p}\right)^2$. Note that for the formulation of the triangle, a reference point with coordinates $(v_{x,i}^{J_p}, 0, 0)$ is used. By d we denote the Euclidean distance and F^J is the number of frames for each gesture, which is also used as a feature. 140

6.2 Experimental results of the fusion approach presented in this Chapter. Numbers denote accuracy. Bold numbers indicate best result, per case. Also “PC,” denote pseudocolored images (subsection 6.3.1.1), “DI” denote dynamic images (subsection 6.3.2), “HF” denote handcrafted features (subsection 6.3.3). Moreover, “DFT,” “FFT,” “DCT” and “DCT” denote the Discrete Fourier Transform, the Fast Fourier Transform, the Discrete Cosine Transform and the Discrete Sine Transform, respectively (Chapter 3), while “TF” denotes their fusion. “L,” “R” and “M” denote Left, Right and Middle cameras, while “CV,” “CS” and “SV” denote cross-view, cross-subject and single-view cases. 145

6.3 Comparison of the proposed approach to state-of-the-art research works using PKU-MMD dataset. Numbers indicate mAP (%). Note that Average Precision (AP) is calculated as the area under the precision-recall curve, while Mean Average Precision (mAP) extends the concept of AP by calculating the average AP values across multiple object classes. Bold indicate best results per task. CV, CS denote cross-view and cross-subject cases, respectively. 147

6.4 Comparison of the proposed approach to state-of-the-art research works using SYSU 3D HOI dataset. Numbers indicate accuracy (%). Bold indicate best results per task. 147

6.5 Comparison of the proposed approach to state-of-the-art research works using UT-Kinect Action3D dataset. Numbers indicate accuracy (%). 147

7.1 Results on PKU-MMD dataset. “Full” denotes the case of training dataset comprising non-occluded samples only; “Full+Arms” denotes the case of augmenting “Full” with samples where one or both arms have been occluded; “Full+All” denotes the case of augmenting “Full+Arms” with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies. 156

7.2 Results on SYSU dataset. “Full” denotes the case of training dataset comprising non-occluded samples only; “Full+Arms” denotes the case of augmenting “Full” with samples where one or both arms have been occluded; “Full+All” denotes the case of augmenting “Full+Arms” with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies. 156

7.3 Results on UTKinect-3D dataset. “Full” denotes the case of training dataset comprising non-occluded samples only; “Full+Arms” denotes the case of augmenting “Full” with samples where one or both arms have been occluded; “Full+All” denotes the case of augmenting “Full+Arms” with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies. 157

1. HUMAN ACTIVITY RECOGNITION (HAR)

1.1 Introduction

Human activity recognition (HAR) consists a contemporary research field that intersects key research areas, including computer vision, machine learning and signal processing. The primary objective is the automated identification of human activities based on a series of observations in the temporal and/or the spatial domain. This typically entails detecting and recognizing gestures, postures, or movements of the human body, followed by the interpretation of the observed activities. Sensors play a crucial role in this process, and they can be either wearable devices or installed in the user's environment [140] and may include accelerometers, gyroscopes, magnetometers, cameras, microphones, and even wearable devices like smartwatches. These sensors capture and gather data, including visual and auditory information or motion data, which are then processed enabling the automated recognition of human activities such as walking, running, sitting, standing, cycling, or more complex tasks like cooking, playing sports, or dancing.

There exist several and diverse applications of HAR across various environments. From a social standpoint, it plays a crucial role in supporting social groups requiring protection or assistance for maintaining their independence, including the elderly, children, and individuals with disabilities [8]. In the broader context of healthcare within people's daily lives, the goals of human activity recognition include but are not limited to:

- *Monitoring of Daily Activities*: HAR approaches are used so as to ensure the continuity of daily activities and detecting abnormal behavior;
- *Remote Patient Monitoring/Rehabilitation*: medical professionals can track the movements of patients and ensure they are performing prescribed exercises correctly;
- *Identification of Dangerous Situations*: recognition of potentially hazardous situations or events involving people, such as falls;
- *Guidance to Individuals*: providing guidance to individuals for maintaining "normal" behavior, thereby fostering independence and enhancing well-being or skills;

Beyond social care, HAR finds applications in several other fields, such as:

- *Monitoring and Surveillance*: detection of suspicious or dangerous events in spaces [8, 99];
- *Human-Computer Interfaces*: recognition of posture and/or motion in interfaces, particularly in the gaming and entertainment industries [331].
- *Sports Analysis*: analysis and location of points of interest in sports videos and evaluation of the performance of athletes [255].

The recognition of human behavior poses several challenges regarding the methodology that should be followed. The uniqueness of each individual, including their distinct physical and physiological characteristics, introduces variability in how movements or gestures are performed and consists probably the main challenge. Additionally, the choice of sensor placement is critical. For instance, the viewpoint of a camera or the placement of a wearable accelerometer can significantly impact the data acquisition process. Moreover, the environment itself is dynamic and subject to rapid changes during monitoring. Factors such as variations in lighting conditions or the movement of objects can obstruct camera recordings, leading to the loss of crucial visual information. These considerations influence the selection of a methodology for a specific application and represent common challenges in human behavior recognition systems.

Recent research has predominantly concentrated on leveraging deep learning techniques [114] to infer conclusions regarding a subject's activity. These techniques fall within the domain of machine learning, which explores the design and study of algorithms that learn from data to make predictions. In machine learning, there are two main types of learning: supervised and unsupervised. The forthcoming Thesis will predominantly focus on supervised learning, where the computational system receives inputs and desired outcomes, aiming to establish a general rule mapping inputs to outcomes.

The process of typical HAR approaches adheres to the generic methodology outlined in Fig. 1.1. Specifically, the pipeline begins with the subject engaging into an activity (e.g., climbing stairs). Sensors, such as a gyroscope integrated into a mobile phone or smartwatch, or a depth camera positioned in the space, capture measurements related to the motion of the activity (e.g., 3D acceleration, moving depth image). In the subsequent stage, these measurements undergo processing, which may include an optional feature extraction stage. While traditional machine learning methods necessitate this feature extraction stage, the flexibility of deep learning-based networks has rendered it optional. Finally, data analysis is conducted using trained models and other techniques to derive conclusions regarding the category to which the subject's activity belongs.

There exist several HAR approaches that are based on either wearable sensors or sensors installed within the subject's environment. Different sensors provide different types of data, and the choice of sensors depends on factors like accuracy, cost, power consumption, and ease of integration into the target system. A plethora of such sensors has emerged during the last few years. In the former case, the most popular ones include smartwatches, hand/body worn sensors, smartphones, etc. Moreover, in the latter case, typical sensors include video/thermal cameras, microphones, infrared, pressure, magnetic, RFID sensors, [44] etc. However, it has been shown that wearable sensors are not preferred by the users, while their usability is below average [126, 176]. We should note that overloading the users' environment with a plethora of sensors may be an expensive task, requiring in some cases many interventions in home furniture and/or appliances, e.g., in case of a home environment. Therefore, several low-cost solutions tend to be based solely on one or more cameras, detecting activities using solely the subjects' captured motion.

Common HAR approaches that are based on cameras use as input some raw sensor

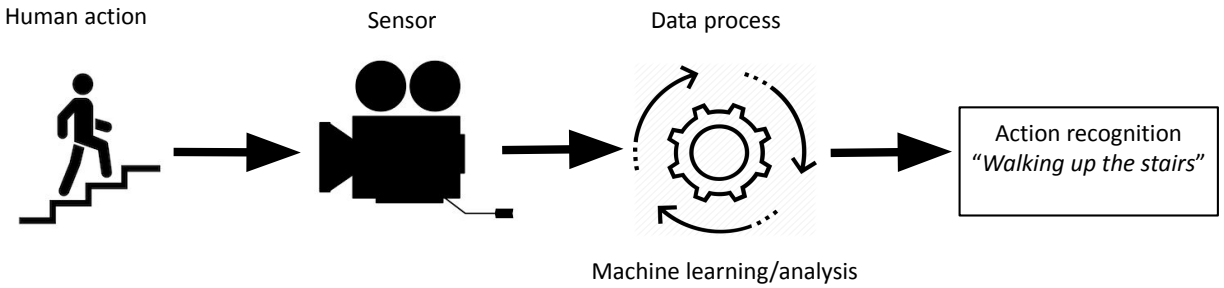


Figure 1.1: A generic HAR pipeline incorporating sensors, model training, evaluation, and interpretation.

data from the performed activity, while their output is the classification result, i.e., the determination of the aforementioned activity. In-between lie the processing and reasoning steps. More specifically, a given HAR approach may consist of (some of) the following steps: a) raw data are pre-processed e.g., for noise and redundancy removal; b) within the video sequence, temporal segmentation takes place, aiming to extract video segments that contain exactly one action to be recognized. Approaches that contain this step perform "segmented" recognition; c) feature extraction, aiming to extract important temporal, spatial or visual features from human motion; d) dimensionality reduction to increase the quality and to decrease the size of features; e) a machine learning algorithm used to train models to recognize and classify human activities.

Note that the feature extraction step requires knowledge and expertise regarding the specific domain of application, in order to provide features that will be able to discriminate between activities. However, a common problem of these features is that even though they may demonstrate satisfactory performance within the given domain of application, they may fail when applied to a similar domain. A solution to the aforementioned problem is omitting the feature extraction step and instead using a classification approach that is based on deep learning. In that case features are not engineered; they are *learnt* from the specific training data. Common deep learning architectures that have been successfully applied in the problem of HAR are the convolutional neural networks (CNNs) [145] and the recurrent neural networks (RNNs) [80], outperforming the majority of traditional machine learning approaches. In case of the segmented HAR tasks, both architectures may be applied, while in case of continuous HAR tasks, RNNs are the most common approach. Although a vast amount of research has been conducted on improving recognition performance, several principal challenges, such as the representation and the analysis of actions, still remain unresolved.

1.2 Basic terminology of motion-based HAR

At this point we will provide the necessary background and terminology regarding motion-based HAR using visual data, which shall be used at the following Chapters of this Thesis.

In the context of computer vision, we may define a *pose* as the temporary suspension of movement, where body joints assume a distinct configuration in space. On the other hand, a gesture may be defined as an expressive, meaningful body motion involving physical movements of the fingers, hands, arms, head, face, or body with the intent of conveying meaningful information and/or interacting with the environment [186]. Thus, poses and gestures are among the basic means of communication between humans, while they may also play a crucial role in human-computer interaction, as they are able to transfer some kind of meaning and/or emotions. However, the more broader notion of human actions is difficult to be restricted in such a simple definition, since contemporary research efforts model actions from simple limb movement to joint complex movement of multiple limbs or even the whole human body [129]. This process is dynamic, and thus it requires a duration of a few seconds.

In the rest of this Thesis, we shall use the following definitions, as they have been formalized by Wang et al. [302], who suggested that activity recognition may be roughly divided into the following tasks: gesture, action, interaction and group activity recognition. More specifically:

- *gestures* are “basic” and instant activities, involving at most a couple of body parts – example gestures are e.g., hand-waving, nodding, etc.;
- *actions*/*activities* require a larger amount of time to be completed and may involve more body parts – example actions are e.g., falling, eating, sitting, etc.;
- *interactions* are performed between two “actors” or between an actor and an object – example interactions of these sub-tasks are e.g., hugging each other and playing guitar, respectively;
- *group activities* may be some combination of the above, typically involving more than two actors – example group activities are e.g., two teams playing a basketball game, a group meeting, etc.

According to Wang et al. [302] HAR tasks may be also be divided into two major categories:

- *segmented recognition*: the given input video sequence contains *only* the action to be recognized. This means that any frame before/after the action, i.e., which does not depict a part of the action, has been removed;
- *continuous/online recognition*: the goal is to recognize actions within a given video; the video may or may not depict a single action.

÷

1.3 Datasets

As we have already mentioned in subsection 1.1, sensors are required to capture human actions. There exist various approaches and techniques which involve some kind of sensor, either "worn" by the subject (e.g., accelerometers, gyroscopes etc.), or monitoring the subject's motion (e.g., cameras). In the latter case, the subject may also wear special "markers" which are used to assist to the identification of several body parts and/or skeletal joints. However, during the last few years several approaches rely solely on a typical RGB cameras, enhanced by depth information, i.e., the user simply needs to stand in front of the camera without wearing any kind of external equipment, while several parts of her/his body are continuously detected and tracked in the 3D space.

Therefore, numerous HAR benchmark datasets have been collected and made accessible to the research community for public use over the last decades. The majority of these datasets have been generated using various sensors, including Motion Capture (Mocap) systems [189], Kinect v1 and Kinect v2 sensors. These datasets typically encompass RGB and depth data (RGB-D), as well as skeleton information, providing a diverse range of combinations for research and experimentation. In this subsection we present some of the most famous datasets that have been commonly used for evaluating HAR. As it has already been mentioned in subsection 1.1, HAR approaches are typically categorized into two major groups: segmented and continuous recognition. We will follow the same categorization for the presentation of the datasets.

The first publicly available datasets such as the KTH [234], were limited to a small number of simple actions e.g., *walking*, *running*, *hand clapping etc.*. Later, a next series of datasets such as the Hollywood dataset [139] targeted more realistic human actions e.g., *answer phone*, *get out of car*, *hand shake* etc., still being limited to a small number of classes. In less than a decade, more challenging datasets such as the UCF101 [254] and the HMDB [135] emerged, containing large numbers of more complex actions, including interactions with objects such as *playing cello*, *horse riding*, *swing baseball bat*, *fencing* etc. However, as we shall discuss in Chapter 2, when working with deep learning approaches, such as the ones that will be presented at the following Chapters of this Thesis, a large-scale multi-class dataset may be the key to effectiveness and robustness. At the following, our goal is to present recent and large-scale datasets, bearing in mind the aforementioned goal.

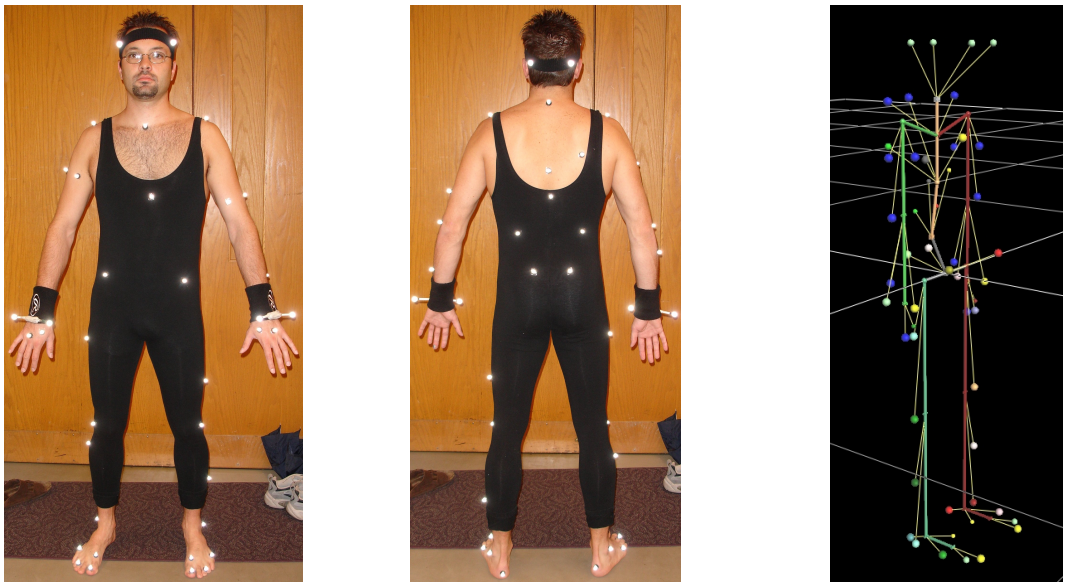
1.3.1 Segmented Datasets

By "segmented" datasets we will refer to those datasets that offer samples that correspond to a whole begin-end action/gestures, with one segment for one action. These datasets are mainly used for classification tasks. At the following we present several segmented datasets commonly used for the evaluation of methods based on deep learning.

CMU Mocap Dataset:[1] CMU Graphics Lab Motion Capture Database is a dataset encom-

[1] http://mocap.cs.cmu.edu/

passing a wide range of human movements and has been recorded using motion capture technology that precisely tracks the positions of markers placed on the subject's body. Figures 1.2a and 1.2b, depict both the front and the back of the subject, with markers placed on his body. Moreover, in Fig. (1.2c), a visualization of the subject's body is depicted: round dots represent the markers, while colored lines represent the bones. This dataset was created in 2001, featuring 45 distinct classes for human interaction, physical activities and sports (basketball, dance), locomotion (running, walking) etc. involving 144 subjects. Overall, it includes a comprehensive collection of 2235 samples comprising RGB and skeleton sequences.



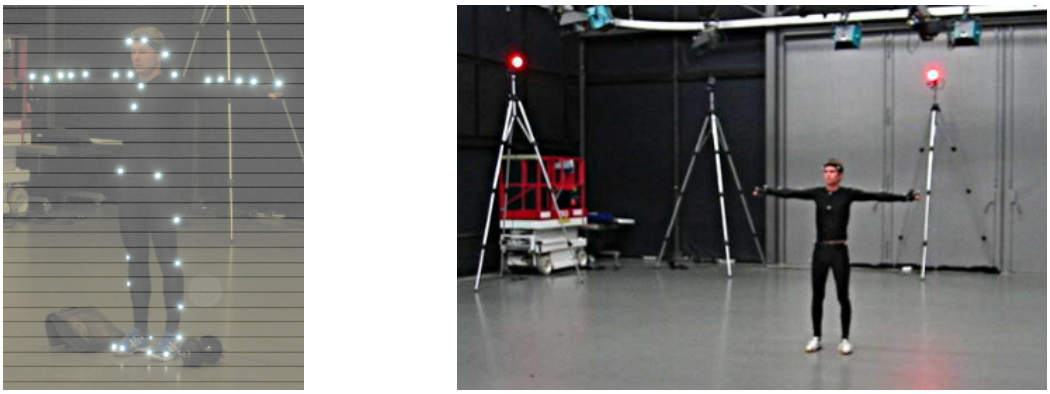
(a) (b) (c)

Figure 1.2: Capture process using Mocap technology in case of CMU Mocap dataset. (a) and (b): positions of markers on the subject's body. (c): visualization of the subject's body – round dots correspond to markers, while colored lines to the body structure.

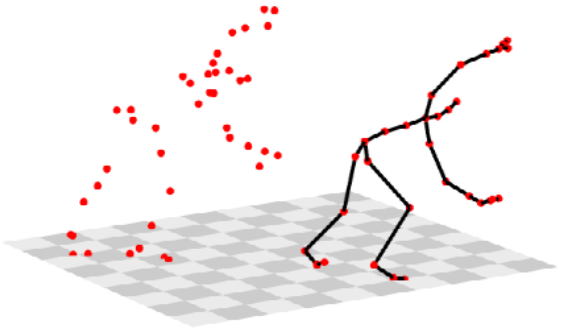
Mocap Database HDM05:[2] The Motion Capture Database HDM05 by Müller et al. [190], consists a well-documented motion capture dataset. It is available in both C3D and ASF/AMC Mocap data formats. C3D is utilized for 3D trajectory-based data, while ASF/AMC supports skeleton-based data. It includes 130 distinct classes, performed by 5 non-professional actors, and comprises a total of 2337 samples. The dataset has been captured using high-resolution cameras arranged in a circular configuration. In Figs. 1.3a and 1.3b the markers on the body and some of the cameras may be observed. Moreover, in Fig. 1.3c the comparison between a C3D file depicting a point cloud and an AMC file depicting a skeleton is illustrated.

MSR-Action3D Dataset: [157] This consists the first RGB-D action dataset. It is a widely used dataset that has been captured using the Kinect v1 and provides sequences of depth

[2]https://resources.mpi-inf.mpg.de/HDM05/



(a) (b)



(c)

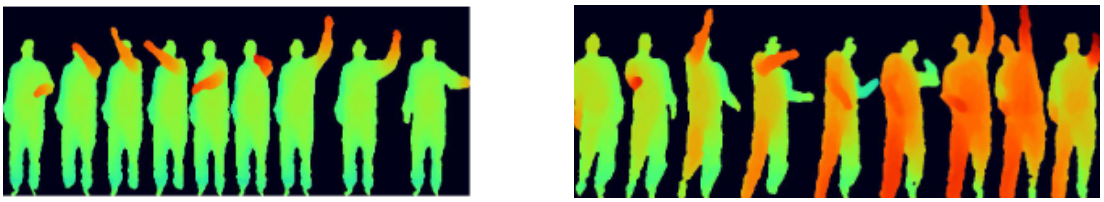
Figure 1.3: Data collection process in case of Mocap Database HDM05. (a) the markers on the body of the human subject; (b) the camera setup used for capturing; (c) comparison between a C3D file depicting a point cloud (left) and an AMC file depicting a skeleton (right). Figure taken from [190].

mpas. The dataset, created in 2010, focuses on activities of daily living, encompassing activities such as high arm wave, horizontal arm wave, tennis serve, draw tick etc. It comprises 20 distinct classes, performed by 7 human subjects, each repeated 3 times. In total, the dataset contains 4020 action samples. In Fig. 1.4 two examples of depth maps are presented, showcasing the actions of drawing a tick and performing a tennis serve.

MSRC-12 Dataset:[3] the Microsoft Research Cambridge-12 Kinect gesture dataset [65] consists of sequences of human movements, represented as body-part locations. It encompasses 594 sequences captured from 30 subjects, performing 12 different gestures. The dataset was constructed in 2012 using the Kinect v1 camera, providing skeleton data for analysis and recognition. The gestures within the dataset have been classified into two categories:

- iconic gestures: these gestures establish a direct correspondence between the ges-

[3]https://research.microsoft.com/en-us/um/cambridge/projects/msrc12/



(a) (b)

Figure 1.4: Depth maps from the MSR-Action3D dataset. (a) draw tick; (b) tennis serve. Figure taken from [157].



(a) (b)

Figure 1.5: Example actions of the MSRC-12 Dataset. Left: “draw tick;” right: “tennis serve.” Figure taken from [65].



(a) (b)

Figure 1.6: Example actions of the MSRC-12 Dataset. Left: “start music/raise volume (of music);” right: “navigate to next menu.” Figure taken from [65].

ture and its reference, creating a clear visual representation. In Fig. 1.5 gestures “shoot a pistol,” “crouch” or “hide” are depicted;

- metaphoric gestures: these gestures represent abstract concepts, conveying meaning beyond a direct visual resemblance. In Fig. 1.6 gestures “start music/raise volume (of music)” and “navigate to next menu” are depicted.

Provided modalities of MSRC-12 are (a) a textual part that aims to provide a description of the gesture kinematics; (b) a static visual part that is composed of ordered series of consecutive static images of the subject performing the gesture; and (c) a moving visual part, i.e., video of the subject performing the gesture.

MSRDailyActivity3D Dataset: [291] This dataset was captured using a Kinect camera

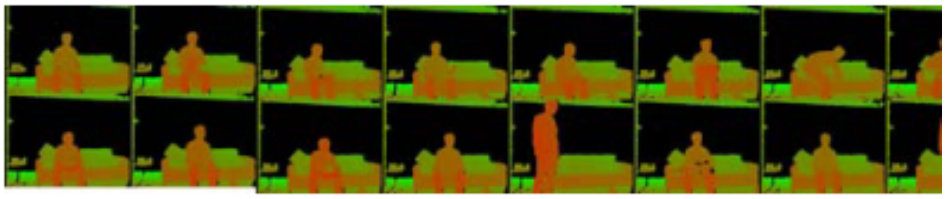
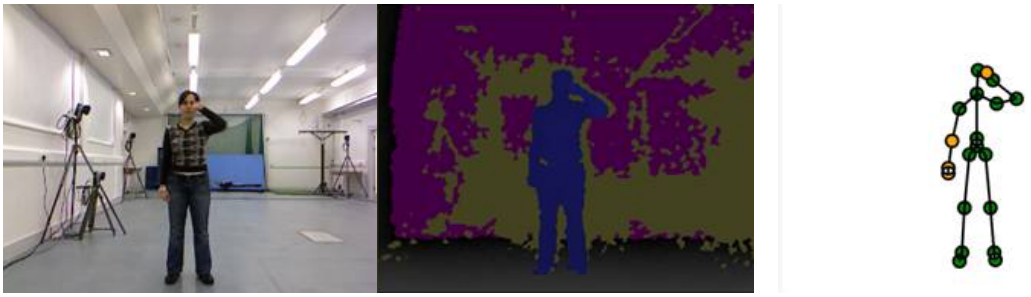


Figure 1.7: Sample frames of the DailyActivity3D dataset. Figure taken from [291].



(a) (b) (c)

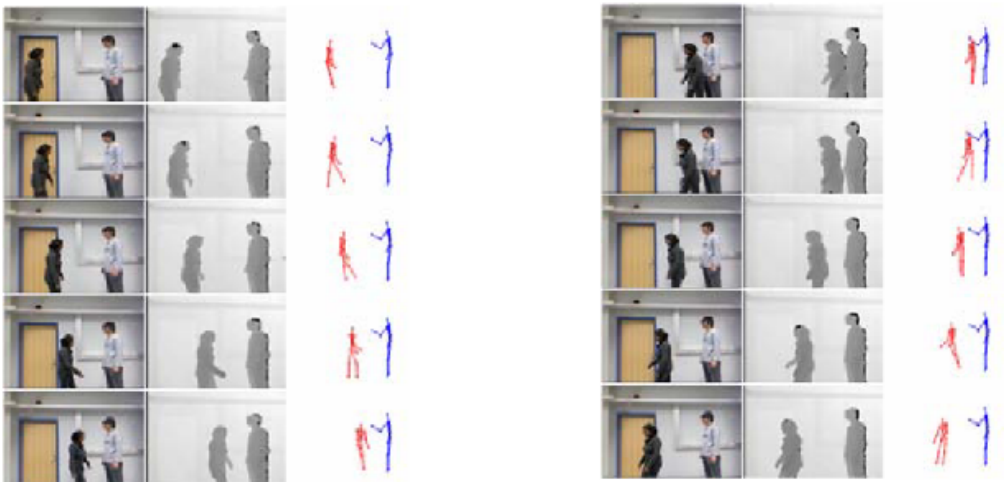
Figure 1.8: Samples from G3D dataset: (a) color video frame; (b) the corresponding depth map; and (c) the extracted skeleton from the human subject. Figure taken from [21].

v1 in 2012 within a living room setting. This specific environment was chosen so as to encompass a broad range of human daily activities. It comprises 16 classes such as drink, eat, read book, call cellphone etc. It was collected by involving 10 human subjects which were either sitting on the sofa or standing close to it. A set of 320 samples has been collected. A fixed camera in front of the sofa captures depth data, and alongside it, skeleton data are recorded. However, the extracted joint positions from the tracker are noisy, mainly because the actors are either sitting on or standing close to the sofa. In Fig. 1.7 several examples from the dataset are illustrated.

G3D Dataset:[4] the G3D dataset [21] comprises a variety of gaming actions that were captured using the Microsoft Kinect and provide RGB, depth and skeleton data. The dataset involves 10 subjects executing a total of 20 gaming actions, such as, “punch right,” “punch left,” “kick right,” “kick left,” etc. Each skeleton in the dataset includes information regarding the player’s position and pose. The majority of sequences in the dataset feature multiple actions conducted in a controlled indoor environment. This environment is characterized by a fixed camera, which is a typical setup for gesture-based gaming scenarios. In Fig. 1.8 we illustrate an indicative color video frame, the corresponding depth map and the extracted skeleton from the human subject.

SBU Kinect Interaction Dataset: [322] this dataset comprises a total of 300 interactions, meticulously recorded in a controlled laboratory environment at Stony Brook University in 2012. The recordings include both color images and depth maps, each at a resolution of 640×480 pixels. Eight subjects actively participated in diverse activities involving interac-

[4] http://dipersec.king.ac.uk/G3D/G3D.html



(a) (b)

Figure 1.9: Sample activities of the SBU Kinect Interaction Dataset: (a) “approaching;” (b) “departing.” Each row contains a color image, a depth map, and extracted skeletons at the first, 25%, 50%, 75%, and the last frame of the entire sequence for each interaction. Red skeleton corresponds to the active subject, blue skeleton corresponds to inactive subject. Figure taken from [322].

tions between two actors. Across these interactions, there are 9 distinct activity classes, including actions such as approaching, departing, pushing, kicking, and more. Examples are illustrated in Fig. 1.9. This dataset is considered as particularly valuable, since it provides comprehensive ground truth labels, specifying the action category and identifying the “active” and “inactive” actors in each segmented video.

Berkeley MHAD Database: Berkeley Multimodal Human Action Database [194] utilizes five distinct sensor systems for data capture and has been created in 2013. It uses a Mocap system to construct the ground truth, 2 Microsoft Kinect v1 cameras for depth data acquisition, 4 multi-view stereo vision cameras for capturing multi-view video data, 6 wireless accelerometers to capture dynamics of motion and 4 microphones to capture audio for video analysis. This experimental setup is illustrated in Fig. 1.10. The dataset consists of 11 classes captured by 12 subjects and it has 660 action sequences. Figure 1.11 depicts snapshots from all the actions captured by the front-facing camera, along with the corresponding point clouds extracted from the Kinect data. The specified set of actions includes:

- actions involving movement in both upper and lower extremities, such as “jumping in place” etc.;
- actions with high dynamics in upper extremities, like “clapping hands” etc.;
- actions with high dynamics in lower extremities, for example, “sitting down.”

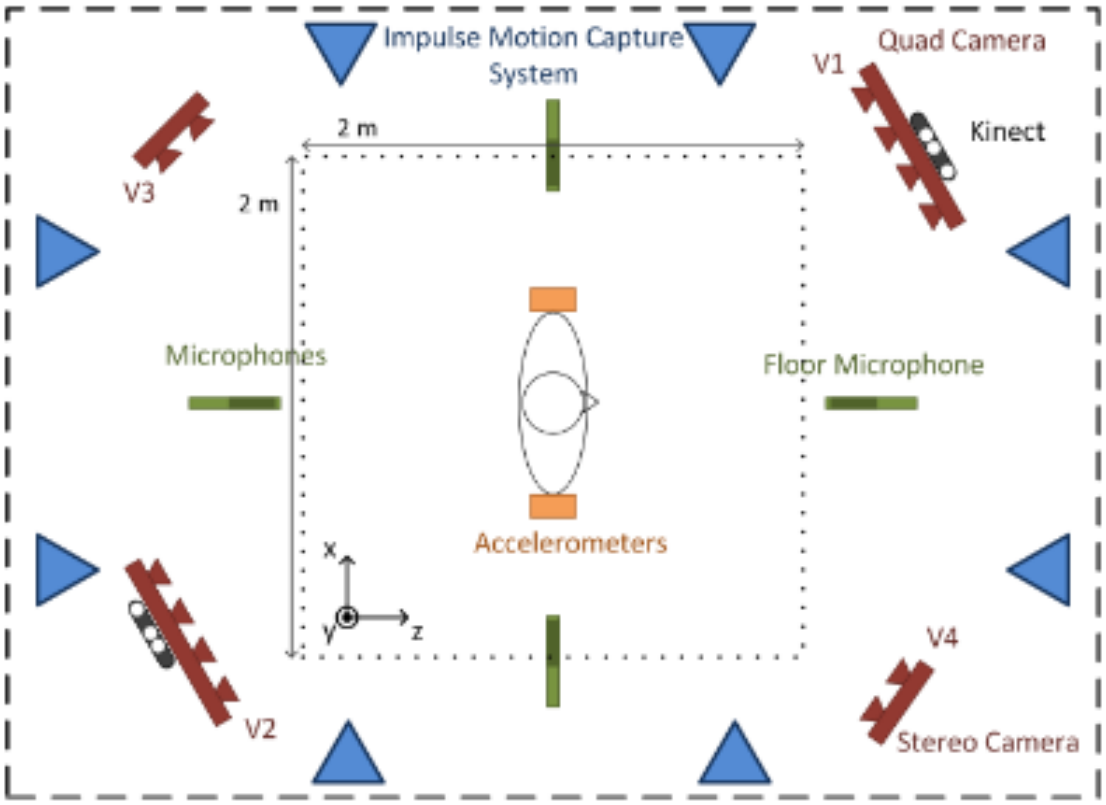


Figure 1.10: The experimental setup that has been used for the acquisition of the Berkeley MHAD Database. Figure taken from [194].

Figure 1.12 illustrates a snapshot of the action throwing, captured from the reference camera of each camera cluster and from the two Kinect cameras. This figure aims to illustrate the wealth of information that can be obtained through multi-view and depth observations, highlighting the advantages over a single viewpoint.

Northwestern-UCLA Multiview Action 3D Dataset: this dataset has been proposed by Wang et al. [292] and utilizes 3 Kinect v1 cameras to capture depth, RGB, and skeleton data. It encompasses 10 distinct classes of actions, including activities like “picking up with one hand,” “picking up with two hands,” “dropping trash,” “walking around,” and more. The actions were performed by ten actors, resulting in a total of 1475 sequences. This dataset was created in 2014. The experiments are conducted under three distinct settings:

- cross-subject setting: training data is composed of samples from 9 subjects, leaving out the samples from 1 subject for testing;
- cross-view setting: training data comprises samples from 2 cameras, while samples from 1 camera are used for testing;
- cross-environment setting: the learned model is applied to the same action but captured in a different environment. Examples of cross-environment testing data are illustrated in Fig. 1.14b.

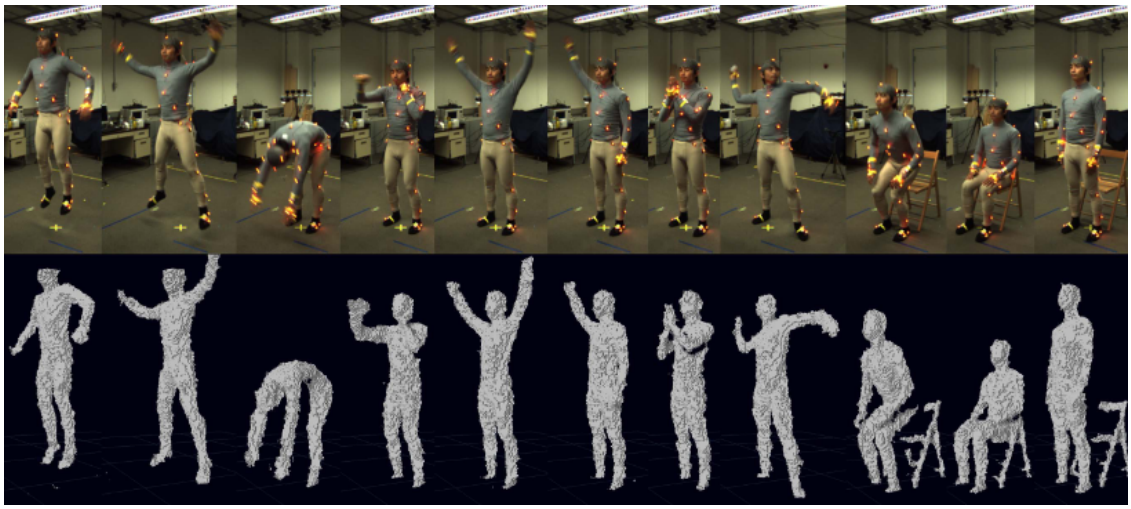


Figure 1.11: Snapshots from all the actions available in the Berkeley MHAD Database, displayed along with the corresponding point clouds obtained from the Kinect depth data. Depicted actions from left to right: “jumping,” “jumping jacks,” “bending,” “punching,” “waving two hands,” “waving one hand,” “clapping,” “throwing,” “sit down/stand up,” “sit down,” “stand up.” Figure taken from [194].

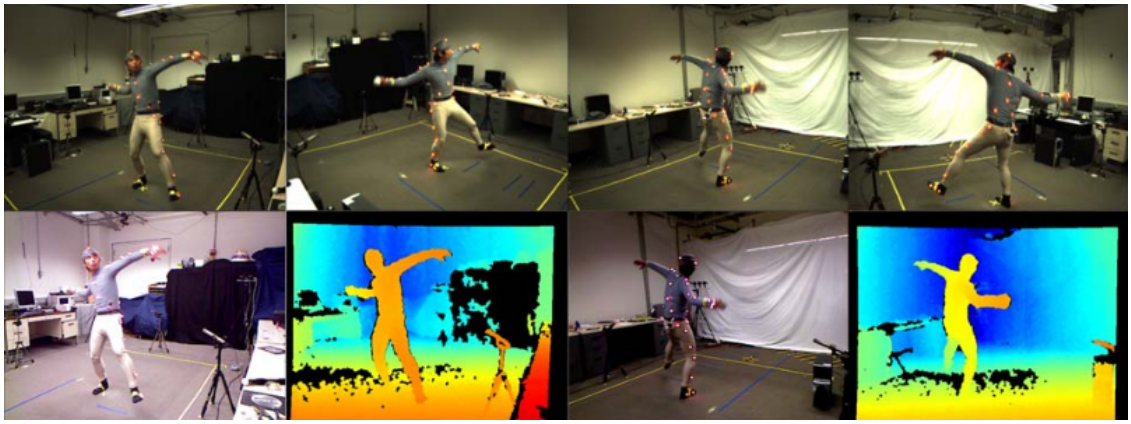


Figure 1.12: Throwing action from the Berkeley MHAD Database as captured by the reference camera in each camera cluster, as well as from the two Kinect cameras. Figure taken from [194].

ChaLearn LAP IsoG Dataset :[5] [290] this large-scale dataset has been created in 2016 and comprises 249 gesture classes with a substantial collection of 47,933 RGB-D video samples. These samples have been captured using the Kinect v1 camera and involved the participation of 21 individuals. The dataset is meticulously partitioned into training, validation, and test sets. This partitioning strategy ensures that samples from different subjects are included in each set, preventing the gestures of a specific subject in the validation and test sets from appearing in the training set. This separation is crucial for

[5]http://www.cbsr.ia.ac.cn/users/jwan/database/isogd.html

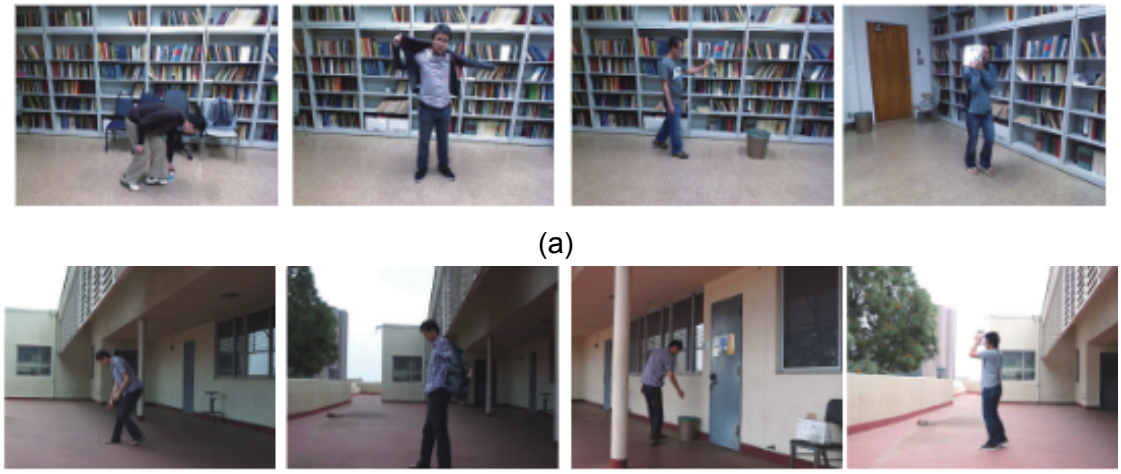


Figure 1.13: Sample frames of Multiview Action3D dataset and specifically from the cross environment setting. Figure taken from [292].

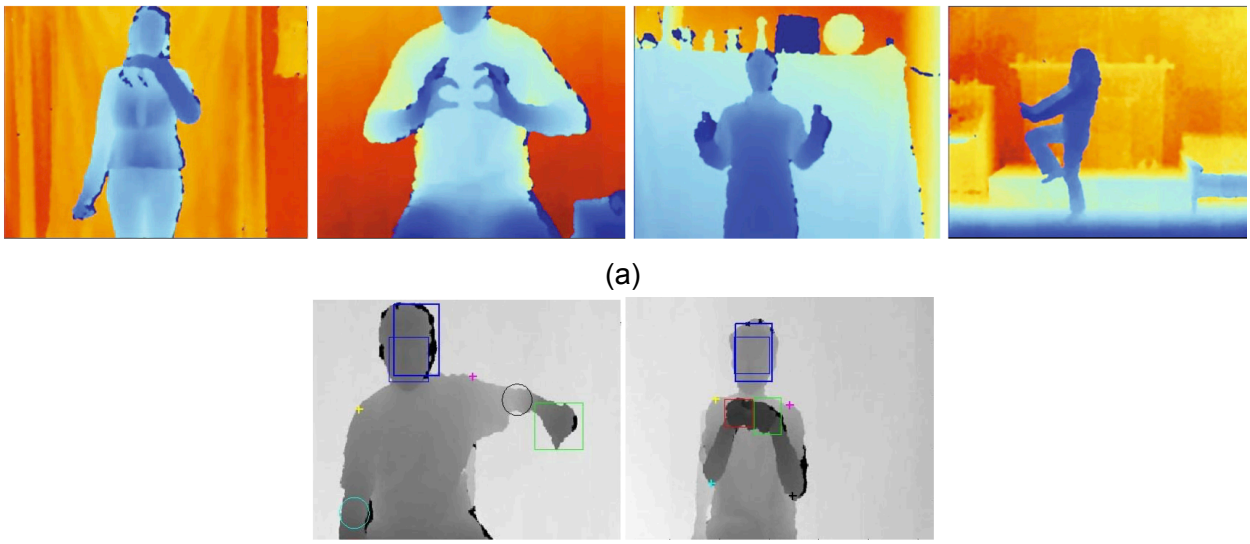


Figure 1.14: Samples from ChaLearn LAP IsoG Dataset: (a) Colorized sample depth maps; (b) body part annotations of head, shoulder, elbow and hand positions, with circles denoting occluded body parts.

evaluating and generalizing the recognition model effectively.

NTU RGB+D Data Set The ROSE LAB[6] at the Nanyang Technological University of Singapore has constructed two human activity datasets, namely NTU RGB+D and NTU RGB+D 120 [235]. The first dataset comprises a total of 60 action classes and contains 56,880 video samples, while the second one is an extension of the first, featuring additional 60

[6] https://www.ntu.edu.sg/rose



Figure 1.15: Sample frames from activities of the two datasets NTU RGB+D datasets. Figure taken from [236].

classes on top of the original 60, resulting to a total of 120 activity classes. Furthermore, the second dataset includes another 57,600 video samples, bringing the total number of samples in this dataset to 114,480. Both of these datasets provide various data modalities, i.e., RGB videos, sequences of depth maps, 3D skeletal data, and infrared (IR) videos for each individual sample are provided. This dataset has been captured using three Kinect V2 cameras, resulting to RGB videos having a resolution of 1920×1080, while the depth maps and IR videos share a common resolution of 512×424. Additionally, the 3D skeletal data comprises the 3D coordinates of 25 body joints at each frame. The activities in both these datasets are categorized into three main groups: daily actions, mutual actions, and medical conditions, while sample frames are illustrated in Fig. 1.15.

We should herein note that the **UTKinect-Action3D Dataset** [311] and the SYSY-3D-HOI dataset [102] also fall in this category. However, since these are among of the datasets that have been utilized for the evaluation of the experiments presented in this Thesis, they will be presented in detail in Chapter 2.

1.3.2 Continuous/Online Datasets

As “continuous/online” datasets we shall refer to those datasets where each video sequence may contain one or more activities/gestures, and the segmented position between different motion classes are unknown. These datasets are mainly used for action detection, localization and online prediction. There do not exist as many datasets as in the case of the segmented ones.

ChaLearn2014 Multimodal Gesture Recognition Dataset:[7] this dataset is a multimodal dataset collected in 2014 by Escalera et al. [56] and is similar to the previously mentioned Chalearn LAP IsoG dataset. It has been collected using the Kinect v1 camera, encompassing RGB, depth, skeleton, and audio modalities. The recorded sequences involve a single user positioned in front of the camera, engaging in natural communicative Italian gestures and featuring a vocabulary of 20 Italian gesture categories. The dataset includes 13,858 video fragments. The gestures are performed by 27 individuals under diverse conditions, encompassing variations in clothing, positions, backgrounds, and lighting.

ChaLearn LAP ConGD Dataset[8]: this dataset has been crated by Wan et al. [290] and is also similar to the previously mentioned Chalearn LAP IsoG dataset. Therefore, it is also a large dataset that has been collected with Kinect V1 camera. It has 47,933 RGB-D gestures in 22,535 RGB-D gesture videos and involves 21 participants. It focuses on cross-user evaluation and provides manually labeled temporal segmentation to obtain the start and end frames for each gesture in continuous videos.

We should herein note the **PKU-MMD Dataset** [163] also falls in this category, even though it also provides activity boundaries, allowing for segmented recognition. However, since this is one of the datasets that have been utilized for the evaluation of the experiments presented in this Thesis, it will be presented in detail in Chapter 2.

1.4 Recent Approaches on HAR

In this Section we will present recent research efforts in the broader area of motion-based HAR using optical data. We will focus solely on deep learning-based methods, in an effort to provide a comprehensive understanding of this research area. We herein remind that a typical categorization considers four main categories, based on the origin of the extracted motion-based visual features. Specifically, research works are categorized into a) RGB-based; b) depth-based; c) skeleton-based; and d) RGB+D-based. Note that this categorization aligns with the taxonomy outlined by Wang et al. [302] and is illustrated in Fig. 1.16. As it has been previously discussed in subsection 1.2, each category is further divided into continuous/online motion recognition. Note that focus will be given into the two of the most dominant neural network architectures that have been widely used in diverse deep learning approaches, i.e., convolutional neural networks (CNNs) and recurrent neural networks (RNNs), however we will also present works that are based on other types structured networks.

1.4.1 RGB-based Motion Recognition using Deep Networks

Color information encoded in the RGB channel is mainly characterized by properties of shape, color, and texture and may be used for subsequent feature extraction so as to con-

[7] https://gesture.chalearn.org/2013-multi-modal-challenge
[8] http://www.cbsr.ia.ac.cn/users/jwan/database/congd.html

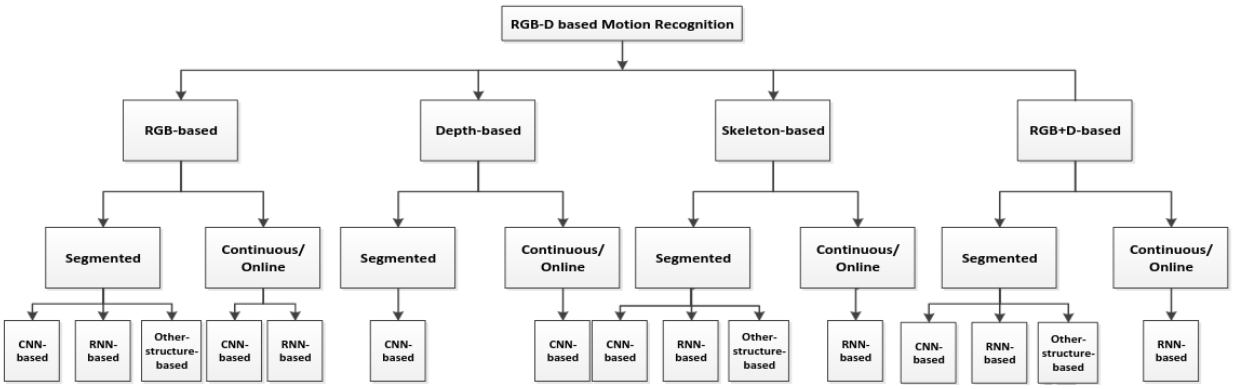


Figure 1.16: A brief taxonomy of RGB+D based motion recognition approaches. Figure taken from [302].

vey rich information. Therefore, many research works are based on the direct employment of network architectures, such as 2D CNNs [90, 134, 246], for extracting features at a per-frame level. However, RNN-based architectures are also used.

1.4.1.1 Segmented RGB-based Motion Recognition

1. **CNN-based approaches**: One approach for encoding simultaneously the spatio-temporal and structural properties of a given scene is based on CNN for feature extraction per frame, while temporal information is encoded upon fusion of features extracted of consecutive frames. A typical example is the work of Karpathy et al. [120] who explored several temporal fusion methods, whilst they proposed the notion of "slow fusion," wherein subsequent network layers gain progressively more global information in spatial and temporal dimensions. In the same context, several pooling methods, in the temporal domain, have been suggested, with a notable example the one presented by Ng et al. [321].

   A second approach is based on the extension of the convolutional operations into the temporal dimension of motion. Consequently. 3D-CNNs have been proposed to address the need of simultaneous extraction of spatial and temporal features, while experimental evaluation verified their dominance over their 2D counterparts [112]. A popular approach is the C3D descriptor, based on 3D pooling layers has bee proposed by Tran et al. [275]. Its main drawback is performance in longer video sequences, which has been addressed by the approach of Varol et al. [281] who proposed a methodology based on long-term temporal convolutions. To decrease the number of network parameters, Sun et al. [263] combined 1D and 2D filters to form a 3D filter.

   Another popular approach is to encode the spatio-temporal dynamics of video sequences into 2D "dynamic" images, which may be then used as input to CNN for recognition. Notable works in this area are the ones of Bilen et al. [17], Fernando

and Gould [64] and Fernando et al. [63] who proposed variations of rank pooling to encode videos into dynamic sets of images.

The final approach in this category is based on separate networks for capturing spatial and temporal features, namely "multiple-stram" networks. Notable works include the one of Simonyan et al. [247] who fused the output of a spatial stream network whose input was a set raw video frames with the output of a temporal one, whose input was a set of optical flow fields. This work was then extended by Wang et al. [293] upon incorporating trajectory-constrained sampling and pooling and by Zhang et al. [323] who replaced optical flow with motion vectors. Wang et al. [294] focused on the temporal structure of video sequences, by extracting features from short video snippets, while Wang et al. [307] focused on region-based features. A typical drawback of such approaches consists of the mismatch among different features, thus a spatial-temporal fusion step is needed Feichtenhofer et al. [59].

2. **RNN-based approaches:** Many research works used RNNs in combination with CNNs to address the problem of temporal feature extraction. Baccouche et al. [12] employed a cascade of 3D CNN and LSTM, each trained separately. Contrary to this approach, the one of Donahue et al. [49] proposed a mixed architecture namely "long-term recurrent convolutional network (LRCN), wherein both networks have been trained simultaneously. Both architectures have been also used for feature extraction, which were then fused such as in the work of Ng et al. [321]. Other works incorporate attention models, such as the one of Sharma et al. [237] which used to learn the importance of relevant frame parts, and the one of Du et al. [53] who proposed a recurrent pose-attention network (RPAN) for adaptive learning of highly discriminative pose-related features. RNN-based approaches also include the recurrent neural network fisher vector (RNN-FV), proposed by Lev et al. [150], upon replacing the GMM probabilistic model in Fisher Vector with an RNN, in an effort to avoid assumptions regarding the distribution of the available training data, a combination spatio-temporal graphs and RNN proposed by Jain et al. [113], the Lattice-LSTM (L2STM) network proposed by Sun et al. [264], used to learn independent hidden state transitions for a given region and the biologically-inspired ShuttleNet1 proposed by Shi et al. [241].

3. **Approaches based on other architectures**: In addition to the prevalent CNN- and RNN-based approaches for motion recognition from RGB modalities, alternative structures have been employed for this task, such as feed-forward hierarchical template matching, [111], Space-time Deep Belief Network (ST-DBN) based on convolutional restricted Boltzmann machine (CRBM) [36], convolutional gated restricted Boltzmann machine (convGRBM) [270], convolution based on independent subspace analysis [142, 271], a three-layer autoencoder, namely "Dynencoder" [314], an LSTM autoencoder, wherein the encoder learns a compact representation, while the decoder reconstructs the input sequence [257] and adversarial mechanisms [77] for training multi-scale convolutional network for generating future frames [179].

1.4.1.2 Continuous RGB-based Motion Recognition

It is apparent that the majority of the previously mentioned methodologies strongly depends on the accessibility of a segmented video dataset for model training. Nevertheless, constructing such a dataset is usually a costly and time-consuming undertaking, especially considering that real-life activities may unfold continuously. The advent of untrimmed video datasets [25, 43, 163, 290, 319] has proven to be a valuable asset for research and has posed challenges in motion recognition within the context of untrimmed videos. Numerous research endeavors in this category draw inspiration from the well-established R-CNN [72, 73, 224], a network designed for region-proposal-based object detection. Consequently, these approaches aim to suggest temporal segments of an input untrimmed video that capture a particular activity. Typically, they initiate this process by generating a reduced set of candidate temporal windows. Subsequently, an action classifier evaluates each proposal independently, categorizing it based on the depicted action. In the following, we will introduce representative methods that leverage both CNNs and RNNs, starting with proposal-based strategies—approaches centered on detecting temporal boundaries of actions.

1. **CNN-based approaches**: In continuation of the previously mentioned two-stream network [247], in the work of Gkioxari and Malik [74] region proposals are categorized at a frame-level, by considering not only static but also motion cues and "action tubes" are produced at both action and video levels. Weinzaepfel et al. [309] opted for selecting the highest-scoring frame-level proposals, tracking them within the whole video using a multi-scale sliding window approach. Shou et al. [245] introduced multi-stage segment-based 3D CNNs to create candidate segments, which are the used for recognizing both present actions and their corresponding boundaries. Peng and Schmid [212] utilized RGB and optical flow to produce comprehensive proposals, subsequently temporally localizing the detected actions by stacking optical flows. A novel network architecture, UntrimmedNet, was proposed by Wang et al. [295], and it was applied to detect windows containing actions. Structured temporal pyramids were employed by Zhao et al. [334] for a similar purpose. In contrast to previous methods, Zhu et al. [337] integrated spatial and temporal information, along with an efficient path trimming method for detecting action boundaries.

   Hou et al. [97] extended R-CNNs to 3D, leveraging the Tube Proposal Network and Tube-of-Interest pooling layer. This way they were able to address variations in tube proposal sizes both in the spatial dimensions and in the temporal dimension. Saha et al. [229] proposed an approach for the regression and classification of 3D region proposals which extend into two consecutive video frames. Kalogeiton et al. [116] introduced an Action Tubelet detector, extending the SSD framework from 2D to 3D. Gao et al. [67] presented a temporal unit regression network (TURN) model for predicting action proposals and refining temporal boundaries, while Singh et al. [249] designed a real-time online algorithm for constructing and labeling action tubes. Additionally, Lea et al. [143] adopted temporal convolutional networks (TCNs), featuring a hierarchy of temporal convolutions, in order to provide more accurate action 3D

boundaries.

2. **RNN-based approaches**: Besides proposal-based methods utilizing CNNs, numerous approaches employing RNNs for temporal modeling have been put forth. Escorcia et al. [57] introduced “deep action proposals” (DAPs), a method grounded on C3D [275] as the visual encoder and an LSTM for the sequence encoder to generate action proposals. However, such approaches have demonstrated computational expensiveness, often dividing the input video to numerous short and overlapping temporal windows, thereby imposing a substantial computational burden. To mitigate this challenge, Buch et al. [24] used a single-stream temporal action proposal method, eliminating the aforementioned drawback.

Moreover, several proposal-independent methodologies have been introduced. Yeung et al. [318] utilized an RNN-based agent, trained for the formulation and the refinement of action instance hypotheses. This agent encompasses an observation and a recurrent network. Singh et al. [248] introduced a multi-stream bi-directional RNN to provide finer action boundaries. In this approach, a tracking algorithm was employed to identify the bounding box of the person, and both motion and appearance streams were trained within the tracked bounding box. The utilization of a bi-directional LSTM enabled the modeling of temporal dynamics not only during, but also amid actions.Ma et al. [174] proposed a modification of the RNN objective to better capture the progression of activities. Huang et al. [100] proposed a dynamic programming approach based on an extended connectionist temporal classification (ECTC). Finally, Dave et al. [42] employed a sequence of RNNs sequentially to make predictions regarding future actions and subsequently corrected them based on observations.

1.4.2 Depth-based Motion Recognition with Deep Learning

In contrast to the RGB modality, the addition of depth offers several advantages. Specifically, it provides illumination invariance, invariance to color and texture changes and facilitates the extraction of 2D human silhouettes and 3D human skeletons. Despite these benefits, there is a scarcity of research on depth-based HAR, mainly due to the absence of color and texture information; depth maps contain only information regarding each pixel’s distance to the camera. Note that color and texture are the main source of information that is encoded by using CNNs, therefore, their absence leads to weakening or even loss of their discriminative representation capabilities [171]. CNNs excel in learning hierarchical features from RGB images, utilizing color and texture information for recognition tasks. In the absence of these visual cues, the discriminative capacity of CNNs may be compromised, making it challenging to achieve optimal performance in depth-based action recognition. Secondly, the existing depth data available for training deep learning models is relatively small-scale. Traditional pipelines in deep learning rely heavily on data-driven approaches, learning representations directly from pixel-level information. When such models are optimized on limited training data, there is a risk of overfitting, where the model may not generalize well to unseen data. The scarcity of large-scale depth datasets poses

a challenge in training robust and generalizable deep learning models for depth-based motion recognition.

Presently, the predominant strategies for depth-based motion recognition involve CNN-based methods. Researchers are actively exploring techniques to address the challenges associated with depth data, aiming to enhance the performance and applicability of deep learning models in this modality.

1.4.2.1 Segmented Depth-Based Motion Recognition

Methodologies in this category are typically CNN-based approaches. Wang et al. [297, 298] introduced the Weighted Hierarchical Depth Motion Maps (WHDMM) + 3ConvNet method for human action recognition from depth maps. They aimed to provide a viewpoint invariant approach and to achieve this the created artificial rotated data samples, to simulate the effect of viewpoint change. To compensate for the effect of the variable action durations, they constructed WHDMMs at several temporal scales. They also adopted a transfer learning approach by re-training and fine-tuning ImageNet models. Wang et al. [304] used rank pooling to produce dynamic images based on depth sequences and produced three different 2D representations, which aimed to simultaneously capture information regarding the pose and the motion of the subject. In a continuation of this work [304] they modified rank pooling in a hierarchical manner.

In another development, Wang et al. [301] combined pairs of dynamic images to represent depth sequences. They used rank pooling in both temporal directions and aggregated both motion and skeleton joint structure information. The produced 2D images were then used as input to CNNs. This work was later extended by Hou et al. [98] upon exploiting spatio-temporal-structural information. Rahmani et al. [219] created synthetic depth images which were then used as a means to learn a view-invariant human poses and motion. They also used MoCap data to fit the aforementioned models into real action examples and proposed a group sparse Fourier Temporal Pyramid, to encoded motion features.

1.4.2.2 Continuous Depth-Based Motion Recognition

Methodologies in this category are typically CNN-based approaches. For continuous gesture recognition, Wang et al. [299] devised a two-step approach comprising segmentation of depth sequence and application of a modification of motion maps. Then, in a continuation of this work [304] they proposed three motion representations, namely “Dynamic Depth Images” (DDI), “Dynamic Depth Normal Images” (DDNI), and “Dynamic Depth Motion Normal Images” (DDMNI), which were based on rank pooling. In all cases, pre-trained CNNs were used for classification.

1.4.3 Skeleton-based Motion Recognition with Deep Learning

In skeleton-based motion recognition using deep learning, methods that have been presented are also categorized as RNN-based, CNN-based, and other-architecture-based, for both segmented motion recognition and continuous/online motion recognition.

Specifically, the first approach involves RNNs, which are particularly well-suited for processing data sequences. RNN-based methods leverage the temporal dynamics of skeleton data for both segmented and continuous motion recognition. These models can capture dependencies and patterns over time, making them effective for understanding the temporal evolution of actions. The second approach employs CNNs and other architecture-based methods for segmented motion recognition. CNNs have demonstrated success in image and video analysis tasks, and their application to skeleton data involves extracting hierarchical spatial features from joint positions. Other architecture-based methods may use combinations of different neural network layers to capture complex spatial dependencies in skeleton sequences. The third approach focuses on RNN-based methods for continuous or online motion recognition. In this context, RNNs are employed to process streaming skeleton data in real-time. These models adapt to the evolving nature of continuous actions and make predictions based on the sequential input of joint positions.

We should herein note that each of the aforementioned approaches exhibits both advantages and drawbacks, and the choice of a specific method depends on the requirements of the given motion recognition task, such as whether the input data is segmented or continuous, and the level of temporal information needed for accurate recognition.

1.4.3.1 Segmented Skeleton-Based Motion Recognition

1. **CNN-based approaches**: Researchers have explored various approaches for skeleton-based action recognition using deep learning. Methods include representing skeleton sequences as matrices and converting them into images for CNN-based feature extraction (Du et al., [50]). Wang et al. [300] introduced Joint Trajectory Maps (JTM) by mapping joint trajectories into the HSV color space, and Hou et al. [96] employed color encoding on orthogonal canvases. Li et al. [153] proposed Joint Distance Maps (JDM) by encoding pair-wise distances between skeleton joints into texture images. Liu et al. [169] enhanced skeleton visualization and used a sequence-based view-invariant transform with multi-stream CNN fusion. Ke et al. [121] designed invariant vector-based features, and Ke et al. [122] represented sequences as clip images for CNN-based learning. Kim and Reiter [253] utilized Temporal Convolutional Neural Networks (TCN) with Res-TCN for interpretable hidden representations and model parameters.

2. **RNN-based approaches**: In this category of approaches, skeleton features are used as input to an RNN to exploit the temporal dynamics of motion. Du et al. [51,52] used parts of the skeleton sequence as input to bi-directional RNNs/LSTMs. Veeriah et al. [284] introduced a differential gating scheme for LSTMs to emphasize

information gain from salient motions. Zhu et al. [338] applied regularizations in fully connected layers and introduced dropout to LSTM neurons. Shahroudy et al. [235] proposed a part-aware LSTM (P-LSTM). Liu et al. [165] presented a spatio-temporal LSTM (ST-LSTM) network to model dependencies between joints in both temporal and spatial domains. Song et al. [252] introduced a spatio-temporal attention model with LSTM for discriminative joint mining. Liu et al. [166] proposed a Global Context-Aware Attention LSTM (GCA-LSTM) to focus joints that carry most information regardind motion, with global context. Zhang et al. [329] investigated simple geometric features with a 3-layer LSTM framework. Lee et al. [149] proposed an ensemble Temporal Sliding LSTM (TS-LSTM) network with short-, medium-, and long-term components. Li et al. [158] used multiple RNNs in a tree-like hierarchy. Zhang et al. [325] addressed large view variations with a self-regulated view adaptation scheme integrated into an LSTM for dynamic determination of observation viewpoints during recognition.

3. **Approaches based on other architectures**: In addition to RNN- and CNN-based architectures, there are several other approaches that have been used in this problem. Salakhutdinov et al. [231] proposed an approach that integrates deep netweorks with structured hierarchical Bayesian models. Wu and Shao utilized deep belief networks (DBN) to model the spatial distribution of skeletal joints. Ijjina et al. [107] employed autoencoders to learn features from skeletal data. Finally, Huang et al. [101] combined the Lie group structure with a deep network to learn improved features.

1.4.3.2 Continuous Skeleton-Based Motion Recognition

Unlike previously mentioned approaches that were based on temporally segmented skeleton sequences, the one proposed by Li et al. [160] applied an LSTM to capture long-range temporal dynamics, i.e., without using sliding windows. This approach was more computationally efficient.

1.4.4 RGB+D-based Motion Recognition with Deep Learning

As it has been previously discussed, each of the typical visual modalities, i.e., RGB, depth and 3D skeletons has its own strengths and limitations. Therefore, it is expected that the combination of their strengths could improve recognition. Corresponding approaches may be classified as follows: (i) CNN-based, (ii) RNN-based, (iii) approaches based on other architectures for segmented motion recognition, and (iv) RNN-based continuous/online motion recognition.

1.4.4.1 Segmented RGB+D-Based Motion Recognition

1. **CNN-based approaches:** Zhu et al. [335] proposed a late fusion 3D CNN-based approach, which relied on C3D [275] and fused RGB and depth. Duan et al. [54] introduced a convolutional two-stream consensus voting network (2SCVN) and 3D depth-saliency CNN stream (3DDSN), both running simultaneously. Wang et al. [301] fused RGB and depth and novel representation, namely "scene flow." Wang et al. [305] trained a CNN using both RGB and depth information. Rahmani et al. [220] presented used depth and skeleton data, aimed to capture interactions among body parts and/or objects and incorporated a CNN for classification.

2. **RNN-based approaches**: Pigou et al. [216] employed a CNN to extract frame-based appearance features incorporating depth information, and an RNN to capture temporal motion dynamics. Li et al. [161] utilized the C3D model [275] to extract RGB and depth features, separately. Upon fusion, they were used as input to an SVM classifier. Zhu et al. [336] also used C3D [275] and convolutional LSTMs [240]. The role of the CNN was to mitigate limitations of LSTM. Luo et al. [173] adopted an RNN-based encoder-decoder to learn video features by predicting a sequence of basic motions.Shi et al. [242] fused depth and skeleton information using a Privileged Information-based RNN (PRNN). In the same context, Mahassein and Todorovic [175] introduced a regularization of LSTM learning where the output of another encoder LSTM (eLSTM) grounded on 3D human-skeleton training data is used as regularization, under the assumption since videos and skeleton sequences involve human motions, similar features will be extracted from both.

3. **Approaches based on other architectures**: Shahroudy et al. [236] fused hand-crafted features with RGB and depth information, by embedding them into a common space. For classification they used stacked layers of non-linear autoencoder-based component factorization, forming a deep shared-specific analysis network. Wu et al. [310] utilized a Gaussian-Bernoulli Deep Belief Network (DBN) to extract high-level skeletal joint features. They also employed a 3D CNN to extract features depth and RGB and investigated several fusion strategies in combination with temporal modeling, demonstrating that fusion could lead to significant improvement of recognition.

1.4.4.2 Continuous RGB+D-Based Motion Recognition

Methodologies in this category are typically RNN-based approaches. Chai et al. [34] introduced a fusion approach for RGB and depth in a two-stream RNN, wherein depth and RGB fusion takes place prior to the LSTM layer. Molchanov et al. [188] proposed joint segmentation and classification, utilizing a recurrent C3D with connectionist temporal classification (CTC) and training a separate network for each modality. Notably, they incorporated the infrared modality within their framework. Recognition was a result of late classifier fusion.

1.5 Research Questions

In the proposed Thesis, new methodologies for HAR, implementing architectures based on deep learning techniques will be explored and evaluated. They will rely on the measurement of physiological and kinematic parameters, as well as on the extraction of visual data recorded by sensors placed in the subjects' environment. Emphasis will be given on visual representations of the recorded data, allowing recognition based on convolutional neural network architectures. Techniques for information fusion will also be investigated.

The goal of this Thesis is to contribute to the field of HAR by introducing innovative approaches that leverage deep learning techniques for recognizing human behavior and activities. Additionally, by focusing on low-cost sensors, the proposed methodologies may potentially be more easily applicable in real-world scenarios.

Building upon the foundational context established in this Chapter, this subsection delineates the specific research questions that direct the focus of this Thesis, which are as follows:

1. How can well-known spectral image transformations of raw signal data image representations of 3D skeletal motion may be used for human activity recognition, and which transformations are most effective in achieving this goal? (Chapter 3);
2. Is it possible to use artificially rotated skeleton sequences as a means for data augmentation, so as to improve the classification accuracy? What is the effect to classification accuracy of alignment of a given skeleton with the viewpoint used for training? (Chapter 4);
3. Is it possible to use a domain adaptation approach to align feature distributions originating from different camera viewpoints? (Chapter 5);
4. Can fusion of multiple modalities of skeletal representation assist to improve performance and at what extent? (Chapter 6);
5. Will the augmentation of training set using occluded samples provide an increase to the performance of a HAR approach and to what extent? (Chapter 7).

In the following subsections, we will outline each specific research question, providing insights into how they align with the research aims and how they will be operationalized within the methodological framework of this study. The exploration of these questions is intended to yield a comprehensive understanding of the topic, fostering advancements in theory and practical applications of human activity recognition.

1.6 Contributions of this Thesis

At the following we summarize the main contributions of this Thesis to the research area of HAR:

1. a 2D image representation of 3D skeletal motion, which is based on well-known 2D image transforms (Chapter 3);
2. a data augmentation and view alignment approach, which is based on the 3D rotation geometric transform (Chapter 4);
3. an adversarial domain adaptation approach, applied on cross-view HAR evaluation (Chapter 5);
4. a multimodal fusion approach, combining RGB and skeletal data modalities;
5. and a data augmentation approach which exploits artificially occluded skeleton action samples.

2. THEORETICAL AND EXPERIMENTAL BACKGROUND

2.1 Artificial Intelligence

Artificial Intelligence (AI) is the field of Computer Science that deals with the design and implementation of programs capable of mimicking human cognitive abilities, thus exhibiting features typically attributed to human behavior. These features include problem-solving, visual perception, learning, drawing conclusions, and understanding natural language. As illustrated in Fig. 2.1, AI is a huge research area, wherein Machine Learning (ML) (see subsection 2.1.1) is a sub-area of AI, while Deep Learning (DL) (see subsection 2.1.3) is a sub-area of ML.

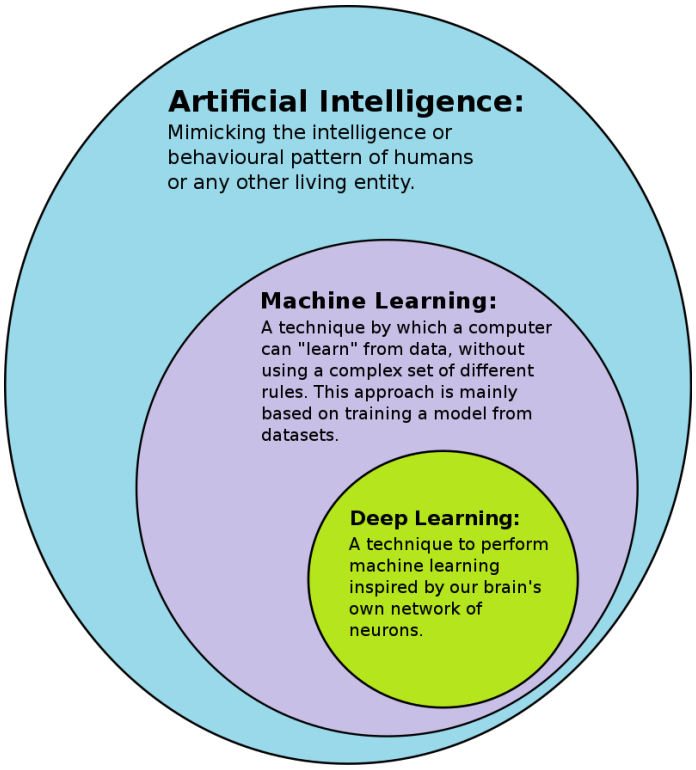


Figure 2.1: A Venn diagram, displaying how deep learning is a subset of machine learning and how machine learning is a subset of artificial intelligence. Figure taken from [222]

A popular definition of AI is provided by Barr and Feigenbaum [13]: “AI is the field of computer science that deals with the design of intelligent (cognitive) computational systems, namely systems that exhibit characteristics associated with intelligence in human behavior.”

However, there exist several other definitions according to which the aim of AI is to build systems that encompass the following objectives:

1. **Think like humans:** AI aims to create systems capable of performing thinking func-

tions in a manner similar to that of humans.

2. **Behave like humans:** The goal is to develop systems that exhibit behavior similar to that of humans in various situations.

3. **Think logically:** AI seeks to design systems capable of making logical decisions and engaging in thinking based on logic.

4. **React logically:** AI seeks to create systems that can react to situations based on logic, similar to how humans react.

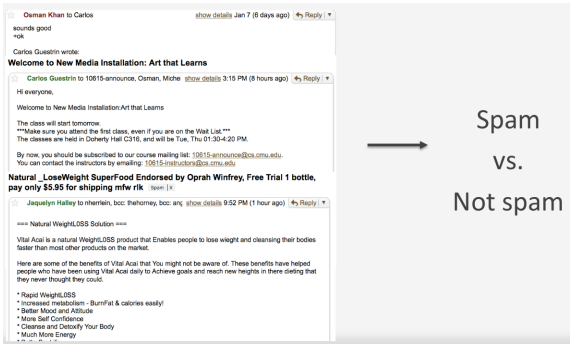
At the following, we attempt to present a brief historical overview of AI. The beginning of AI era is probably 1943, i.e., when McCulloch and Pitts [181] proposed artificial neurons (i.e., the “perceptrons”), demonstrating how they can implement simple logical functions. Minsky [196], in 1951, constructed the SNARC, the first machine using neural networks with 40 neurons. In the same year, the first AI games, i.e., chess and checkers, were developed. In 1955, the first computer program capable of proving theorems from Russell’s work was created, while in 1956, the first AI conference took place at Dartmouth, attended by key figures in the field. The 1960s saw the creation of the first robot, but the initial enthusiasm of the 1960s was followed by criticism, suggesting that systems were only suitable for simple “toy problems.” In the 1970s, systems were developed containing the necessary knowledge to behave like human experts in various fields, called “expert systems” or “knowledge Systems.”

In the early 1970s, Prolog programming language was introduced [148]. However, due to limitations such as computational power, it was impractical to solve large-scale problems. Moravec’s paradox [2] highlighted the contrast between simple tasks for computers and complex tasks for humans, like facial recognition or obstacle avoidance. Computational vision and robotics faced challenges during that time.

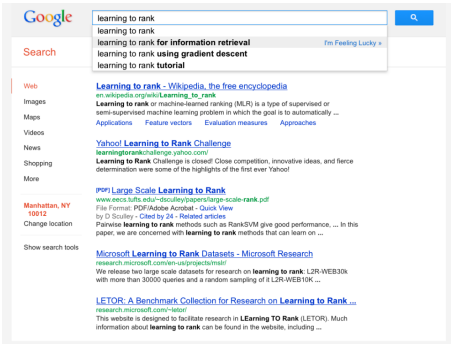
The late 1980s brought an explosion of expert systems [60], micro-level specialized knowledge for specific problems, leading to research funding. The revolution of knowledge emphasized that intelligent behavior depends on accumulated knowledge. Major breakthroughs during the 1980s in the area of machine learning were the Hopfield Networks [94] and the Backpropagation algorithm [146]. Support Vector Machines (SVMs) for use in classification and regression tasks were reintroduced by Vapnik in 1992 [282]. In 1997, Deep Blue defeated the world chess champion, Garry Kasparov, marking a breakthrough in computing power [200].

The rise of Deep Learning [78] since the early 2010s has been driven by the availability of vast amounts of data and powerful computational resources, such as GPUs, which are adept at handling the complex matrix operations involved in neural networks. Deep Learning is a branch of machine learning, modeling high-level abstractions using multiple processing layers, thus being capable of discovering intricate structures in large data sets. This approach has led to breakthroughs in tasks that once seemed nearly impossible for machines to perform, like translating texts, recognizing spoken words, and classifying images with an accuracy comparable to human performance.

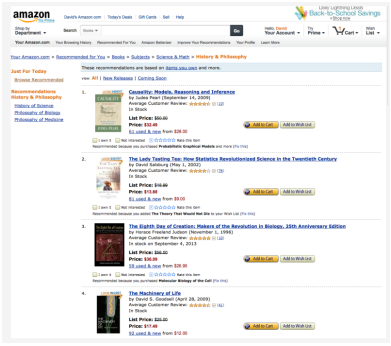
Within the context of contemporary applications, AI has found multifaceted utility across various domains (see Fig. 2.2). A prominent application is the efficient identification of unwanted or spam emails, where AI algorithms employ advanced pattern recognition techniques to filter and categorize incoming messages (Fig. 2.2a). AI has also revolutionized internet search engines (Fig. 2.2b), enhancing the accuracy and relevance of search results through sophisticated algorithms that understand user intent. Moreover, in the realm of e-commerce, AI plays a crucial role in providing personalized shopping recommendations (Fig. 2.2c) by analyzing user preferences and behavior. Additionally, in the financial sector, AI algorithms are employed for predicting stock prices (Fig. 2.2d), leveraging complex data patterns and market trends. Image recognition capabilities of AI come to the forefront in identifying objects and areas within photos (Fig. 2.2e), contributing to advancements in fields such as computer vision. Facial recognition (Fig. 2.2f) is another noteworthy application, having implications in security and authentication systems. The integration of AI into digital assistants enhances their capabilities, allowing for more natural language understanding and personalized interactions (Fig. 2.2g). Furthermore, the pursuit of autonomous driving (Fig. 2.2h) has seen significant strides with the application of AI in developing advanced driver assistance systems (ADAS) and self-driving technologies, aiming to redefine the future of transportation.



(a)



(b)



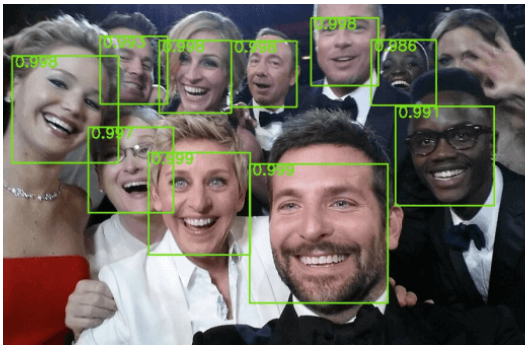
(c)



(d)



(e)



(f)



(g)



(h)

Figure 2.2: Several modern applications of AI, including (a) spam filtering; (b) search engines; (c) visual recognition; (d) stock price prediction; (e) semantic segmentation; (f) face detection; (g) home assistants; (h) autonomous driving.

2.1.1 Machine Learning

Machine Learning (ML) [273] stands as one of the most significant domains within the field of artificial intelligence. It involves algorithms that "learn" and improve through experience, utilizing data. The approach followed in ML is to construct a model using a dataset referred to as the "training set." The effectiveness of these models is evaluated using a set of data known as the "testing set." ML closely intersects with statistics, as both fields are related with the analysis of data. The process entails refining algorithms based on experience, emphasizing the iterative nature of machine learning's development. The core concept revolves around constructing models that can generalize patterns from the training data to make predictions or decisions on new, unseen data. At the following we will present the main approaches used for making models learn by using data.

- *Supervised Learning*: This type of learning involves the construction of a function that maps given inputs to known, desired outputs originating from a training set. The ultimate goal is to generalize this function for inputs with unknown outputs, typically assessed using a testing set. Examples of supervised learning tasks include *classification*, where the algorithm categorizes inputs, and *regression*, where it predicts continuous outputs based on input features.

- *Unsupervised Learning*: In this type of learning, algorithms build models for a set of inputs without knowledge of desired outputs for the training set. Common applications include *clustering*, where the algorithm groups similar data points, and *dimensionality reduction*, aiming to simplify the representation of input data.

- *Semi-Supervised Learning*: Semi-supervised learning combines a small number of labeled data with a larger set of unlabeled data. Positioned between supervised and unsupervised learning, this approach aims to leverage the benefits of both, addressing scenarios where obtaining labeled data is resource-intensive.

- *Reinforcement Learning*: Reinforcement learning focuses on algorithms learning through direct interaction with an environment. Inspired by learning through "reward and punishment," the goal is to maximize a reward function. The model autonomously discovers which actions yield the greatest benefit, making it suitable for dynamic environments where outcomes depend on sequential decision-making.

The aforementioned learning paradigms form the basis of machine learning, a crucial field within the broader landscape of AI. The intersection of algorithms and data analysis defines the methodologies employed for developing intelligent systems. Modern applications of machine learning primarily target:

- *Data Classification*: ML is extensively applied for categorizing and classifying data into distinct groups or classes. This involves training models to recognize patterns and make predictions based on input features.

- *Predictions of Future Behavior*: ML excels in making predictions about future behavior based on historical data. These predictions can range from forecasting financial trends to predicting user preferences in recommendation systems. ML, especially when imbued with a form of "intelligence," is considered a subset of AI. It is closely associated with the field of data mining, although there is a fundamental difference:
  - *Machine Learning*: Relies on *known* properties of the data. Aims to construct models that generalize based on identified patterns in the data.
  - *Data Mining*: Attempts to discover *unknown* properties of the data. Focuses on uncovering hidden patterns and relationships within the data.

Furthermore, ML is also linked to the area of optimization. In many problems, the objective is to minimize or maximize the value of a specific function. ML techniques are often employed to optimize parameters and improve the performance of models in various applications. Machine learning is also closely connected to the concept of generalization. While optimization algorithms aim to minimize errors within a specific training set, machine learning algorithms strive to minimize errors on examples they encounter for the first time. This implies the ability of learning models to generalize and adapt to new data beyond the training set. Moreover, ML is tightly linked to the field of statistics in terms of methods. However, there is a difference in their objectives. While statistics focuses on data analysis, machine learning seeks to learn patterns that "generalize" to new, optimistic data. This ability to generalize makes ML particularly powerful in handling and predicting outcomes in diverse environments and conditions.

2.1.2 Neural Networks

Neural networks (NNs) [89] are a fundamental component of both AI and ML, designed to emulate the human brain's ability to learn and make decisions. Specifically, NNs are networks (graphs) whose nodes consist of artificial neurons, while their edges contain weights. The weights are calculated through a learning process. All inputs are modified by these weights and summed to produce the output. NNs find applications in various domains and can be broadly categorized into the following:

- *Function Approximation/Regression*: NNs are adept at approximating and modeling functions. They can learn complex relationships between inputs and outputs, making them suitable for regression tasks.
- *Classification/Pattern Recognition*: Another significant application is in classification tasks where the network learns to recognize and categorize patterns or sequences. This includes image and speech recognition, as well as identifying patterns in various datasets.
- *Data Processing (Filtering, Compression, etc.)*: NNs may be employed for data processing tasks, such as filtering and compression. They learn to extract relevant features from data, enabling efficient processing and analysis.

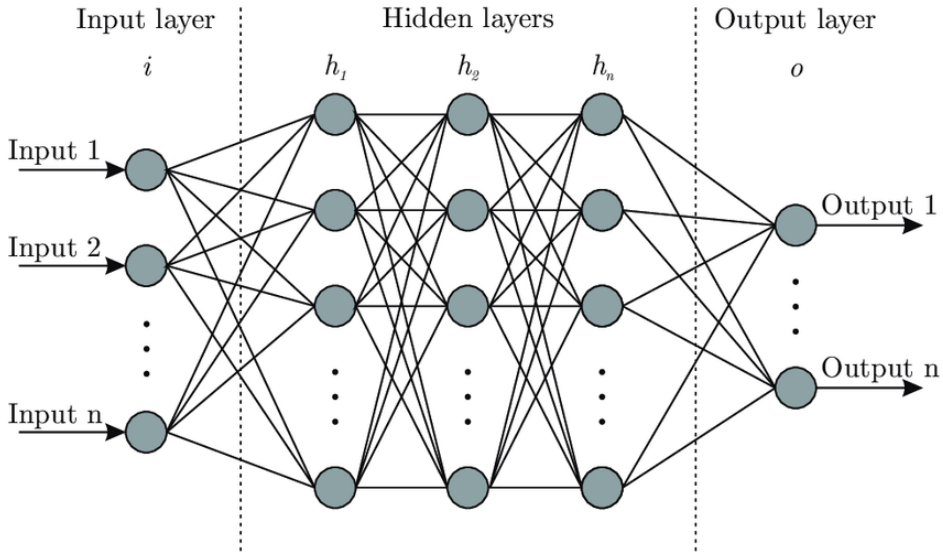


Figure 2.3: The layers of a feed-forward neural network. Figure taken from [232]

These categories showcase the versatility of NNs, making them powerful tools in solving a wide range of problems across different domains. The ability to learn from data and adapt to complex patterns contributes to their effectiveness in various applications.

Neural networks consist of interconnected nodes, known as artificial neurons, organized into layers, including the input layer, hidden layers, and an output layer (as illustrated in Fig. 2.3). The architecture involves (a) neurons in the input layer receiving initial data; (b) hidden layers processing intermediate patterns, and (c) the output layer producing final predictions. The learning process of neural networks involves training, during which they adapt their parameters according to input data and the expected output. This adjustment is guided by the evaluation of the network's performance, utilizing a loss function to quantify the disparity between predicted and actual outcomes. This iterative training loop allows neural networks to refine their internal representations and enhance their ability to make accurate predictions over time.

In the context of neural networks, the activation function plays a crucial role: neurons within a neural network employ activation functions to introduce non-linearities, enabling the network to grasp intricate relationships within the data. Several commonly used activation functions include the sigmoid function, hyperbolic tangent (tanh), and rectified linear unit (ReLU). These functions contribute to the network's capacity to capture and represent complex patterns and non-linear dependencies in the input data, enhancing its learning capabilities.

NNs undergo training using optimization algorithms like gradient descent, adjusting weights and biases iteratively to minimize the loss function, measuring the disparity between predicted and actual outcomes. This optimization technique updates network parameters opposite to the loss function gradient until a minimum point is reached. Weight and bias adjustment occurs during training to enhance the network's predictive accuracy on new, unseen data. Techniques such as dropout and batch normalization are employed to pre-

vent overfitting, ensuring the network generalizes effectively. These processes are pivotal for neural networks to generalize well and deliver reliable predictions on diverse datasets.

Neural networks have diverse applications across multiple domains, demonstrating their versatility in solving complex problems. They play a crucial role in image and speech recognition, natural language processing, autonomous vehicles, healthcare diagnostics, and more. Ongoing research is dedicated to improving interpretability and addressing these challenges, with a growing emphasis on ethical considerations to ensure responsible and accountable deployment of artificial intelligence systems. Finally, neural networks have become a cornerstone of modern AI, continually evolving and contributing to advancements in various fields.

2.1.3 Deep Learning

Deep learning [147] is a subset of machine learning and stands as a transformative approach in the realm of AI. It harnesses the power of NNs, which are structured in layers, to process vast amounts of data for tasks such as image recognition, natural language processing, and predictive analytics. Unlike traditional machine learning methods that require explicit programming for feature extraction, deep learning models are adept at *learning* the necessary features to make accurate predictions or classifications from raw data. This capability allows deep models to achieve remarkable accuracy, often surpassing human-level performance in tasks such as facial, voice and object recognition.

Deep learning models are primarily characterized by their deep structures, which include multiple layers of neurons, each responsible for transforming its input data into progressively more refined and abstract representations. The term “deep” refers to the number of layers through which data is processed, making the architecture significantly more profound than traditional NNs. Each layer of neurons builds on the features recognized by the previous layer, enhancing the network’s ability to learn complex patterns in large datasets. This learning process is fine-tuned through back-propagation, an essential mechanism in training deep learning models. Back-propagation efficiently calculates the gradient of the network’s error with respect to its adjustable parameters, thereby guiding the network to optimize these parameters through methods such as gradient descent. This iterative adjustment is central to deep learning’s ability to refine its predictions over time, thereby increasing its decision-making accuracy.

At the following, we briefly present the main DL architectures that are used are designed to handle specific types of data, each being equipped with unique structures and operational methodologies, so as to optimize its performance for its intended applications. Specifically, these architectures are:

1. **Convolutional Neural Networks** (CNNs) [144] are primarily tailored for processing grid-like data, such as images, employing convolutional layers that deploy a series of learnable filters across the input, aiming at the identification of spatial hierarchies by capturing fundamental features like edges and colors in the initial layers and then

progressing to more complex features such as parts of objects in deeper layers. This capability makes CNNs extremely effective for tasks involving image recognition and video analysis, where understanding spatial relationships is crucial. Renowned CNN models such as LeNet [145], AlexNet [133], VGG [246], and ResNet [90] have progressively enhanced how deep networks manage large-scale image recognition tasks.

2. **Recurrent Neural Networks** (RNNs) [182] are designed to handle sequential data like time series or natural language. They process inputs through loops, allowing information to persist from one step of the sequence to the next, which is vital for continuity in understanding sequences. This makes RNNs particularly suitable for speech recognition, language modeling, and translation. However, RNNs typically face challenges with long-range dependencies within input sequences due to issues such as vanishing and exploding gradients. To address these issues, variants such as Long Short-Term Memory (LSTM) units [93] and Gated Recurrent Units (GRUs) [69] have been developed. These models incorporate mechanisms to effectively retain information over longer sequences, enhancing the network's ability to learn from data with extended dependencies.

3. **Autoencoders** [92] are unsupervised learning models that aim to learn a compressed, distributed representation of a dataset, typically for dimensionality reduction or feature learning. Structured with an encoder to compress the input and a decoder to attempt its reconstruction, autoencoders are used not just for reducing data dimensions but also for applications like de-noising images or detecting anomalies in data streams.

4. **Generative Adversarial Networks** (GANs) [40] feature two neural networks—the generator and the discriminator—that engage in a sort of competition. The generator creates new data instances, while the discriminator evaluates them against actual data. This setup helps refine the generator's output continually until the generated data is indistinguishable from real data, making GANs highly effective for producing realistic synthetic images and for applications in video generation and unsupervised learning.

5. **Transformers** [283] focus in advancing natural language processing, utilize self-attention mechanisms to determine the relevance of different words in a sentence, irrespective of their positions. This architecture has been fundamental in improving performance in tasks like text summarization, translation, and sentiment analysis. Notable examples such as Google's BERT [47] and OpenAI's GPT series [170] have been based on transformers, setting new benchmarks in various NLP tasks.

These deep learning architectures showcase the field's ability to adapt and evolve, extracting complex patterns and features from massive datasets to meet the demands of specific applications. This adaptive capability underlies the extensive success and ongoing development of deep learning technologies across multiple sectors. The practical

applications of deep learning are vast and continuously growing. In the healthcare sector, deep learning algorithms play a crucial role in enhancing diagnostic procedures by providing more accurate interpretations of imaging data than traditional methods, which can lead to earlier and more precise diagnoses. In the automotive industry, deep learning is at the heart of autonomous vehicle technology, enabling cars to interpret and react to complex environments in real time, thus paving the way for safer and more efficient transportation systems. Additionally, in media and entertainment, deep learning techniques are used to create advanced special effects and animations, offering viewers more immersive experiences. As the availability of computational resources and data continues to increase, the potential for deep learning to drive significant advancements across various industries is immense. This expansion is expected to foster innovative applications and efficiencies, revolutionizing how we interact with technology on a daily basis.

2.1.4 Convolutional Neural Networks

The area of computer vision has significantly benefited from the use of deep learning architectures. The dominant deep architecture is undoubtedly the Convolutional Neural Network (CNN) [145]. The latter, resembles to traditional neural networks and has almost eliminated the need of extracted handcrafted features for the description of low-level visual properties. This is achieved by its key component, i.e., the *convolutional* layers. Their role is to learn a set of convolutional filters. The dimension of these filters is rather small (i.e., a few pixels). They are formed by grouping neurons in a rectangular grid and slide across the whole image during the forward pass of the algorithm. This way, the responses, when a filter is centered at any pixel 2.4, produce a 2D activation map (Fig. 2.5). Training takes place as with every other NN; a forward propagation of data and a backward propagation of error do take place to update weights. Ultimately, the filters learnt by the network will activate when they encounter certain types of visual features.

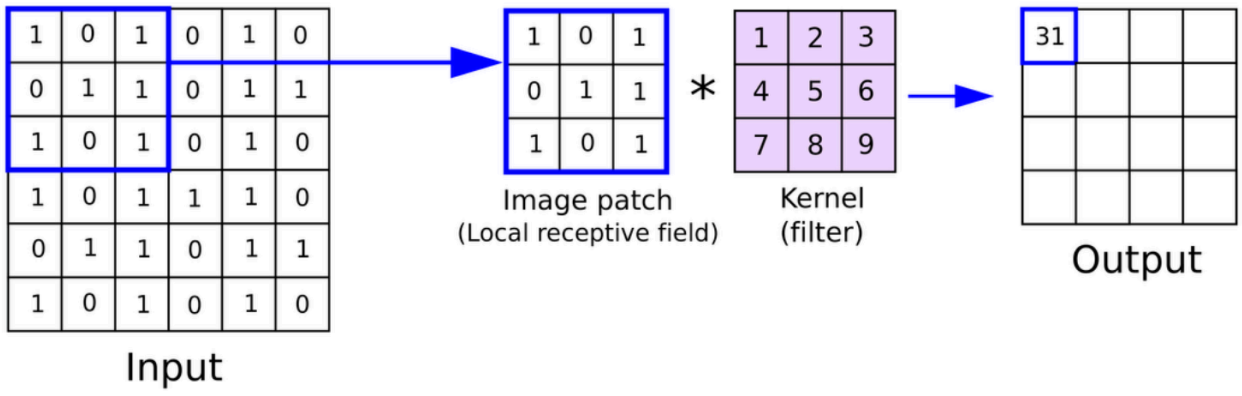


Figure 2.4: A visual example of convolution. Figure taken from [226].

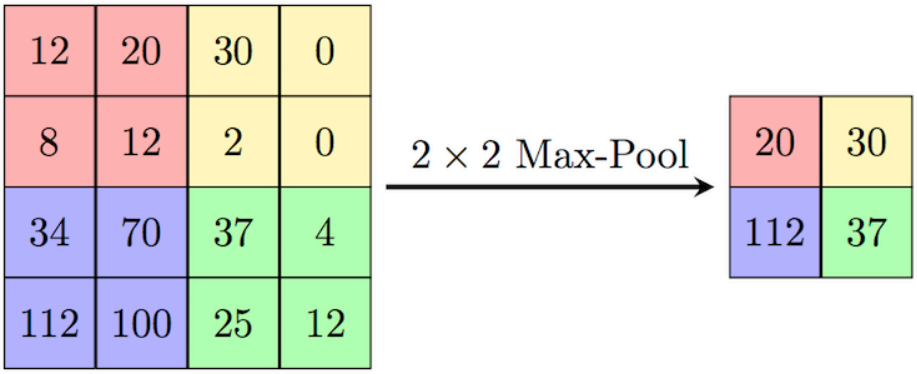


Figure 2.6: A visual example of Max Pooling. Figure taken from [180].

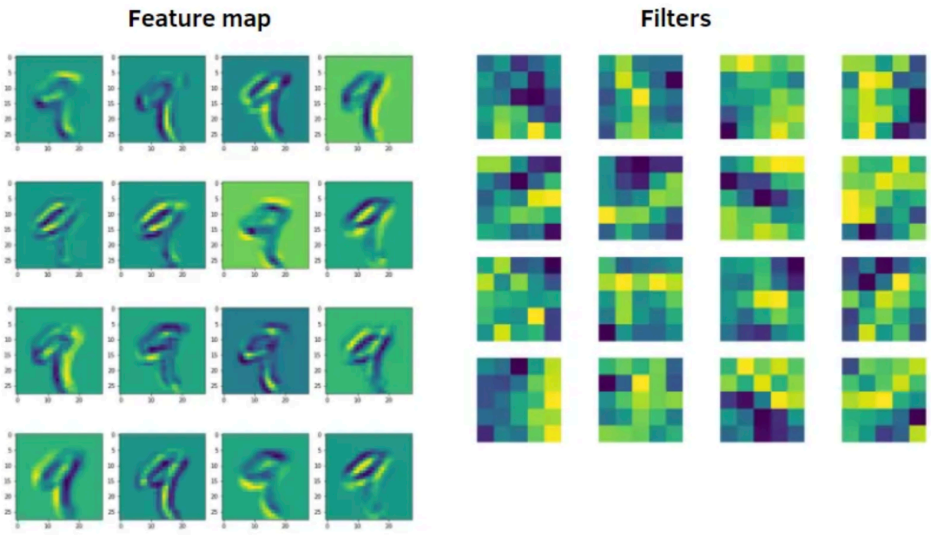


Figure 2.5: Example filters that have been learned by a CNN and corresponding feature maps. Figure taken from [6].

Pooling layers are usually placed between single or sets of serial or parallel convolutional layers. Their role is to progressively reduce the representation size. To achieve this, they take small rectangular blocks from the convolutional layer. By subsampling them, they end up to produce a single output from each block (Fig. 2.6). This way, a step towards reducing the complexity of the network and controlling overfitting is performed. Finally, in *dense* layers (also known as "fully-connected" layers) each node is connected to all activations of their previous layer (Fig. 2.7) and their role is to perform classification based on the extracted features by the convolutional layers (which may have been subsampled by pooling layers). A visual example of a CNN comprising all the aforementioned types of layers is illustrated in Fig. 2.7.

The architecture of a CNN resembles to the one of a traditional neural network (NN), however, its goal is to learn a set of convolutional filters. The *convolutional* layers are those that play the key role in the whole process. Their neurons are grouped in rectangular grids,

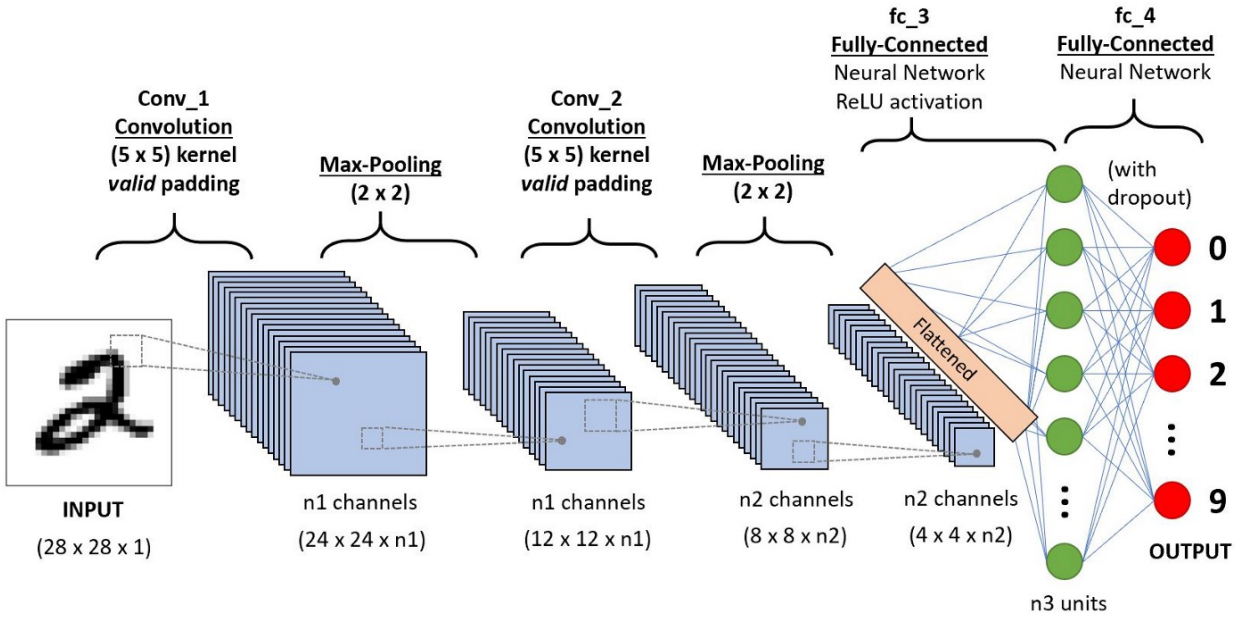


Figure 2.7: A convolutional neural network, comprising convolutional, pooling and fully-connected layers. Figure taken from [238].

so that each would perform a convolution in a part of the input image. Learning process aims to learn the parameters of this convolution. Moreover, it is the use of multiple layers which non-linearly processing the network's input, that renders CNNs able to "learn" to extract features. Specifically, the output of each layer is fed to the next layer and ultimately, they become able to learn multiple levels of representations which correspond to multiple levels of abstraction.

Finally, within our approach, we also adopt the widely used *dropout* regularization technique [256]. Its goal is to reduce overfitting by preventing complex co-adaptations on training data, i.e., dependencies among neighboring neurons. To achieve this, several random neurons are ignored ("dropped-out") during training and also their weights are not updated. A visual example of dropout is illustrated in Fig. 2.8.

2.2 Neural Network Training Techniques

One of the most common and serious problems when trying to train a supervised learning model is the lack of a sufficient amount of labeled data. When labeled data is scarce, the generalization capabilities of the produced model are severely affected; the quality of the produced results as well as model evaluation are degraded. More specifically, for several tasks of practical and research interest within the broader area of computer vision, e.g., image/video classification, the collection of an adequate number of labelled data is either infeasible or too costly. Moreover, as demonstrated in [261], for several tasks,

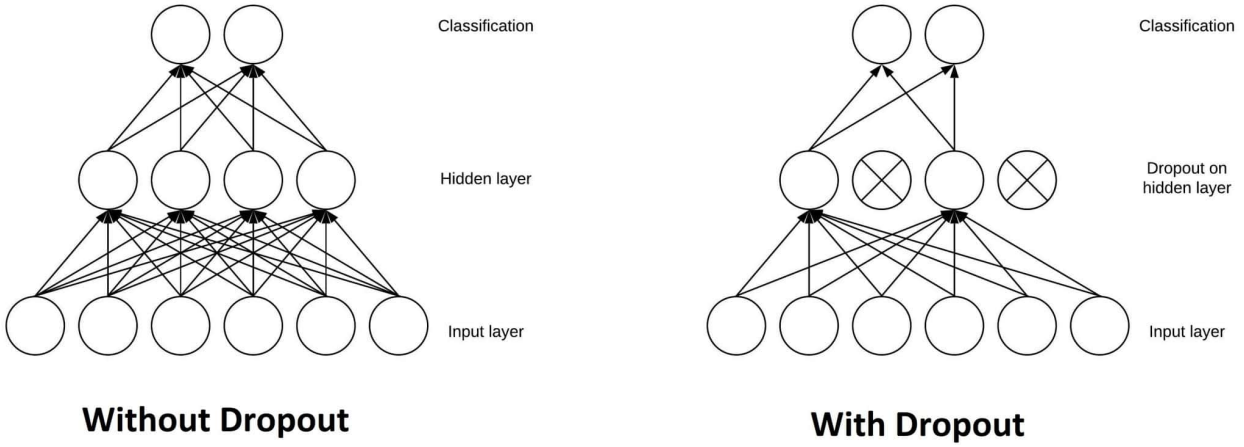


Figure 2.8: An example of deactivation of neurons using the dropout technique. Figure taken from [95].

the performance of models may only increase logarithmically with increasing volume of available training data. For these reasons, much research has been recently devoted to the construction of methods that are robust against insufficient labeled data. In particular, two classes of methods have been widely adopted to deal with the aforementioned issue, namely: *data augmentation* and *domain adaptation*.

2.2.1 Data augmentation

Data augmentation [280] is a vital technique in the field of machine learning and data pre-processing. It involves the manipulation of training data by applying various transformations, perturbations, or modifications to generate new instances that are *similar but not identical* to the original data. The primary goal of data augmentation is to expand the training dataset, increase its diversity, and improve the robustness of machine learning models. This process is especially valuable when dealing with limited training data or to prevent overfitting, where a model memorizes the training set rather than learning to generalize from it.

One of the most common applications of data augmentation is in computer vision and particularly in image classification tasks [184, 214]. For example, considering a dataset comprising images of cats, data augmentation techniques can be used to create new training examples by applying transformations such as rotation, scaling, flipping, and changes in brightness and contrast to the original images (See Fig. 2.9). By introducing these variations, the model learns to recognize cats from different angles, under varying lighting conditions, and in diverse orientations. This, not only increases the size of the training set but also improves the model's ability to handle real-world variations in images.

In the field of natural language processing (NLP), data augmentation is also relevant [61]. For text data, augmentations include synonym replacement, paraphrasing, word shuffling, and text summarization. These augmentations are able to assist NLP models learn to

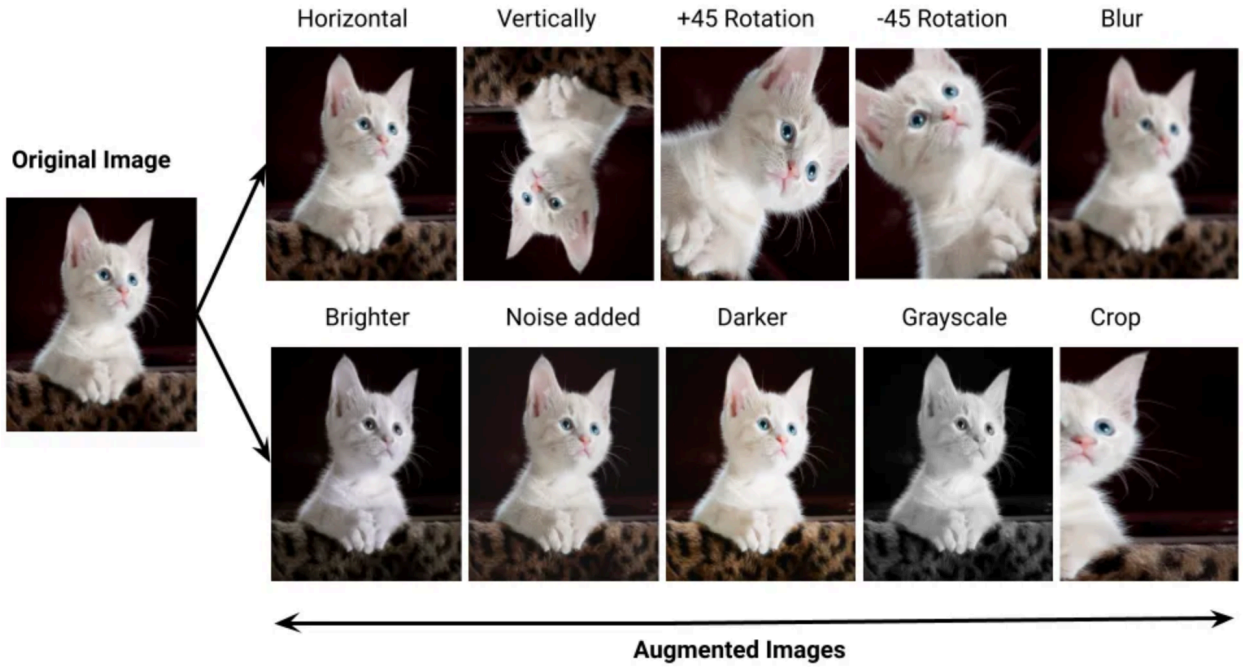


Figure 2.9: Several data augmentation techniques applied to the problem of image classification. Figure taken from [197]

understand and generate diverse language patterns. In audio processing, data augmentation techniques can involve pitch shifting, time stretching, and adding noise to audio samples [308]. This diversifies the acoustic data and helps models to better recognize speech or other audio patterns under different acoustic conditions.

Data augmentation is a powerful technique for reducing overfitting, a common problem in machine learning [243]. Overfitting occurs when a model becomes too specialized in capturing the training data' noise and idiosyncrasies rather than learning the underlying patterns. By introducing augmented data during training, models are exposed to a broader range of examples, making them less sensitive to minor variations in the input. This results in models that generalize better to unseen data, which is essential for real-world applications. Additionally, data augmentation is useful when acquiring more labeled data is costly or time-consuming. Instead of manually collecting additional data, data augmentation allows to effectively create synthetic data, maximizing the utility of the original dataset.

In summary, data augmentation is a fundamental technique in machine learning that involves creating new training data instances by applying various transformations to the original data. It enhances the size and diversity of the training dataset, leading to more robust and generalizable machine learning models. However, a typical data augmentation strategy uses one or a combination of techniques which rely on domain knowledge [243], [280]. Note that such data synthesizers, e.g., as in the case of HAR [48, 151] are typically hard to implement, and are mostly domain-dependent and specific to a particular problem/task.

2.2.2 Domain Adaptation

Deep learning algorithms are pivotal for tackling a range of computer vision tasks, including classification, segmentation, object detection, and image retrieval. However, these powerful tools face significant challenges. Primarily, they require vast amounts of labeled data, and manual labeling is laborious and tedious. Moreover, a deep learning model performs optimally only when the test data closely mirror the training data distribution. For example, data from mobile phone cameras versus high-end cameras exhibit significant differences in distribution, which can severely impair model performance, highlighting limitations e.g., in traditional transfer learning methods.

To better understand this, let us consider a machine learning model trained to recognize objects in *indoor* scene images. It effectively identifies chairs, tables, and sofas within this controlled environment. However, using this model for outdoor scenes introduces significant challenges due to different statistical properties between indoor and *outdoor* environments. Here, domain adaptation techniques are crucial, since they are able to allow the model to adjust to new environments without extensive retraining.

Thus, whenever such a new dataset is introduced, it necessitates not only initial annotation but also re-training of the already trained model to suit this new data. Training a substantial model on large datasets like ImageNet [45] can be resource-intensive, demanding considerable computational power and time—often weeks. Re-training for each new dataset becomes impractical and often unfeasible.

Domain adaptation [250] is a technique that is crucial in machine learning, particularly within supervised and transfer learning contexts. Specifically, it involves adapting a model trained on a source domain to effectively perform in a different target domain. This adaptation is essential when the source and target domains differ significantly in their statistical properties or characteristics. A typical network structure for implementing adversarial-based domain adaptation methods might include optional generators for creating synthetic data and a domain discriminator to classify source and target domains.

A fundamental challenge in domain adaptation is the difference in data distribution between the source and target domains, often referred to as “domain shift” or “dataset shift.” Typically, models are trained assuming that training and testing data are drawn from the *same* distribution. However, when this assumption does not hold –such as in domain adaptation scenarios– the model’s performance can degrade significantly on the target domain.

The main concepts in domain adaptation include:

- **Source Domain**: This refers to the original data distribution upon which a model is trained. For instance, in the previously described scenario, the source domain could be represented using a dataset of images taken with cellphone cameras. These images provide the labeled examples that the model initially learns from.
- **Target Domain**: This is the new data distribution where the model needs to apply its learned abilities, despite not having been originally trained on it. In the previously

described scenario, the target domain consists of images captured with a high-end DSLR camera. The model, which was trained on cellphone images, should now adapt to perform accurately with the DSLR images.

- **Domain Translation**: This concept involves establishing a meaningful correspondence between two distinct domains. In the context of domain adaptation, it concerns bridging the gap between the source and target domains so that the knowledge learned from one can be effectively transferred to the other.
- **Domain Shift**: This concept refers to the variations in the statistical distribution of data across different domains. These shifts can occur between training, validation, and testing sets, or, as in the previous example, between different types of data collection devices like cellphones and DSLR cameras. A domain shift can significantly impact the performance of a model if it is not addressed properly. A special case is the covariate shift, i.e., the case where the distribution of input data (features) changes between the training environment and the test environment, but the conditional distribution of the output given the input remains the same.

Common techniques for domain adaptation include:

- **Supervised Domain Adaptation**: This method uses labeled data from the target domain to fine-tune the model, enhancing its performance in that specific domain. It is highly effective when some target domain labels are available.
- **Unsupervised Domain Adaptation**: This more challenging approach does not rely on labeled data from the target domain. Instead, it utilizes the shared knowledge between the domains, employing techniques like aligning feature distributions, using domain-invariant representations, and applying adversarial training to make the model domain-agnostic.
- **Semi-Supervised Domain Adaptation**: Combining elements of both supervised and unsupervised approaches, this method uses both labeled and unlabeled data from the source and target domains. It aims to leverage available labels while transferring knowledge from the source to adapt the model effectively.

Current domain adaptation strategies [58] may be further divided into two main categories based on their architectural depth as “shallow” and “deep:”

- **Shallow domain adaptation** approaches typically employ instance-based and feature-based methods to harmonize the distributions across domains. A common strategy involves minimizing the distance between these domains using various metrics, such as Maximum Mean Discrepancy (MMD) [81], Wasserstein distance, Correlation Alignment (CORAL) [260], Kullback-Leibler (KL) divergence [136], and Contrastive Domain Discrepancy (CDD) [118]. These measures help align the statistical properties of the source and target domains, facilitating smoother transitions in model performance.

- **Deep domain adaptation** approaches leverage the power of neural networks, including convolutional neural networks [192], autoencoders [46], and adversarial networks [278], to bridge the domain gap. These advanced techniques often integrate distance metrics into one or multiple layers of the networks. By doing so, they measure and minimize the discrepancies in feature representations between corresponding layers of networks dedicated to source and target data. This integration enables a deeper and more effective adaptation, ensuring that models maintain high performance even when applied to significantly different target domains.

Deep domain adaptation techniques may be further categorized into into discrepancy-based, reconstruction-based, and adversarial-based methods, each aiming to bridge the gap between different data distributions without the need for extensive labeled data. Specifically:

- **Discrepancy-based** methods focus on aligning the marginal distributions of the source and target domains while maintaining the conditional distributions. Examples include the Deep Adaptation Network (DAN) [172] and the Deep Transfer Network (DTN) [332]. These networks add layers specifically designed to adapt the features from the source domain for use in the target domain, allowing for effective feature transfer across different distributions.

- **Reconstruction-based** methods employ structures like autoencoders [289] to minimize reconstruction errors and learn invariant features across domains. This approach helps in aligning the domain-specific features by reconstructing the target domain data from the source domain representations, thereby enabling the model to generalize better across different domains.

- **Adversarial-based** methods draw inspiration from generative adversarial networks (GANs) [77] and focus on minimizing the distributional discrepancies through adversarial objectives. These methods utilize a dual-model architecture, where the generator aims to fool the discriminator by producing domain-invariant features. Techniques such as the use of Gradient Reversal Layers (GRL) [66] in the training process help in reversing the gradient flow, thereby aligning the domain features effectively. Moreover, advanced approaches like Multi-Adversarial Domain Adaptation (MADA) [210] consider class-wise discrepancies, allowing for a more granular adaptation that maintains class integrity across domains. A typical structure of the implementation of adversarial-based methods for domain adaptation is illustrated in Fig. 2.10.

2.2.3 Adversarial Discriminative Domain Adaptation (ADDA)

Adversarial Discriminative Domain Adaptation (ADDA) [278] is a powerful technique within the field of machine learning that addresses the challenges of domain adaptation through

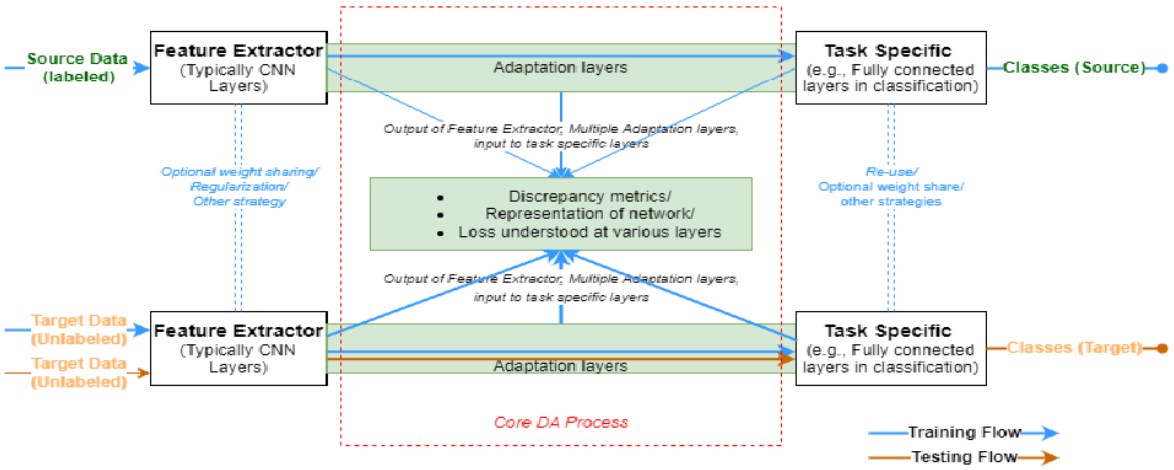


Figure 2.10: Typical network structure/architecture of deep neural networks implementing adversarial-based methods for domain adaptation. The generator(s) are optional – used only to create synthetic data if required. Domain Discriminator attempts to discriminate (classify) source and target domains, while task-specific (e.g., classification) is performed by task-specific network. Figure taken from [250]

an adversarial process. Unlike traditional domain adaptation methods that primarily focus on minimizing distance metrics between source and target domains, ADDA leverages the adversarial learning concept inspired by Generative Adversarial Networks (GANs). The primary goal of ADDA is to adapt a model trained in one domain (the source) to perform effectively in a different (target) domain, particularly when no labeled data is available in the target domain. ADDA works by employing a two-step process to achieve effective domain adaptation. First, a feature extractor and classifier are trained on the source domain to achieve high accuracy on source domain data. The second step introduces the adversarial component, where a new feature extractor is trained for the target domain, aiming to fool a domain discriminator that tries to distinguish between source and target domain features. A visual overview of ADDA is illustrated in Fig. 2.11.

Specifically, the main steps of ADDA may be described as follows:

- **Source Model Training**: in the first step, a model comprising a feature extractor and a classifier is trained on the source domain. This model is optimized to perform well on tasks like classification or detection, using labeled data from the source domain.
- **Adversarial Training**: once the source model is trained, a new feature extractor is trained for the target domain, but with a twist. This target feature extractor is trained in an adversarial way, against a domain discriminator. The discriminator's job is to classify whether features are from the source or the target domain. Conversely, the target feature extractor is trained to generate features that are indistinguishable from those produced by the source domain model. This adversarial battle encourages the target features to align with the source features, reducing domain discrepancy

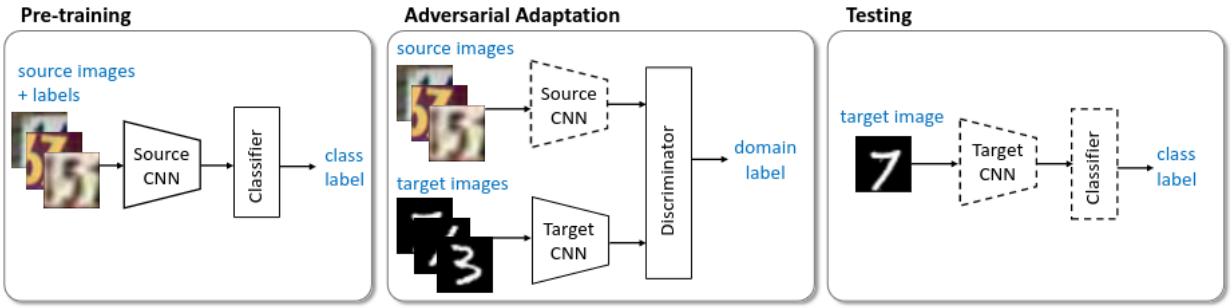


Figure 2.11: An overview Adversarial Discriminative Domain Adaptation (ADDA) approach. A source encoder CNN is firstly trained using labeled source image examples. Next, adversarial adaptation is performed by learning a target encoder CNN such that a discriminator that sees encoded source and target examples cannot reliably predict their domain label. During testing, target images are mapped with the target encoder to the shared feature space and classified by the source classifier. Dashed lines indicate fixed network parameters. Figure taken from [278]

without using any target domain labels.

ADDA offers several benefits over traditional domain adaptation methods, since it a) does not require any target domain labels, making it highly useful in unsupervised domain adaptation scenarios; b) can effectively handle substantial domain shifts by aligning feature distributions in an adversarial way, often outperforming classical methods that rely on minimizing statistical distances directly; and c) can be integrated into existing deep learning architectures with minimal modifications, making it a versatile tool for a wide range of applications.

.

2.3 Acquisition of Human Motion Data

2.3.1 The Microsoft Kinect SDK

The Microsoft Kinect sensor [333] has played a significant role in revolutionizing human-computer interaction and 3D sensing technology. It was first released in 2010 by Microsoft, initially for use with the XBOX 360 game console and within its lifetime two versions, namely Kinect v1 and Kinect v2, have been presented. In Fig. 2.12 we depict the two versions of the sensor.

This innovative technology immediately found applications in various fields, including gaming, healthcare, and robotics. In gaming, it offered immersive experiences by allowing users to control in-game characters and objects through body movements and gestures. Fig. 2.13 shows how a user interacts with the computer using Kinect. Additionally, Kinect has been utilized in healthcare for rehabilitation and telemedicine applications, leveraging



Figure 2.12: The two versions of the Kinect sensor. Left: Kinect v1; Right: Kinect v2.

its motion tracking capabilities for monitoring and therapy. Its influence extends to robotics as well, where it facilitates the development of robots capable of perceiving and interacting with their environment.

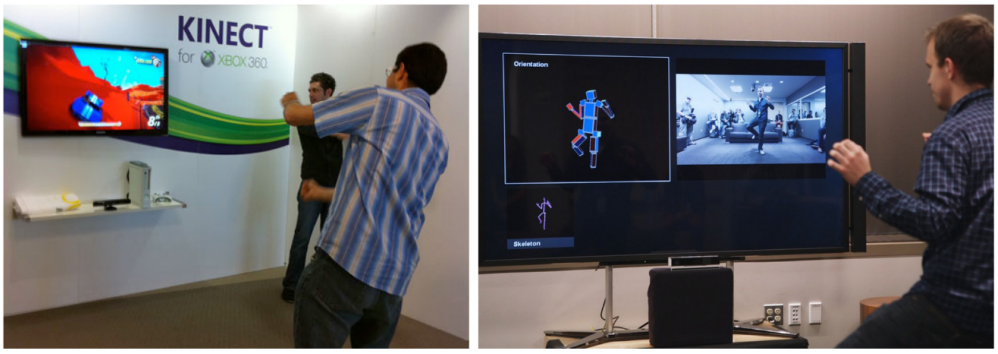


Figure 2.13: User interaction using the Kinect sensor.

In the context of this work, the Microsoft Kinect is used to capture the raw skeletal joint motion data. Both Kinect devices include RGB cameras, infrared projectors, and several other sensors (microphones etc.) enabling them primarily to capture the motion of users and secondarily to recognize their voice commands. They use structured light (v1) or time-of-flight calculations (v2) to map depth to each pixel. In Fig 2.14 we can visually identify the components of the Kinect sensor v2.

Specifically, both Kinect sensors are able to extract the following set of data modalities:

1. RGB video sequences, enabling the observation of human subjects performing various actions in standard color format;
2. depth sequences, corresponding to RGB data, and providing information regarding the depth of objects in the scene, which is essential for understanding 3D spatial relationships;
3. 3D positions of skeleton joints over time, offering valuable insights into human motion;

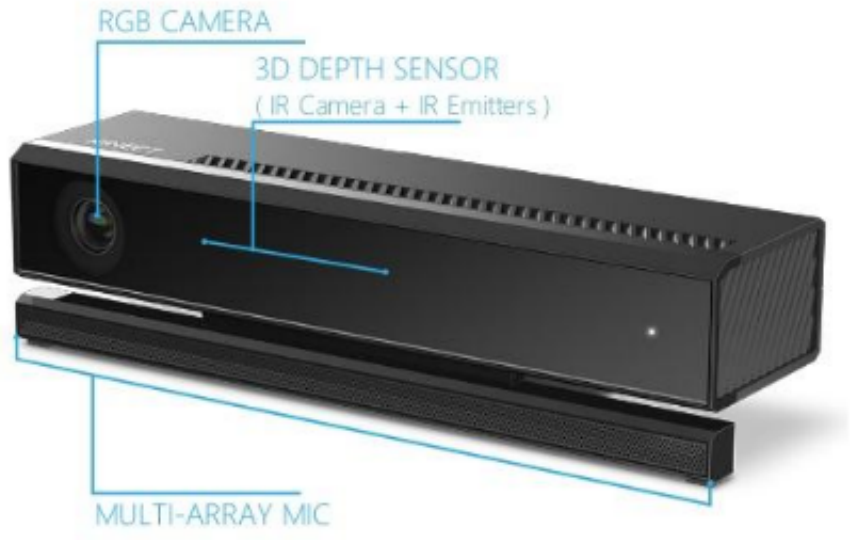


Figure 2.14: The components of the Kinect sensor include: RGB-D camera, 3D Depth camera, IR emitter and array of microphones.

4. infrared sequences sequences produced by some depth cameras having applications in depth mapping, object tracking, etc.

In Fig. 2.15 we may observe RGB images and depth images along with skeletal information (positions of skeletal joints). Note that this visual representation provides a comprehensive view of the data, facilitating a better understanding of the scene.

One of the most valuable benefits of Kinect to the research community was the ability of the extraction of the 3D positions (i.e., x, y and z coordinates) of a human's skeletal joints in real time, by using its SDK. More specifically, a structured graph of joints is continuously streamed. Graph nodes correspond to the most representing body parts (e.g., skeletal joints of arms, legs, head, etc.), while graph edges follow the structure of joints within the human body. A parent–child relationship is implied from top to bottom, i.e., *Head* is the parent of *SpineShoulder*, which is the parent of *ShoulderLeft* and *ShoulderRight*, etc. In Fig. 2.16 we illustrate the 25 human skeleton joints, that are extracted using the Kinect SDK and the v2 sensor. Moreover, in Fig. 2.17 we illustrate the differences between Kinect v1 and Kinect v2. As observed, the latter provides more comprehensive information about skeletal joints.

It is noteworthy to describe in brief the notion behind the Kinect SDK's ability to extract skeletal data by using RGB and depth information [244]. Specifically, and as indicated in Fig. 2.18 the skeleton extraction algorithm is based simple depth comparison features, wherein yellow crosses indicates the pixel x being classified. Individual features provide a weak decision regarding which part of the body the pixel belongs to, but in combination they are sufficient to accurately disambiguate all trained parts [244]. Also, in Fig. 2.18,

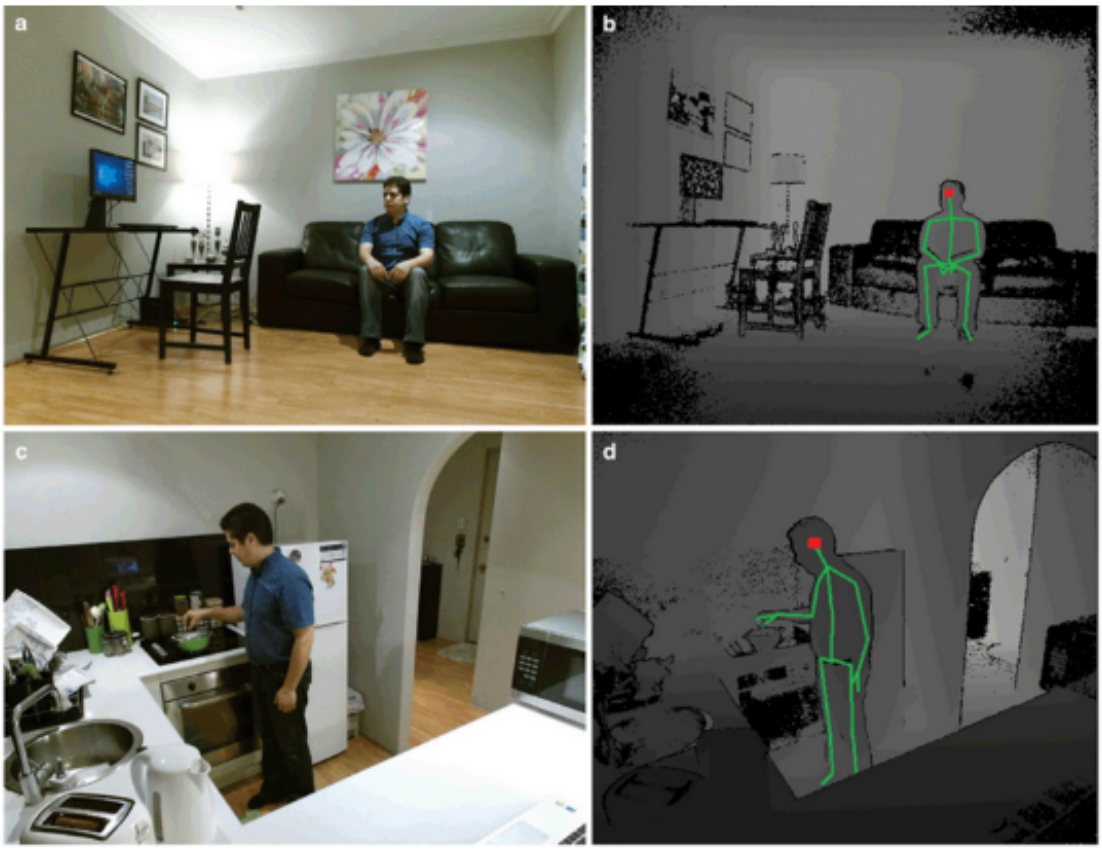


Figure 2.15: Left column: RGB video frames extracted of sequences of a person performing some activities; Right column: the corresponding depth frames with overlaid skeleton sequences.

red circles indicate the offset pixels as defined by:

$$f_\theta(I, \mathbf{x}) = d_I\left(\mathbf{x} + \frac{\mathbf{u}}{d_I(\mathbf{x})}\right) - d_I\left(\mathbf{x} + \frac{\mathbf{v}}{d_I(\mathbf{x})}\right) \ . \tag{2.1}$$

Upon observing Fig. 2.18 and specifically in (a) we may note that the two example features give a large depth difference response, while in (b), the same two features at new image locations give a much smaller response. Finally, for classification, randomized decision forests are used, as also illustrated in Fig. 2.18. Note that a forest is an ensemble of decision trees, each consisting of split nodes and leaf nodes.

It is also interesting to herein mention, that randomized decision forests are trained using synthetic data (Fig. 2.19). Since the human body is capable of an enormous range of poses, this is significantly difficult to simulate. Therefore, a large database of motion capture (mocap) of human actions, consisting of approx. 500K frames was created so as to span the wide variety of poses people would make in an entertainment scenario. Since the classifier would somewhat generalize to unseen poses, all possible combinations of the different limbs were not recorded; in practice, a wide range of poses had been

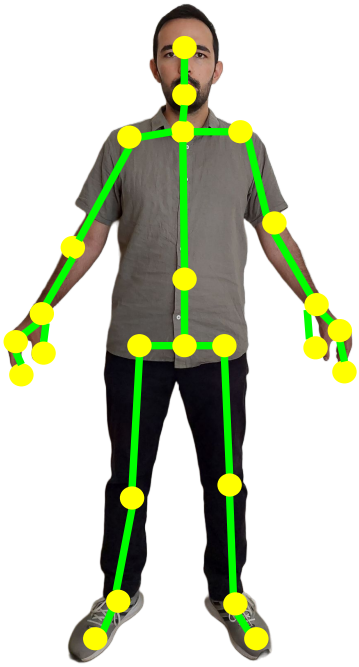


Figure 2.16: Extracted human skeleton using the Kinect v2. Yellow circles indicate joints, while green lines indicate their parent-child relationships within the human body.

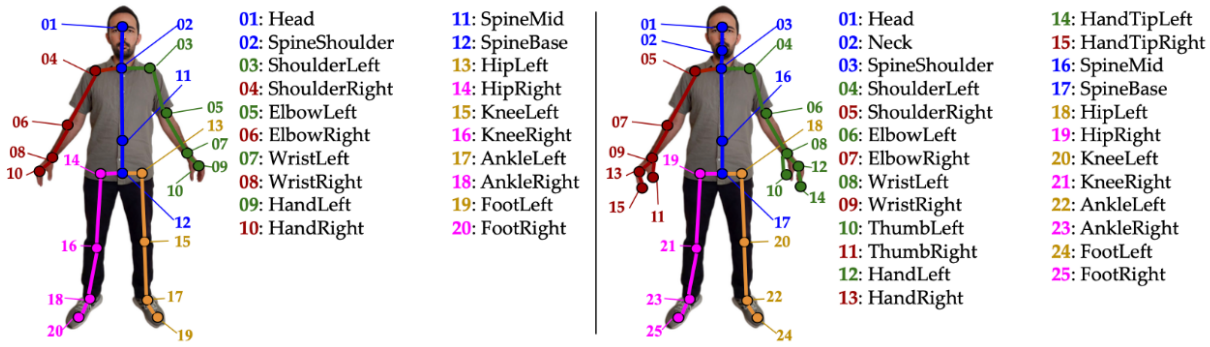


Figure 2.17: Extracted human skeletons using the Kinect v1 (left) and v2 (right). In both cases joints have been grouped to form body parts, i.e., left arm (green), right arm (red), torso (blue), left leg (orange) and right leg (magenta). Figure taken from [287].

proved sufficient. Moreover, variations in rotation about the vertical axis, mirroring left-right, scene position, body shape and size, or camera pose, had been semi-automatically added. Finally, and since the classifier did not utilize any temporal information, focus was given only in static poses and not moving skeletons. The whole process of the approach followed for the extraction of skeleton joints, uses as input depth images to firstly extract body parts and then 3D joint proposals, ultimately leading to the identification of joints, while it may be seen in Fig. 2.20.

While Microsoft has discontinued the production of Kinect v1 and v2 devices, it has pre-

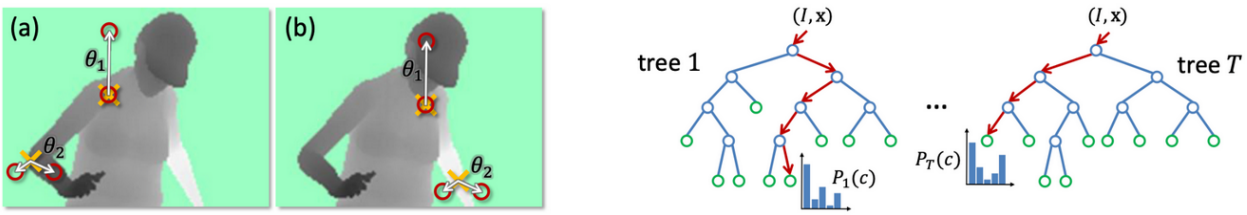


Figure 2.18: Left: depth comparisons between extracted features within the body part classification process. Right: randomized decision forests used for classification of image pixels to body parts. Figure taken from [244].

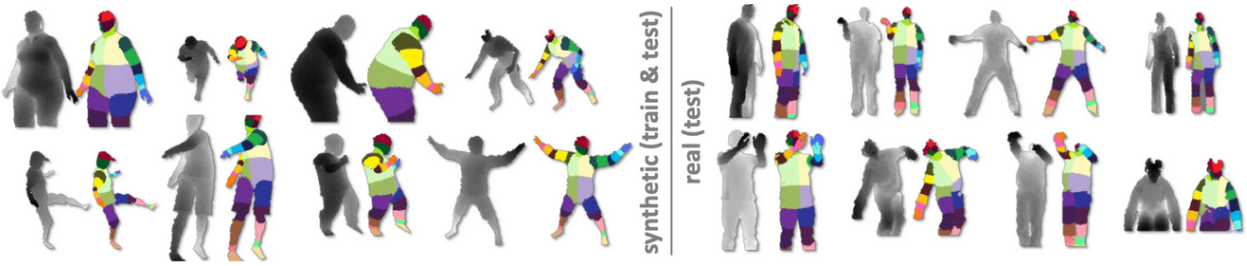


Figure 2.19: Synthetic and real poses used for training and evaluation of the Kinect skeleton extraction algorithm. Figure taken from [244].

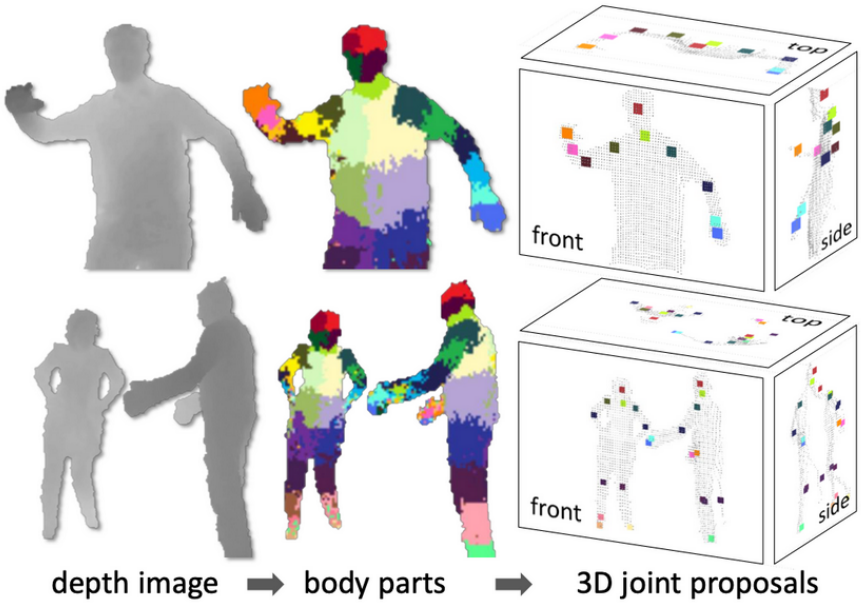


Figure 2.20: A sequence of images demonstrating the transformation from depth images to body parts and then to 3D joint proposals, ultimately leading to the identification of joints. Figure taken from [244].

sented Azure Kinect DK, which is a developer kit and PC peripheral which employs the use of artificial intelligence sensors for computer vision and speech models, and is connected to the Microsoft Azure cloud. While the previous iterations of Microsoft's Kinect

primarily focused on gaming, this device is targeted towards other markets such as logistics, robotics, health care and retail. Developers can create applications connected to Microsoft's cloud and AI technologies. It is mainly used in volumetric capture workflows through the use of software tools that can connect many Azure Kinects into one volumetric capture rig, allowing users to create interactive virtual reality experiences with human performances. The Azure Kinect DK is illustrated in Fig. 2.21.



Figure 2.21: The Azure Kinect DK sensor.

Apart from the Kinect sensors, there exist several other implementations that combine the extraction of RGB and depth data. In Fig. 2.22 we showcase a non-exhaustive list of other RGB-D sensor devices which have been available in the market, namely Asus Xtion[1], Creative Senz3d[2], Stereolab Zed[3], Intel Realsense[4], Orbec Astra Pro[5], Mynt Eye[6].



Figure 2.22: Several consumer RGB-D sensors. Top row: Asus Xtion, Creative Senz3d, Stereolab Zed; Bottom row: Intel Realsense, Orbec Astra Pro, Mynt Eye.

[1] http://xtionprolive.com/asus-3d-depth-camera
[2] https://finance.yahoo.com/news/creative-senz3d-tm-interactive-gesture-093000803.html
[3] https://www.stereolabs.com/zed-2/
[4] https://www.intelrealsense.com/
[5] https://www.orbbec.com/
[6] https://medium.com/robotics-weekends/mynt-eye-d1200-3d-camera-first-look-852da6389c42

2.4 OpenPose and PoseNet Libraries

OpenPose [30] and PoseNet [125] are two prominent computer vision frameworks that focus on human pose estimation, a crucial task in understanding and interpreting human movement. Human pose estimation involves identifying key points on a person's body, such as joints and body parts, and reconstructing their overall pose. This information is valuable in various applications, including gesture recognition, action recognition, and augmented reality.

OpenPose[7] is an open-source library for real-time multi-person keypoint detection and pose estimation. Developed by the Carnegie Mellon University Perceptual Computing Lab, OpenPose is widely used in computer vision and human-computer interaction applications. The key feature of OpenPose is its ability to detect and track the positions of multiple body keypoints, including those for the head, torso, arms, and legs, in real-time. One of the strengths of OpenPose is its versatility. It can work with various input sources, including images and video streams, and is capable of estimating not only the body pose but also facial landmarks and hand keypoints. OpenPose uses a bottom-up approach, initially detecting a large number of body part candidates and then assembling them into individuals. This enables it to handle complex scenarios involving occlusions, interactions between multiple people and other common imaging artifacts as it may be observed in Fig. 2.23.

Developers and researchers often integrate OpenPose into projects for applications such as motion analysis, gesture recognition, and virtual reality. The open-source nature of the project encourages collaboration and continuous improvement, making it a popular choice in both academia and industry.

PoseNet[8] on the other hand, is a pose estimation model developed by Google that operates in the browser using TensorFlow. It is designed for single-person pose estimation and is lightweight, making it ideal for web-based applications. PoseNet relies on a MobileNet architecture for efficient inference on a wide range of devices. The model estimates key points on a person's body, allowing developers to incorporate real-time pose estimation directly into web applications. PoseNet is widely used for interactive websites, fitness apps, and other projects where lightweight and accessible pose estimation is required.

It is trained to estimate the positions of key body parts, such as shoulders, elbows, wrists, hips, knees, and ankles. The output is a set of confidence scores and coordinates for each detected joint. The model offers two variants: PoseNet Single Person, designed for tracking a single person's pose, and PoseNet Multi Person, an extension that allows simultaneous tracking of multiple individuals. PoseNet is commonly used in applications like augmented reality (AR) filters, fitness apps, and interactive web experiences.

While both OpenPose and PoseNet share the goal of pose estimation, they differ in their scope and use cases. They represent two influential approaches to human pose estima-

[7] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[8] https://github.com/tensorflow/tfjs-models/tree/master/posenet

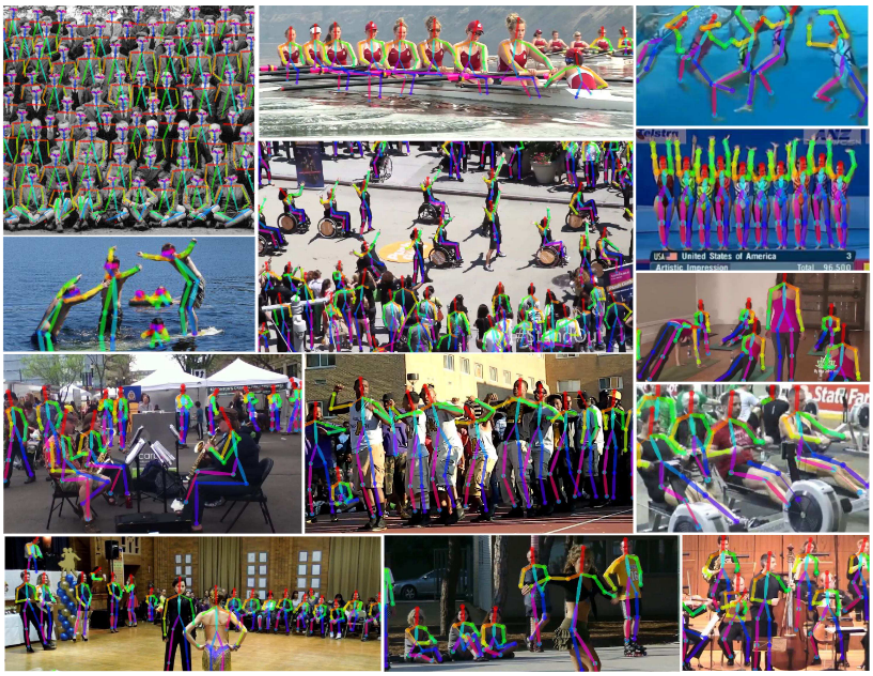


Figure 2.23: The output from OpenPose involves real-time detection of keypoints for the body, feet, hands, and face. Results contain viewpoint and appearance variation, occlusion, crowding, contact, and other common imaging artifacts. Figure taken from [27].

tion, catering to different needs in the computer vision and machine learning communities. The choice between these frameworks depends on the specific requirements of a project, considering factors such as accuracy, real-time performance, and the target platform. As the field of pose estimation continues to evolve, both OpenPose and PoseNet contribute significantly to advancing our understanding of human movement in diverse applications.

2.4.1 Datasets

In this subsection we present the three human activity datasets, which we employed to assess the methodologies that will be presented within the next Chapters. The selection of these datasets was predicated on several criteria, including the type of camera used for the recordings and the specific aspects of the dataset they provide, such as skeleton, RGB, depth data etc. The first dataset, known as the PKU-MMD dataset [163] consists the first of our choices. The second dataset is the SYSU 3D Human-Object Interaction (HOI) [102] dataset. The third and final dataset is the UTKinect-Action3D [311]. All three datasets provide RGB, depth, and 3D skeleton data sequences per activity, acquired using Kinect cameras. Note that PKU-MMD is a large-scale dataset, comprising tens of thousands of examples of a plethora of activities. Contrary to it, SYSU 3D Human-Object Interaction (HOI) is a small dataset that focuses on interactions between humans and objects, while UTKinect-Action3D dataset consists of simple activities. In the subsequent paragraphs, we aim to provide additional details about each of them, individually.

2.4.1.1 PKU-MMD Data Set

The main dataset that we have used for the experimental evaluation of our approach is the PKU-MMD dataset [163]. It is a large-scale benchmark dataset that focuses on human action understanding. It has a wide range of human activities and contains 2 Phases. In our experiments we focused on using the Phase #1. More specifically, Phase #1 contains approx. 20K action instances from 51 action classes which divided in 41 daily living actions (drinking, brushing teeth, drink water, etc.) and 10 interaction actions (hugging, kicking other person, handshaking, etc), spanning into 5.4M video frames. The whole set of the 41 classes of Phase #1 is depicted in Table 2.1. Videos last about 3$\sim$4 minutes, while the recording ratio is set to 30 fps. Each video typically contains approximately 20 action instances. For the data collection, 66 human subjects have been involved and took part in 4 daily and 2 interactive action videos. The subjects were divided in 9 groups, each group was able to freely choose to perform 6 random sequences among a set of the 54.

Moreover, each activity has been recorded by 3 camera views, namely *L* (left), *M* (middle) and *R* (right). As illustrated in Fig. 2.24 fixed angles have been used, i.e., -45°, 0° and $+45^{\circ}$. Note, that the height of all cameras is in this case the same and remains fixed and equal to 120cm, while the area, within which users perform actions is pre-determined. Specifically, the length and width of the activity area were equal to 180cm and 120cm, respectively. The Microsoft Kinect v2 camera was used for all recordings, and for each action instance the following data modalities are provided: a) raw RGB video sequences depicting one or more test subjects performing an action with resolution of 1920×1080; b) depth sequences, i.e., depth information of the aforementioned RGB sequences with resolution of 512×424; c) infrared radiation sequences of the aforementioned sequences were collected and stored frame by frame in 512×424; and d) extracted 3D positions of human skeleton joints. The available modalities are illustrated in Fig. 2.26 while a set of extracted frames of several activities is illustrated in Fig. 2.27. On the other hand, in Phase #2, 13 subjects took part in 49 action categories and 2000 videos were created with approximately 7 action instances. In that case, each video lasts about 1$\sim$2 minutes, while while the recording ratio is set again to 30 fps. We should herein note that Phase #2 videos were unavailable during the experimental evaluation of this work, thus, they are not be used at the following chapters. A brief visual summary of the PKU-MMD dataset is illustrated in Fig. 2.25.

For annotation purposes the authors used volunteers with prior experience in labeling. In order to evaluate and verify the accuracy of the data labeling they used the cross-view method thanks to multi-view capturing. Moreover, and in order to establish a standard assessment procedure for the outcomes achieved in this benchmark, the authors have established a set of criteria for evaluating precision and recall scores within detection tasks. They suggest two different dataset partition configurations, specifically:

- *Cross-View Evaluation*: in the context of cross-view evaluation, the authors have selected video sequences captured by the middle and right cameras, in order to form the training set, while the left camera's video sequences have been used as testing set. In this particular evaluation, the training set consists of 717 video samples, while

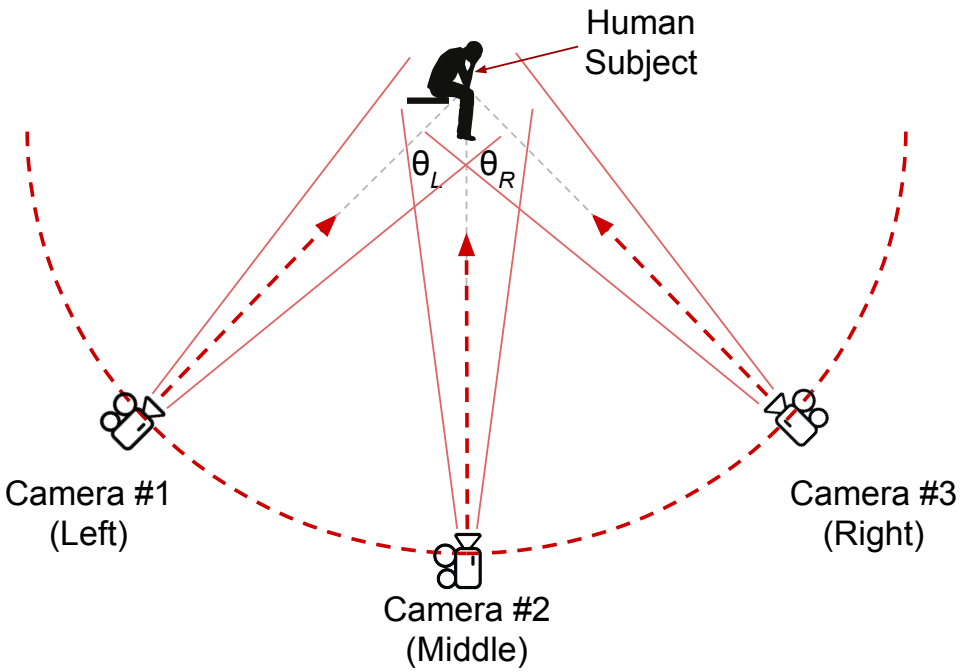


Figure 2.24: Camera setup of the PKU-MMD dataset. Three cameras are faced in front of the subject, capturing her/him under three viewpoints, under angles $\theta = \pm 45^{\circ}$ Cameras $\#1$, $\#2$, $\#3$ will be denoted as L, M, R, respectively.

the testing set comprises 359 video samples.

- *Cross-Subject Evaluation*: in the context of cross-subject evaluation, the authors have divided the subjects into two groups: a training group comprising 57 subjects and a testing group comprising the remaining 9 subjects. Within this evaluation framework, the training set comprises 944 long video samples, while the testing set comprises 132 such samples.

2.4.1.2 SYSU 3D Human-Object Interaction (HOI) Set

The SYSU 3D Human-Object Interaction (HOI) [102] dataset is a simpler RGB-D activity dataset with a specific focus on human-object interactions. It has been designed to facilitate comprehensive method evaluations in this domain. The dataset has been constructed with the involvement of 40 participants who were asked to perform 12 different activities in a freeform manner. Each of the 12 activities required participants manipulating one of six different objects, i.e.: phone, chair, bag, wallet, mop, and besom. These activities are presented in Table 2.2. Note that there are several activities which appear highly similar, e.g., mopping and sweeping. In total, the dataset comprises 480 video clips, offering a rich and varied collection of human-object interaction scenarios.

The videos in this dataset have been recorded, using a Kinect v1 camera and exhibit varying durations, ranging from 1.9 sec to 21 sec, providing a diverse temporal perspective.

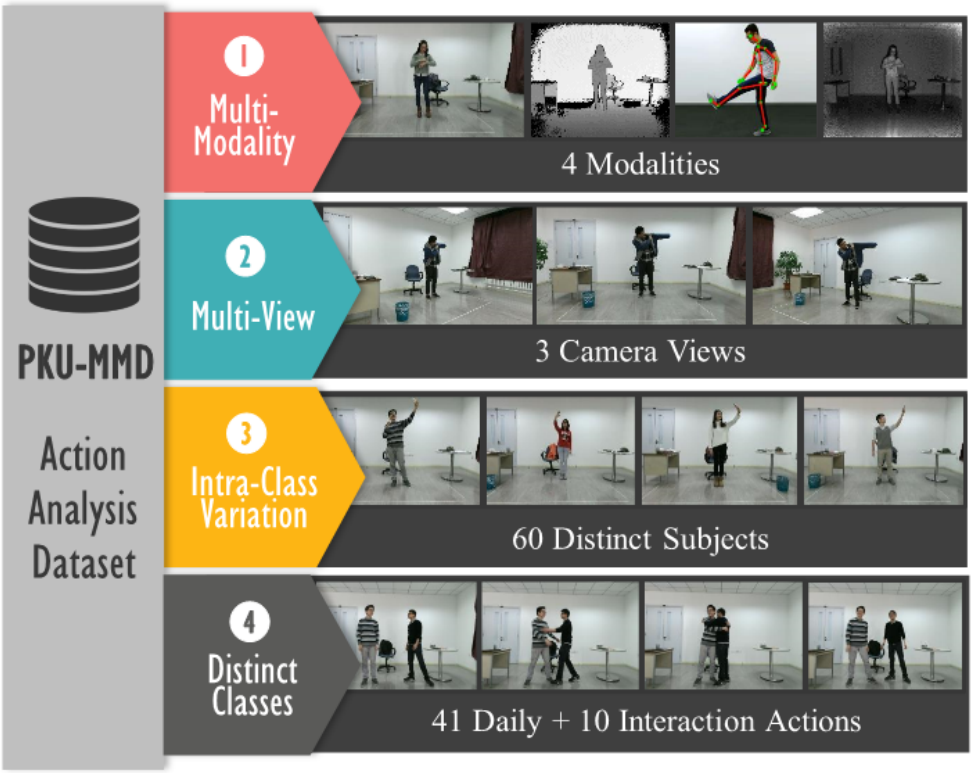


Figure 2.25: A brief visual summary of the properties of the PKU-MMD dataset, consisting of 4 modalities, 3 camera views, 60 distinct human subjects and 51 daily and interaction activities. Figure taken from [163].

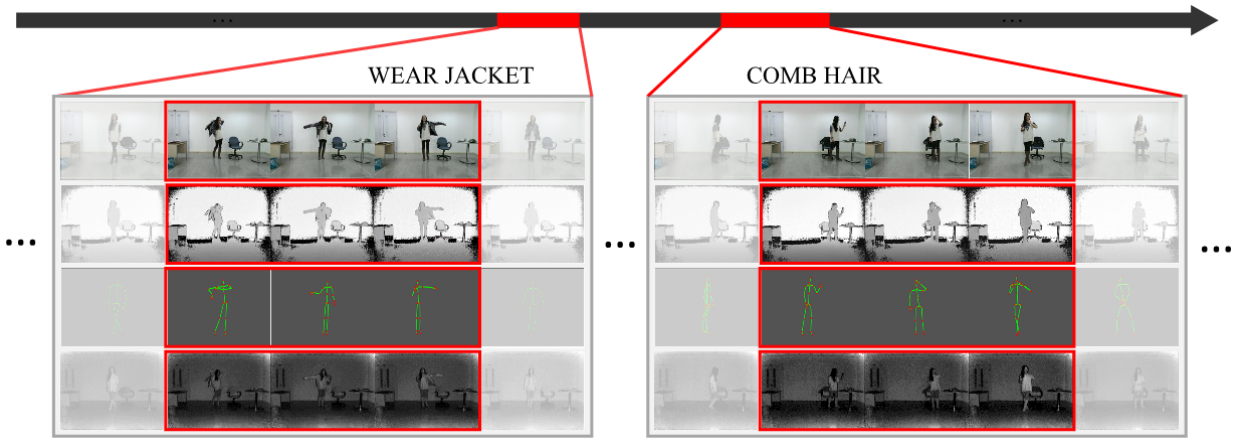


Figure 2.26: A set of 3 consecutive frames from the PKU-MMD, from each of the four modalities, i.e., per row and from top to bottom: RGB, depth, skeleton and IR. Figure taken from [163].



Figure 2.27: A set of extracted frames from the PKU-MMD, illustrating a subset of the collected 51 actions for single person activities and person-to-person interaction activities. Figure taken from [163].

Table 2.1: The 41 activity classes comprising the PKU MMD Phase #1 dataset.

PKU MMD Actions		
bow	kicking other person	take off a hat/cap
brushing hair	kicking something	take off glasses
brushing teeth	make a phone call/answer phone	take off jacket
check time (from watch)	pat on back of other person	take out something from pocket
cheer up	pickup	taking a selfie
clapping	playing with phone/tablet	tear up paper
cross hands in front (say stop)	point finger at the other person	throw
drink water	pointing to something with finger	touch back (backache)
drop	punching/slapping other person	touch chest (stomachache/heart pain)
eat meal/snack	pushing other person	touch head (headache)
falling	put on a hat/cap	touch neck (neckache)
giving something to other person	put something inside pocket	typing on a keyboard
hand waving	reading	use a fan (with hand or paper)/feeling warm
handshaking	rub two hands together	wear jacket
hopping (one foot jumping)	salute	wear on glasses
hugging other person	sitting down	wipe face
jump up	standing up	writing

Table 2.2: The 12 activity classes comprising the SYSU 3D-HOI dataset.

SYSU 3D-HOI actions		
drinking	pouring	calling phone
playing phone	wearing backpacks	packing backpacks
sitting chair	moving chair	taking out wallet
taking from wallet	mopping	sweeping

The dataset comprises various data modalities for each video clip. As in the previous cases, the authors offer RGB frames, which are high-quality color images, serving to capture the visual facets of the activities. In addition, depth sequences are also included, being aligned to the RGB frames, thereby supplying crucial depth information to augment

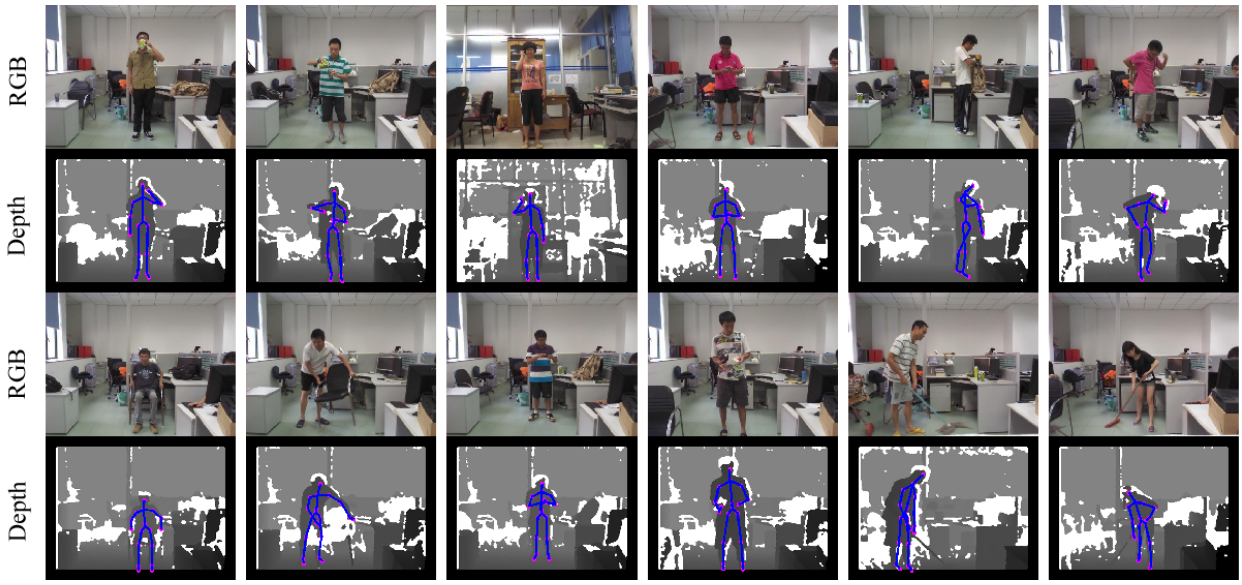


Figure 2.28: Examples of one sample per class from the SYSU 3D HOI dataset. Each row displays an RGB sample (phone, chair, bag, wallet, mop and besom) at the top and the corresponding depth channel with superimposed skeleton data is illustrated below. Figure taken from [102].

the visual data. Moreover, the dataset incorporates skeleton data, extracted by the Kinect API. For each of the 40 subjects, recorded, RGB video frames of resolution 640×480 px., depth sequences of resolution 640×480 px., and 3D skeleton data sequences comprising 20 joints are provided.

2.4.1.3 UTKinect-Action3D Dataset

The last dataset that we have used in this work was the UTKinect-Action3D Dataset [311]. Also in this case the footage was recorded using a Kinect v1 device within indoor environments. UTKinect includes 10 distinct action types which are summarized in Table 2.3. The dataset comprises 10 subjects (9 males, 1 female), and each subject performs each action twice. Three data channels were recorded: RGB, depth, and skeletal joint locations, and these channels are perfectly synchronized. The videos were captured at a frame rate of 30 fps. It's essential to note that only frames with a tracked skeleton were recorded, resulting in non-continuous frame numbers and an effective frame rate of approximately 15 fps.

The dataset encompasses four distinct components to facilitate a comprehensive understanding of the subject's actions in each video. Firstly, it includes RGB frames in JPEG format with a resolution of 640×480. Secondly, depth images are provided, with a resolution of 320×240. The third and final component consists of skeletal joint locations.

Table 2.3: The 10 activity classes comprising the UTKinect-Action3D dataset.

UTKinect-Action3D Dataset		
walking	sitting down	standing up
picking up	carrying	throwing
pushing	pulling	waving hand
clapping hands		



Figure 2.29: 10 activities of the UTKinect dataset with RGB and depth frames. First row: walk, stand up, sit down, pick up, carry. Second row: throw, push, pull, wave hands, clap hands. Figure taken from [311].

2.4.2 Typical HAR Evaluation Protocols

In Chapters 3 – 7 we will be mainly based on the following protocols to evaluate the proposed HAR approaches:

- **Single-view experiments**: in this case, both training and testing sets will derive from the same viewpoint. This is the most typical HAR evaluation approach. Considering the 3-camera setup of Fig. 2.24, a possible case is training and testing with samples captured using camera #1.

- **Cross-view experiments**: in this case, different viewpoints were used for training and for testing. The goal of these experiments is to test the robustness of a given HAR approach in case a typical transformation occurs (e.g., comprising a translation and a rotation); This could i.e. correspond to an abrupt viewpoint change, such as the ones that typically occur in real-life situations. Considering the 3-camera setup of Fig. 2.24, a possible case is training with samples captured using camera #1 and testing with samples captured using camera #2, or training with samples captured using cameras #1 and #2 and testing with samples captured using camera #3. Therefore, the goal of cross-view evaluation is to assess the robustness of an approach in relation to various transformations, such as translations and/or rotations.

- **Cross-subject experiments**: in this case subjects are split in training and testing groups, i.e., each may "participate" only into one of these groups. Note that all viewpoints may be used both for training and testing, while the goal of this evaluation is to test the robustness of a given HAR approach into intra-class variations. In real-life situations this is expected to happen when a system is trained e.g., within a laboratory environment and is deployed into a real ambient-assistive living environment.

2.4.3 Evaluation Metrics

In order to evaluate the performance of our classification models we needed some common metrics in the field of machine learning, which we herein present. Specifically the metrics that have been used are the following: Precision, Recall, Accuracy and F1-score. In this Section, we will analyze these metrics and also focus into the concept of the confusion matrix. Additionally, we will offer a simple and straightforward example, to illustrate how these metrics function and may be used within a classification task.

The aforementioned metrics are particularly useful when dealing with binary classification problems, where the goal is to classify data into one of two predetermined classes, typically referred to as the "positive" class and the "negative" class. They are often used together to provide a comprehensive evaluation of a classification model's performance. The choice of which metric to prioritize depends on the specific problem and its requirements.

To explain these notions, we should first begin by defining the following outcomes of a classification algorithm:

- **True Positive** (TP): these are the examples that the algorithm correctly classified as being "positive."

- **True Negative** (TN): these are the examples that the algorithm correctly classified as being "negative."

- **False Positive** (FP): these are examples where the algorithm incorrectly classified as being "positive" when they were actually "negative." These are also known as Type I errors.

- **False Negative** (FN): these are examples where the algorithm incorrectly classified as being “negative.” These are also known as Type II errors.

Bearing that in mind, let us consider a medical diagnosis scenario. Therein, we need to ensure that as many true positive cases as possible are detected, even if it results in more false positives. Specifically, it would be preferable that a patient is diagnosed for a disease she/he ultimately doesn’t have than it would be to receive a negative test result for a disease she/he actually has, since the latter could result in a serious illness not being treated, and could literally be a deadly mistake. In contrast, in the case of a spam email filter, minimizing the number of false positives might be more important, even if some spam emails are not detected, since it is preferable to have spam in the inbox than it would be to have an important email in spam.

We are now able to provide a brief overview of the metrics used:

1. **Precision** (P) measures the accuracy of positive predictions made by a classification algorithm. It is defined as the ratio of true positive predictions to the total number of positive predictions (both true positives and false positives):

$$P = \frac{TP}{TP + FP} \, . \tag{2.2}$$

   Note that high precision indicates that when the model predicts a positive result, it is likely to be correct.

2. **Recall** (R) measures the ability of a model to identify all relevant instances in the dataset. It is defined as the ratio of true positive predictions to the total number of actual positive instances:

$$R = \frac{TP}{TP + FN} \, . \tag{2.3}$$

   High recall suggests that the model can effectively identify most of the positive instances. Note that recall is often referred to as “Sensitivity of True Positive Rate.”

3. **Accuracy** (Acc.) is a straightforward metric that measures the overall correctness of a model’s predictions. It is the ratio of correctly predicted instances (both true positives and true negatives) to the total number of instances:

$$Acc. = \frac{TP + TN}{TP + FP + TN + FN} \, . \tag{2.4}$$

   Note that while accuracy is easy to understand, it can be misleading in imbalanced datasets where one class is significantly more prevalent than the other.

4. **F1-Score** (F1) is defined as the harmonic mean of precision and recall, providing a balanced measure of a model’s performance:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \, . \tag{2.5}$$

		Predicted
Actual	Predicted class Positive (cat)	Predicted class Negative (dog)
Actual class Positive (cat)	True Positive (TP) **430**	False Negative (FN) **30**
Actual class Negative (dog)	False Positive (FP) **15**	True Negative (TN) **601**

Figure 2.30: A toy example of a confusion matrix for the binary classification task cats vs. dogs.

It takes both false positives and false negatives into account, making it particularly useful when there is the need to balance precision and recall. Moreover, it consists a good metric to use when there is an uneven distribution between the positive and negative classes in the dataset.

Revisiting the medical diagnosis and the spam email filter scenarios, within the former the highest possible recall is desired, whilst in the latter, the highest possible precision is desired.

Given the aforementioned definitions of TP, TN, FP and FN, we are now able to define the notion of the confusion matrix. Specifically, a confusion matrix in case of a binary classification task is typically organized in a 2×2 table, as illustrated in Fig 2.30 and is used to assess the performance of classification models. However, it may not be used only in binary but also in multi-class classification tasks. It works by providing a detailed breakdown of the model's predictions, comparing them to the actual true values. At the following and for the sake of the explanation, we will explain the application of a confusion matrix into a binary classification task.

As illustrated in Fig. 2.30, a confusion matrix consists of four main components, corresponding to TP, TN, FP and FN. At the following, we provide a toy arithmetic example for better understanding of how a confusion matrix is formed.

To begin, let us assume two categories: the first one being "dog," and the second being "cat." We designated the "cat" class, as being the class of interest, thus denoting the "positive" class, while the "dog" class denoting the "negative" class. As illustrated in Fig:

2.30, the “cat” class consists of a total of $430 + 30 = 460$ samples. On the other hand the “dog” class consists of a total of $15 + 601 = 616$ samples.

Specifically, by observing the confusion matrix we are able to calculate TP, FN, FP and TN as follows:

- TP: we refer to the 430 samples that belong to the “cat” class and have been correctly classified as “cat,”
- FN: we refer to the 30 samples that belong to the “cat” class and have been misclassified as “dog,”
- FP: we refer to the 15 samples that belong to the “dog” class and have been misclassified as “cat,”
- TN: we refer to the 601 samples that belong to the “dog” class and have been correctly classified as “dog.”

Therefore, we are now able to calculate Acc., P, R and F1 as follows:

- by using Eq. 2.2, $P = \frac{TP}{TP+FP} = \frac{430}{430+15} = 0.96$,
- by using Eq. 2.3, $R = \frac{TP}{TP+FN} = \frac{430}{430+30} = 0.93$,
- by using Eq. 2.4, $Acc. = \frac{TP+TN}{TP+TN+FP+FN} = \frac{430+601}{430+601+15+30} = 0.96$,
- by using Eq. 2.5, $F1 = 2 \cdot \frac{P \cdot R}{P+R} = 2 \cdot \frac{0.96 \cdot 0.93}{0.96+0.93} = 0.94$.

As we can observe from the above results, in the specific case the classification model appears to be performing well, since it combines high precision and recall values.

2.4.4 Implementation and Libraries

All experiments were performed on a personal workstation running Ubuntu 18.04/20.4/22.4 (64 bit) with the following specifications: an Intel™i7 5820K 12 core processor on 3.30 GHz and 16GB RAM, using NVIDIA™Geforce GTX 2060 GPU with 8 GB RAM. The deep architectures have been implemented in Python, using Keras 2.2.4 [38] with the Tensorflow 1.12 [1] backend. All data pre-processing and processing steps have been implemented in Python 3.6 using NumPy [195] , SciPy [115] and OpenCV [22]. Specifically:

- Python[9] is a versatile, open-source programming language with a rich history. It was initially released in 1991 by Guido van Rossum at Centrum Wiskunde & Informatica in the Netherlands. Over the years, Python has gained immense popularity due to its readability, particularly appealing to beginner developers. It has found special use in the fields of data science and machine learning. Python’s strength lies in its extensive support for various libraries, such as the following:

[9] `https://www.python.org/`

- Keras[10] is a high-level neural network API running on top of TensorFlow. It serves as an interface for TensorFlow, providing users with a straightforward and user-friendly way to design, construct, and train neural networks. Its appeal lies in its flexibility, simplicity, and extensive documentation, which has contributed to its popularity within the machine learning community. With Keras, developers can quickly prototype and experiment with deep learning models without delving too deeply into the lower-level details of the framework. This accessibility has made it a preferred choice for many machine learning practitioners.
- TensorFlow[11] is a Google-developed end-to-end platform for developing and deploying machine learning and deep learning models across a wide range of applications. It is well-known for its flexibility, scalability, and robust ecosystem of tools and resources.
- NumPy[12] which stands for “Numerical Python,” is a fundamental library used for numerical computing in Python. It is particularly known for its support of multi-dimensional arrays and matrices, making it an essential tool for tasks involving numerical and scientific computations.
- SciPy[13] meaning “Scientific computing in Python” is an open-source library for scientific computation, using NumPy, with a plethora of mathematical algorithms. It includes various modules for statistics, linear algebra, integration, and optimization.
- Scikit-Learn[14], often abbreviated as “sklearn” is an open-source library machine learning for Python. It offers an easy-to-use and effective toolkit for a range of machine learning applications, including classification, regression, clustering, and more.
- OpenCV[15] is an acronym for “Open Source Computer Vision Library,” which, as the name suggests, is an open source library focused on computer vision and image processing which provides numerous types of tools and functions

Conlcuding, Python’s extensive library set and community support, make it a good choice for a wide range of applications, including scientific research, web development, and artificial intelligence.

[10] https://keras.io/
[11] https://www.tensorflow.org/
[12] http://www.numpy.org/
[13] https://www.scipy.org/
[14] https://scikit-learn.org/stable/
[15] https://opencv.org/

3. HUMAN ACTIVITY RECOGNITION USING CNNS

3.1 Introduction

As we have already stated in Chapter 1, understanding human activity from video has been gaining increasing attention in recent years, forming a significant aspect of computer vision tasks. We remind that Wang et al. proposed a categorization of activity into the following sub-problems: gesture, action, interaction, and group activity recognition [302]. Specifically, gestures demand relatively less time, while actions are more time-consuming and typically involve multiple body parts. interactions encompass either a person and an object or two individuals, and group activities are combinations of the aforementioned. Moreover, challenges in human action understanding include representation, analysis, and recognition of actions [16]. Earlier recognition approaches, like Schuldt et al.'s (2004), relied on hand-crafted features and traditional machine learning algorithms, such as SVMs [234]. However, their accuracy diminishes significantly with an increased number of actions and may lack robustness to viewpoint changes. Recent hardware advances, particularly in graphics processing units (GPUs), facilitate fast training of deep neural network architectures [134]. These approaches eliminate the need for a feature extraction step, learning features within the network, and their accuracy scales with the available training examples.

Moreover, the advent of cost-effective sensors like Microsoft Kinect has provided depth data, enriching human activity datasets with multi-modal raw data, including RGB video and depth information. Depth data allows for the extraction of skeleton sequences, representing 3D coordinates of human joints over time. While depth modality is invariant to illumination changes and reliable for body silhouette estimation, RGB information, with its color and texture, remains significant for discriminating actions, particularly those involving human-object interactions. Combining different modalities offers diverse perspectives on actions, suggesting meaningful correlations. Skeletal information proves valuable for accurately capturing human body posture. In scenarios limited to sequence data for motion features, CNN-based methods face the challenge of efficiently encoding skeleton sequences while capturing spatio-temporal activity features.

Based on the aforementioned developments, the novelty presented in this Chapter is a novel 2D visual representation of 3D temporal human skeletal motion, which is based on well-known 2D image transformations and is applied to the problem of HAR. Specifically, these transforms are the Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and Discrete Sine Transform (DST). In brief, raw signal images capturing 3D motion of human skeletal joints over space and time are transformed using these methods, resulting in "activity" images that capture the spectral properties of signal images. For classification, a deep CNN architecture is applied. Evaluation takes place using the PKU-MMD dataset [163], while the approach's efficacy is demonstrated in both recognizing activities of daily living (ADLs) in real-like environments [141] and in broader scenarios.

The rest of this Chapter is organized as follows: in the remaining of this Section, and

specifically in Subsection 3.1.1 we present an initial arm gesture experiment that served as the basis and the motivation of all methodologies that have been implemented in the context of this Thesis. In Section 3.2 we present related work on HAR, focusing on research efforts that extract 3D skeletal motion and use 2D image representations prior to classification. Then, in Section 3.3 we present the HAR methodology that is based on the creation of 2D signal and activity images from raw 3D skeletal data. The proposed approach is evaluated and corresponding results are discussed in Section 3.4 and finally, conclusions are drawn in 3.5.

3.1.1 Initial experiment

In order to provide a proof of concept for the proposed framework for HAR, we begin with a simple, initial experiment that focuses on arm gestures. In other words we attempt to solve a simpler sub-problem of HAR so as to assess the feasibility and the efficiency of the proposed description and recognition scheme. Specifically by "arm gesture" we define any gesture that involves palms, wrists, shoulders and/or elbows. We use a deep learning Convolutional Neural Network architecture. Our approach is based on the Kinect v1 sensor and its Software Development Kit (SDK) in order to detect and track the subject's skeletal joints in the 3D space. We then select a subset of these joints, i.e., all that are involved at any of the gestures of our data set. Then, we create an artificial image based on these 3D coordinates. We apply the Discrete Fourier Transform on these images and use the resulting ones to train the CNN. We compare this approach with the one of Paraskevopoulos et al. [205], where a set of hand-crafted statistical features on joint trajectories had been used. We evaluate this approach using a dataset of 10 arm gestures that we have created specifically for this problem.

The problem of arm gesture recognition has attracted many research efforts during the last decade. In this section our goal is to present works that aim to recognize simple arm gestures, i.e., as the ones we also aim to recognize in the context of this work. We shall present both approaches that use traditional machine learning techniques and also approaches that are based on deep learning. Feature-based works typically rely on traditional machine learning approaches such as artificial neural networks (ANNs), support vector machines (SVMs), decision trees (DTs) or K-nearest neighbor classifiers (KNN). In [18] SVMs and DTs are trained on 3D skeletal joint coordinates, while in [138] distances in reference to the *Spine Center* are used as features with a KNN classifier. Within the approach of [177], cascades of ANNs are used to firstly classify the gesture side (left/right) and then recognize the gesture type. In the work of [185] SVMs are used to recognize distinctive key poses, while Decision Forests are then applied to recognize gestures as sequences of such poses. Other approaches [32,106,225,227] exploit the fact that a gesture may be considered as a temporal sequence that typically varies in terms of speed and use the Dynamic Time Warping Algorithm (DTW) [5], while in some cases, a machine learning algorithm complements the classification scheme. Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs) have also been frequently used [9, 75, 82, 274, 320] since they are able to solve temporal pattern recognition problems. Their input typically

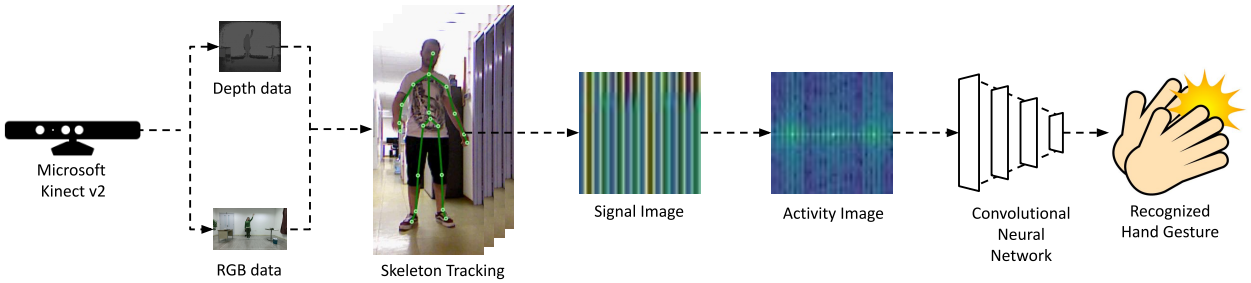


Figure 3.1: A graphic illustration of the proposed methodology for arm gesture recognition. Specifically, 8 skeleton joints are tracked during a gesture. Then, a signal image is constructed. Upon application of the Discrete Cosine Transform, the activity image is constructed and given as input to the CNN which classifies it to the depicted class, i.e., *clapping*.

consists of 3D positions of skeletal joints or body parts.

Paraskevopoulos et al. [205], proposed an approach for feature extraction using measurements on skeletal joints and evaluated it using several machine learning algorithms. Oyedotun et al. [198], proposed an approach for the recognition of arm gestures from the American Sign Language using CNNs and auto encoders. 3D CNNs have been used by Molchanov et al. [187] to detect drivers' arm gestures and by Camgoz et al. [26] for continuous gesture recognition. Moreover, Sanchez et al. [230] used a CNN for arm gesture recognition, which would then serve as an estimation of hand pose and orientation. Several CNN architectures to predict 3D joint locations of a hand given a depth map have been investigated by Oberweger et al. [193]. A synthetically trained CNN has been presented by Tsai et al. [276]. A method based on multi-scale and multimodal deep learning has been presented by Neverova et al. [191]. A two-stage approach that combines feature learning from RGB and depth with CNNs and PCA to extract the final features has been presented by Li et al. [155]. Deep dynamic NNs were used by Wu et al. [310] for both gesture segmentation and recognition. A Deep Belief Network and a CNN were used by Tang et al. [267] for learning features that are insensitive to movement, scaling, and rotation from hand posture images.

3.1.1.1 Skeletal Data Extraction

We may define a *gesture* as the transfer of a subset of joints from point A to point B, using a vaguely predefined trajectory. Note that a gesture may also be defined as a continuous stream of poses, between A and B, i.e., an ordered of temporary motion suspensions where body joints assume a distinct configuration in space [205]. Also, we limit our recognition in arm gestures, i.e., gestures that are based on motion of hands, yet involving palms, wrists, elbows and shoulders. We herein present the proposed approach for classification of arm gestures. A graphic representation of the proposed methodology is illustrated in Fig. 3.1.

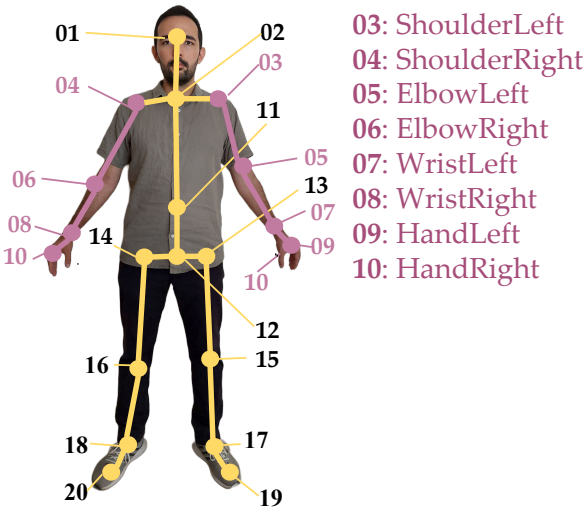
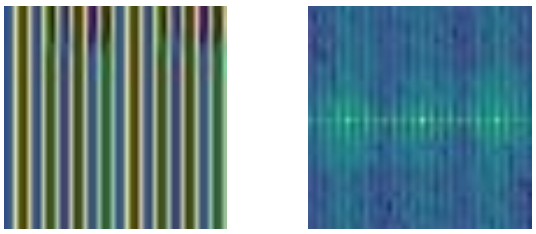


Figure 3.2: The 8 selected skeleton joints that have been captured using Kinect v1 and have been used in the initial experiment.

Specifically, using a set of 20 3D skeletal joints, extracted using the Kinect v1, we select a subset of them that is typically involved in hand gestures. More specifically, we use the following 8 joints: *Hand Left*, *Wrist Left*, *Elbow Left*, *Shoulder Left*, *Hand Right*, *Wrist Right*, *Elbow Right* and *Shoulder Right*, that correspond to hands and arms. This subset of joints is illustrated in Fig. 3.2.

Inspired by the work of Jiang and Yin [114] who worked using raw sensor measurements from inertial sensors, as a first step "signal" images are created, upon concatenating the signals that are produced by skeletal motion. Specifically, the motion of each skeletal joint in the 3D space, over time is treated as 3 independent 1D signals. Each of them corresponds to a coordinate. Therefore, for a given joint j, let $S_{j,x}(n), S_{j,y}(n), S_{j,z}(n)$ denote the 3 1D signals that correspond to its 3D motion and for the coordinates x, y, z, respectively. Thus, in the signal image, $S_{j,x}(n)$ corresponds to row $3 \times j - 2$. Accordingly, $S_{j,y}(n)$ and $S_{j,z}(n)$ correspond to rows $3 \times j - 1$ and row $3 \times j$. This way, the signal image $\mathbf{S}$ for a given activity and for N joints, is created upon concatenation of the $3 \times N$ signals, thus its dimension is $3 \times N \times T_s$, where T_s is the duration of this activity.

However, in real problems, since different subjects may perform the same activity with different duration and also different activities require different duration, it should be obvious that T_s is variable. In order to address the problem of temporal variability between subjects and between activities, so as to allow for signal concatenation, as discussed, a linear interpolation step is imposed. This way the length of all activities T_a is fixed. Note that in order to set value of T_a the process begins with the selection of a value close to the mean of all activities, yet the exact value that is finally used is chosen upon experimentation and fine-tuning.



a. b.

Figure 3.3: (a) a sample signal image; and (b) the corresponding activity image from gesture “clapping.” Figure best viewed in color.

3.1.1.2 Activity Images

Let $\mathbf{S}$ denote a signal image with dimensions $W \times H$ and $\mathbf{S}_{nm}$ the pixel at coordinates (m, n). From each signal image an “activity” image $\mathbf{A}$ is created. This may be done by applying the following image transformation [76, 110] on a given signal image $\mathbf{S}$: the 2-D Discrete Cosine Transform (DCT), which is defined as:

$$\mathbf{A}(u,v) = a_u a_v \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \mathbf{S}(x,y) \left(\cos \frac{\pi(2m+1)u}{2W} \cos \frac{\pi(2n+1)v}{2H} \right) , \tag{3.1}$$

where $x \in [0, W-1], y \in [0, H-1], u \in [0, W-1], v \in [0, H-1]$ and also:

$$a_u = \begin{cases} \frac{1}{\sqrt{W}}, u = 0 \\ \sqrt{\frac{2}{W}}, 1 \leq u \leq W-1 \end{cases} \tag{3.2}$$

and

$$a_v = \begin{cases} \frac{1}{\sqrt{H}}, v = 0 \\ \sqrt{\frac{2}{H}}, 1 \leq v \leq H-1 \end{cases} ; \tag{3.3}$$

Note that from the DCT transform only the magnitude is preserved, while the phase is discarded and further processed by normalizing using the orthonorm. Obviously, the result is a 2-D image, with the same dimension as the signal image $\mathbf{S}$.

In Fig. 3.3 a signal image example of the class “clapping” and the corresponding activity image are illustrated. Furthermore, in Fig. 3.4 the 10 corresponding activity images, created upon applying the aforementioned transform on the signal images $\mathbf{S}$ are illustrated.

At this point we focus only on the classification of a gesture into a set of predefined classes. Therefore, we should clarify that we do not perform any temporal segmentation to detect the beginning and the ending of a possible gesture; instead we consider this problem as solved. We use pre-segmented videos and only aim to recognize gestures per segment; each may contain at most one gesture. Since gestures typically vary in terms of duration even when performed by the same user, we perform a linear interpolation step to ensure

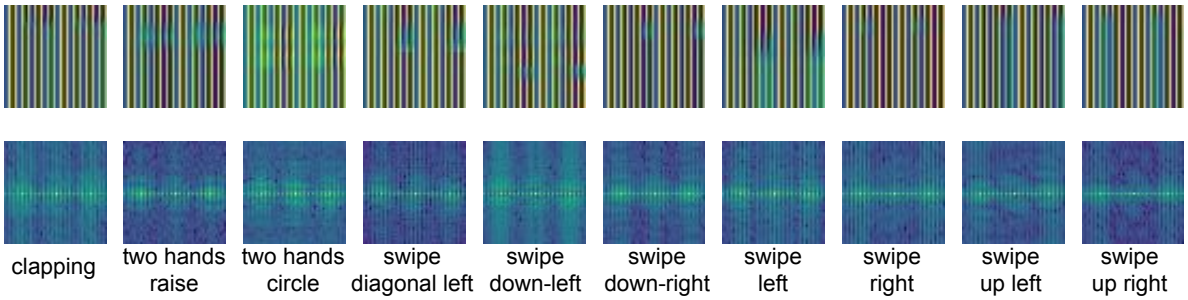


Figure 3.4: A sample signal image and the corresponding activity image from all 10 gestures recognized throughout our experiments. Upper row: signal images; lower row: activity images. Figure best viewed in color.

the same length on all gesture sequences in order to concatenate them to form the signal image. In the context of this experiment, the length is set to $T_a = 60$. Moreover, as we have already mentioned, we use 8 3-D skeletal joints, therefore, all signal images have size equal to 24×60.

3.1.1.3 Gesture Classification

For arm gesture classification we have used a CNN that is presented in detail in Fig. 3.5. Specifically, the first convolutional layer filters the 24×60 input activity image with 16 kernels of size 3×3. A second convolutional layer filters the 24×60 resulting image with 16 kernels of size 3×3. Then the first pooling layer uses "max-pooling" to perform 3×3 sub-sampling. The third convolutional layer filters the 11×29 resulting image with 32 kernels of size 3×3. Then, the fourth convolutional layer filters the 11×29 resulting image with 128 kernels of size 3×3. A second pooling layer uses "max-pooling" to perform 2×2 sub-sampling. Then, a flatten layer transforms the output image of size 5×14 of the second pooling to a vector, which is then used as input to a dense layer using dropout. Finally, a second dense layer using dropout produces the output of the network.

3.1.1.4 Experiments

In order to evaluate our initial experiment, we have constructed a real-life dataset. More specifically, we have involved 10 users (5 male and 5 female, aging between 22 and 30 years old) in the process. In order to obtain spontaneous gestures, none of these users was neither involved in the research conducted in the context of HAR, nor had been previously involved in similar tasks. Moreover, we only provided them with an intuitive description of the way gestures should be performed, allowing "noise" in the collection process. This way, we feel that the collected gestures were as "natural" as possible. The set of gestures consists of the following: *Swipe-left*, *Swipe-right*, *Swipe-up-left*, *Swipe-up-right*, *Swipe-down-left*, *Swipe-down-right*, *Swipe-diagonal-left*, *Clapping*, *Two-hands-raise*

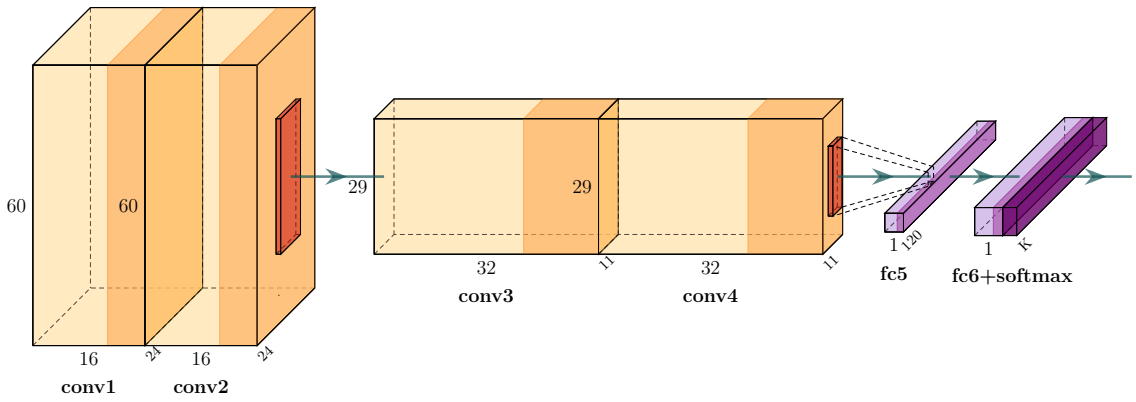


Figure 3.5: The CNN architecture that has been used in our initial experiment for arm gesture classification.

and *Two-hands-circle*. In Fig. 3.6 we illustrate a sequence of RGB frames with overlaid the skeleton for several gestures; the remaining may be easily conceived, since they are symmetric to some the aforementioned.

Note that we have asked all users to perform each gesture at least 10 times and then we manually cleaned the data set by removing the wrongly performed gestures and keeping only 5 samples per gesture, per user. This resulted to a total of 500 gestures. All samples were manually segmented using a video editor; sample gestures begin approx. 100msec upon the beginning of the corresponding video, while end approx. 100msec prior to its ending. This way we were able to isolate gestures from other movements. For each gesture, we also recorded the relevant skeletal data and extracted the 3D coordinates and the corresponding sets of features of Paraskevopoulos et al. [205], for all the aforementioned joints involved in these gestures, i.e., Shoulders, Elbows, Wrists and Hands. We performed a cross-subject evaluation: 8 users were used for training, while the 2 remaining ones were used for testing.

3.1.1.5 Results of the Initial Experiments

We compared our approach with the one of Paraskevopoulos et al. [205] that was based on a set of hand-crafted features and Extremely Randomized Trees (ERT) or Support Vector Machines (SVM) with non-linear kernels, which showed the best performance among a set of well-known classifiers. However, the proposed approach showed superior performance. As it may be seen in the results that are summarized in Table 3.1. Although the performance of the ERT was satisfactory, as expected, still it was inferior to the one of the proposed CNN architecture. However, we should note that the produced models from the ERT and the SVM were far less complex than the one of the CNN, which means that less time would be needed for prediction, something crucial for certain applications.

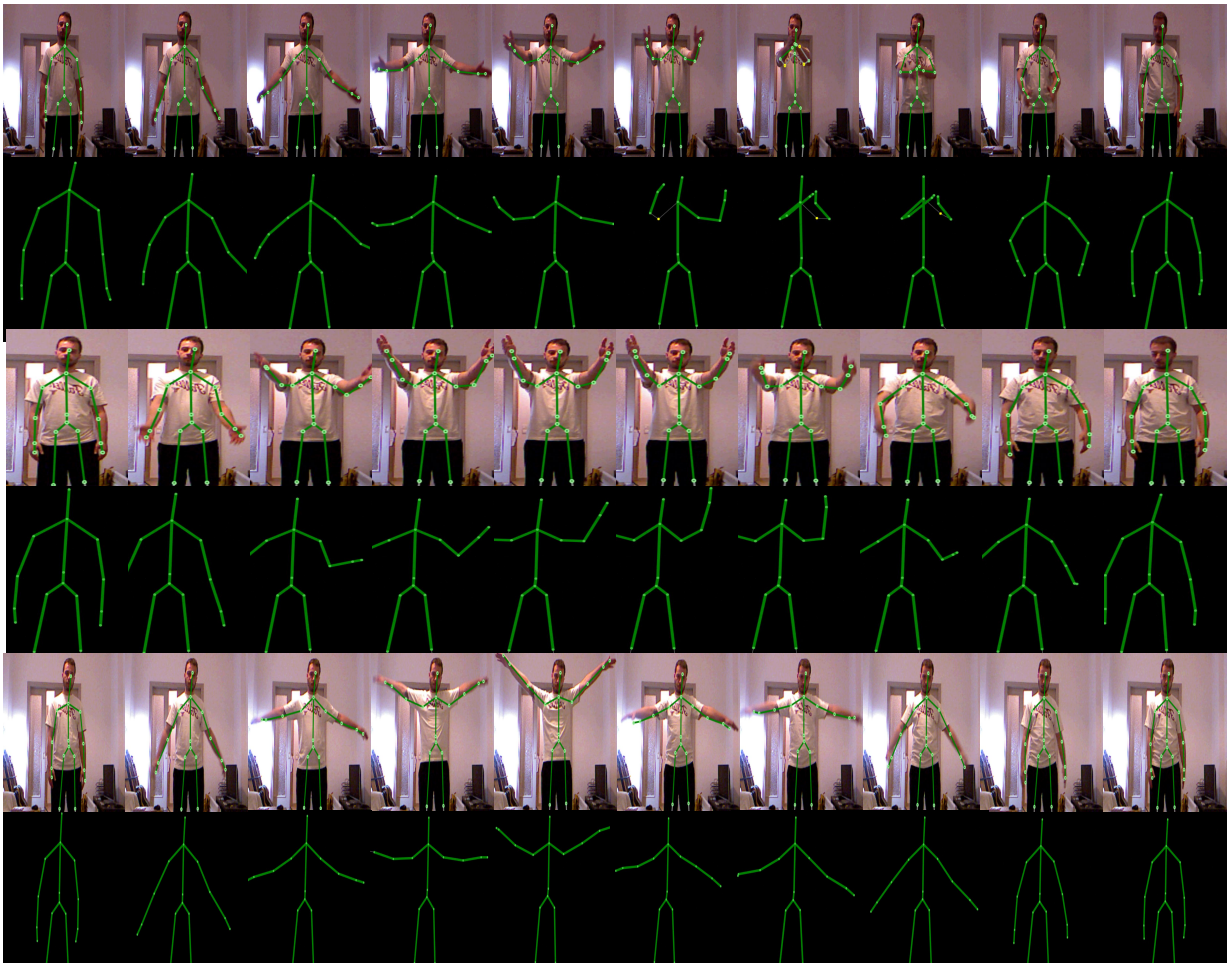


Figure 3.6: Examples of gestures taken from the data that has been created for the arm gesture experiments. For each example skeleton sequences overlaid to RGB data and single are illustrated. From top to bottom: *clapping*, *two hands raise* and *two hands circle*.

3.1.1.6 Preliminary Conclusions

In the beginning of this Chapter we presented initial results of a simple approach for the recognition of arm gestures, using a convolutional neural network. We based our approach on the creation of "activity" images, each resulted upon application of the Discrete Cosine Transform to a given signal image. This image was constructed upon concatenation of raw signals that correspond to the location of skeletal joints in the 3D space. We compared the proposed approach to previous work that used a set of hand-crafted features and two machine learning algorithms, namely extremely randomized trees (ERTs) and support vector machines (SVMs) with non-linear kernels. To this, we used a custom dataset that consisted of 10 gestures involving either one or both hands. Although the performance of the ERT was satisfactory, as expected, still it was inferior to the one of the proposed CNN architecture. However, we believe that these early results on arm gesture recognition are quite satisfactory and that the notions of signal and activity images may be applied to the

Table 3.1: Results of the initial arm gesture classification experiment and comparison to the approaches presented in Paraskevopoulos et al. [205], using extremely randomized trees (ERT) and support vector machines (SVM). Numbers denote accuracy. Best result indicated in bold.

	proposed	**ERT from [205]**	**SVM from [205]**
accuracy	**0.90**	0.82	0.72

more complex problem of human activity recognition, which will be investigated within the next sections of this Chapter.

3.2 Related Work on HAR based on 2D Representations of 3D Skeletal Motion

The problem of human action recognition has attracted many research efforts, which have been continuously growing during the last decade. In this section we aim to present approaches that are based on the representation of the 3D skeletons extracted by human activities with 2D image representations. The latter may be used directly as input into a CNN, typically without any feature extraction step; features are learned by the convolutional layers. In this section our goal is to present state-of-the-art research works on HAR focusing primarily on the way the aforementioned 2D representations are formed.

3.2.1 Visual Skeletal Representations

Huynh-The et al. [105] proposed a technique named, "pose-transition feature to image," transforming skeletal information from video sequences to skeleton-based images. Two geometric features are extracted, i.e., joint-joint distance and joint-joint orientation. For each video frame, a row vector F is composed with four normalized values. The first two values correspond to the distance and orientation between two arbitrary joints, while the last two values correspond to the distance and orientation between two arbitrary joints of two consecutive frames, respectively. The RGB skeleton-based image is formed by stacking the feature vectors F of all skeleton frames in the video, and encoding the normalized values as color pixels. Similarly, Pham et al. [215] represented the skeletal information with an enhanced action map, named "enhanced Skeleton Posture-Motion Feature" (SPMF). Joint motion and posture are encoded using the aforementioned joint features, the Euclidean distance between two joints and the joint-joint orientation. Additionally, the Adaptive Histogram Equalization (AHE) which is a color enhancement method, is adopted for increasing contrast and highlighting the texture and edges of the motion maps.

In the work of Du et al. [50], in order to preserve the spatial information the set of joints is split into five subsets corresponding to arms, legs and the trunk. Pseudo-colored images are generated by corresponding x, y and z spatial coordinates to R, G and B components, respectively. To preserve temporal information, spatial representations are chronologically arranged. Wang et al. [303] proposed a representation of "joint trajectory maps,"

wherein the motion direction. Maps were constructed by appropriately setting saturation and brightness so as texture would correspond to motion magnitude; each was based on the projected trajectory of the skeleton to a Cartesian plane.

Similarly, Hou et al. [96] transformed the extracted skeleton joints into a representation called "skeleton optical spectra," so that hue changes would reflect to the temporal variation of skeletal motion. Li et al. [153] proposed the representation of "joint distance maps," and opted for encoding the pair-wise joint distances in the 3 orthogonal 2D planes and also used a fourth one to encode distances in the 3D space, while hue was used to encode distance variations. Each map was separately classified and a late fusion scheme was adopted. Therein, pairwise distances of joints generate four JDMs. The first three maps correspond to distances in the three orthogonal planes (xy, yz and xz), while the fourth encodes distances calculated in the 3D space (xyz). The existence of the fourth JDM improved the robustness of the method when tested on multiple viewpoints. Again, the image's hue expresses the variations of joint distances.

Furthermore, Li et al. [152] presented a deep learning model that preserves spatial and temporal features, taking advantage of both LSTM and CNN architectures. Inspired by [215] and [303], spatial domain features and temporal domain features are extracted from skeleton sequence data. The former consist of relative position distances between joints and distances between joints and the lines connecting two joints. Temporal domain features are generated using the aforementioned methods to construct JDMs and JTMs. Spatial domain features are used as input to LSTM networks, while temporal domain features are used to train a CNN. To fuse resulting feature vectors a multiply score fusion is adopted.

Liu et al. [168] proposed a spatio-temporal representation for skeleton sequences that also incorporates the various durations of the different actions performed. The constructed images are based on a three-channel image patch, termed "Skepxel," which is composed by arranging the indices of the skeleton joints in a 2D grid and encoding their coordinate values along the third dimension. Each Skepxel might have a different arrangement of the joints but in order to keep the representation of the skeleton sequence compact, only a few, highly relevant arrangements are selected. A group of Skepxels are generated for a single skeleton frame and the final image is compactly constructed by concatenating the group of skepxels in a column-wise manner for an N-frame sequence. In an effort for invariance to the initial position and orientation of the skeleton, Liu et al. [169] applied transforms to skeletal sequences. The representation of each joint consisted of its 3D space coordinates, also adding time and joint label to create a 5D space. Upon projection to a 2D image using two of the aforementioned dimensions, the remaining three were used as R, G, B values to form pseudo-colored images. An approach addressing the problem of view invariance was presented in the work of Liu et al. [169] where a transformation creating a 5D representation of joints has been proposed. They adopted a 5D space, consisting of the 3D coordinates of each joint and the additional two dimensions of time and joint label. Upon projection to a 2D image using the dimensions of time and joint label, the remaining three dimensions are used as R, G, B, channels to form pseudo-colored images.

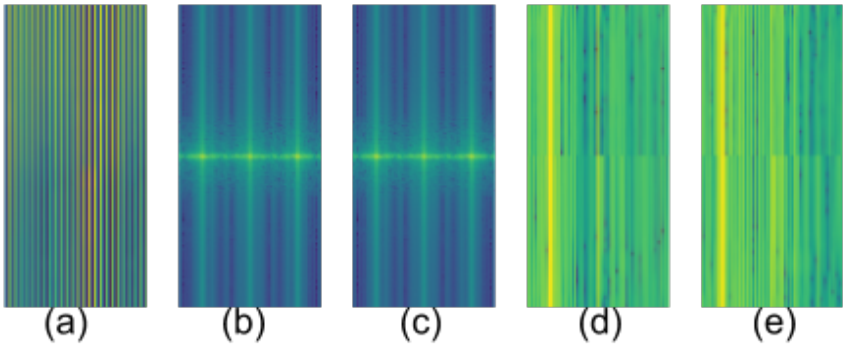


Figure 3.7: (a) A signal image; activity image resulting upon (b) DFT; (c) FFT; (d) DCT; (e); DST. Action is *playing with phone/tablet*. DFT and FFT images have been processed with log transformation for visualization purposes. Figure best viewed in color.

Similarly, Yang et al. [317] proposed a "tree structure skeleton image" (TSSI), based on the idea that spatially related joints in original skeletons have direct graph links between them. For their method, human skeleton graph structure is rearranged using a depth-first tree traversal order and therefore, the spatial correlations between joints are better preserved. Finally, Ke et al. [121] proposed a skeletal representation capable of extracting translation, rotation and scale invariant features. In their method, five subsets of joints are selected to represent the following body parts, arms, legs and trunk. For each body part the cosine distances and the normalized magnitudes are calculated, creating two feature arrays, which are then transformed into gray scale images. Instead, they extracted translation, rotation and scale invariant features by subsets of joints as in [50]. From each, they extracted cosine distances and normalized magnitudes from vector representations generated from pairwise relative positions between joints. These representations were concatenated to form a 2D representation.

3.3 Methodology

As it has already been mentioned in s ubsection 3.1.1.1, our work has been partially inspired by the one of Jiang and Yin [114]. We first create an image by concatenating the aforementioned 75 1-D signals, which form the "signal" image. From each signal image we create an "activity" image, by applying one of the following transforms: a) the 2-D Discrete Fourier Transform (DFT); b) the 2-D Fast Fourier Transform (FFT); c) the 2-D Discrete Cosine Transform (DCT); and d) the 2-D Discrete Sine Transform (DST). Note that the aforementioned transforms provide 2D compact representations of 3D temporal skeletal motion. Moreover, these transforms are invariant to the position of the subject within the capture area, while they also consist an intermediate feature representation of the skeletal motion. In Fig. 3.7 we illustrate an example signal image and the 4 corresponding activity images.

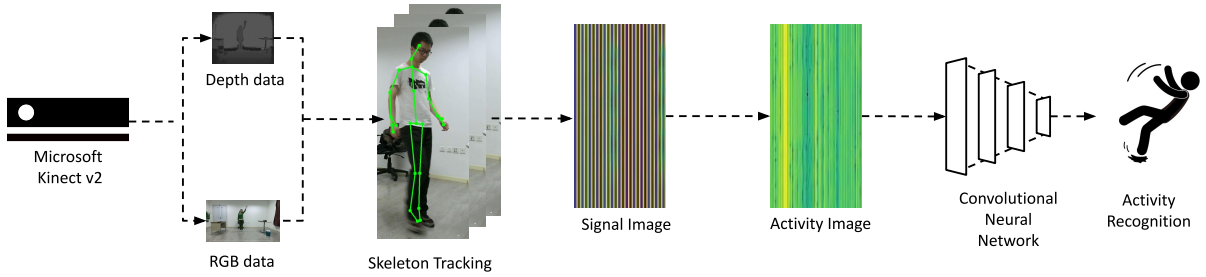


Figure 3.8: A graphic illustration of the proposed methodology for activity recognition. Specifically, 25 skeleton joints are tracked during a gesture. Then, a signal image is constructed. Upon application of the Discrete Cosine Transform, the activity image is constructed and given as input to the CNN which classifies it to the depicted class, i.e., *falling*.

We herein remind that the goals of this Thesis are limited to activity classification. Therefore, it belongs to the category of segmented recognition (see Section 1.2), as it does not perform a temporal segmentation step. Instead, we work using pre-segmented video sequences, aiming to only recognize the performed actions within each segment. We also assume that each segment contains exactly one action. The herein presented methodology is presented in Fig. 3.8.

3.3.1 Signal Images

The first step to create "signal" images is to concatenate the signals that are produced by skeletal motion. The process is the same as the one described in subsection 3.1.1.1. We herein remind that the motion of each skeletal joint in the 3D space, over time is treated as 3 independent 1D signals, while each of them corresponds to a coordinate. Therefore, for a given joint j, let $S_{j,x}(n), S_{j,y}(n), S_{j,z}(n)$ denote the 3 1D signals that correspond to its 3D motion and for the coordinates x, y, z, respectively. Thus, in the signal image, $S_{j,x}(n)$ corresponds to row $3 \times j - 2$. Accordingly, $S_{j,y}(n)$ and $S_{j,z}(n)$ correspond to rows $3 \times j - 1$ and row $3 \times j$. This way, the signal image $\mathbf{S}$ for a given activity and for N joints, is created upon concatenation of the $3 \times N$ signals, thus its dimension is $3 \times N \times T_s$, where T_s is the duration of this activity.

However, in real problems, since different subjects may perform the same activity with different duration and also different activities require different duration, it should be obvious that T_s is variable. In order to address the problem of temporal variability between subjects and between activities, so as to allow for signal concatenation, as discussed, a linear interpolation step is imposed. This way the length of all activities T_a is fixed. Note that in order to set value of T_a the process begins with the selection of a value close to the mean of all activities, yet the exact value that is finally used is chosen upon experimentation and fine-tuning. Thus, in the context of this work, the length is set to $T_a = 159$. Moreover, given $N = 25$, the dimension of $\mathbf{S}$ is equal to 75×159. An example signal image is illustrated in

Fig. 3.7a.

3.3.2 Activity Images

Let $\mathbf{S}$ denote a signal image as it was define in 3.1.1.2 with dimensions $W \times H$ and $\mathbf{S}_{nm}$ the pixel at coordinates (m, n). From each signal image an "activity" image $\mathbf{A}$ is created. This may be done by applying one of the following image transforms [76, 110] on a given signal image $\mathbf{S}$:

a)the 2-D Discrete Fourier Transform (DFT), which is defined as:

$$\mathbf{A}(u,v) = \sum_{x=0}^{W-1}\sum_{y=0}^{H-1} \mathbf{S}(x,y) e^{-j2\pi(ux/W+vy/H)} \ ,$$

where $x \in [0, W-1], y \in [0, H-1], u \in [0, W-1], v \in [0, H-1]$; b) the 2-D Fast Fourier Transform (FFT), which is a fast implementation of DFT; c) the 2-D Discrete Cosine Transform (DCT), which is defined as:

$$\mathbf{A}(u,v) = a_u a_v \sum_{x=0}^{W-1}\sum_{y=0}^{H-1} \mathbf{S}(x,y) \left(\cos\frac{\pi(2m+1)u}{2W} \cos\frac{\pi(2n+1)v}{2H}\right) \ , \tag{3.4}$$

where $x \in [0, W-1], y \in [0, H-1], u \in [0, W-1], v \in [0, H-1]$ and also:

$$a_u = \begin{cases} \frac{1}{\sqrt{W}}, u = 0 \\ \sqrt{\frac{2}{W}}, 1 \leq u \leq W-1 \end{cases} \tag{3.5}$$

and

$$a_v = \begin{cases} \frac{1}{\sqrt{H}}, v = 0 \\ \sqrt{\frac{2}{H}}, 1 \leq v \leq H-1 \end{cases} \ ; \tag{3.6}$$

d) the 2-D Discrete Sine Transform (DST), which is defined as:

$$\mathbf{A}(u,v) = a_u a_v \sum_{x=0}^{W-1}\sum_{y=0}^{H-1} \mathbf{S}(x,y) \left(\sin\frac{\pi(2m+1)(u+1)}{2W} \sin\frac{\pi(2n+1)(v+1)}{2H}\right) \ , \tag{3.7}$$

where $x \in [0, W-1], y \in [0, H-1], u \in [0, W-1], v \in [0, H-1]$ and a_u, a_v are given by Eqs. 3.5 and 3.6, respectively. Note that from each transform only the magnitude is preserved, while the phase is discarded. Also, note that DST and DCT are further processed by normalizing using the orthonorm. Obviously, in all cases the result is a 2-D image, with the same dimension as the signal image $\mathbf{S}$. Moreover, although FFT is a fast implementation of DFT, as it has been mentioned, it was amongst the goals of this work to assess if, due to the expected differentiation between them, any differences in performance would show in practice. In Fig. 3.7 a signal image is illustrated, along with the 4 corresponding activity images, created upon applying the aforementioned transforms on it.

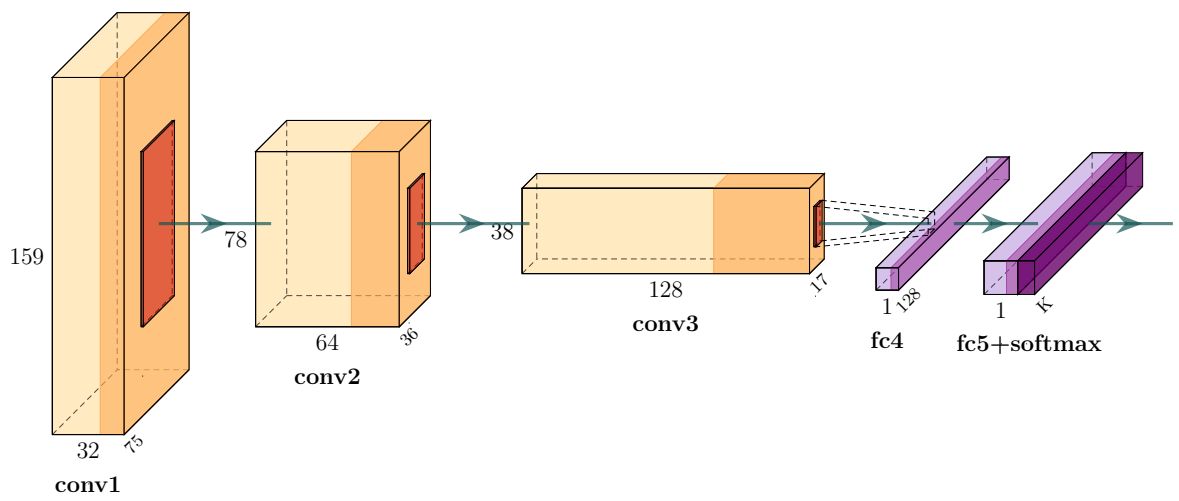


Figure 3.9: The proposed CNN architecture for recognition of human activities based on the 2D activity representation presented in this Chapter.

3.3.3 Network Architecture and Training

The architecture of the proposed CNN is presented in detail in Fig. 3.9. Specifically, the first convolutional layer filters the 159×75 input activity image with 32 kernels of size 3×3. The first pooling layer uses "max-pooling" to perform 2×2 sub-sampling. The second convolutional layer filters the 78×36 resulting image with 64 kernels of size 3×3. A second pooling layer uses "max-pooling" to perform 2×2 sub-sampling. A third convolutional layer filters the 38×17 resulting image with 128 kernels of size 3×3. A third pooling layer uses "max-pooling" to perform 2×2 sub-sampling. Then, a flatten layer transforms the output image of the last pooling to a vector, which is then used as input to a dense layer using dropout. Finally, a second dense layer produces the output of the network. To avoid overfitting, the most popular approach which is also adopted in this work is the use of the dropout regularization technique [256]: at each training stage several nodes are "dropped out" of the network. This way overfitting is reduced or even prevented, since complex co-adaptations on training data are prevented. For training the CNN, the ReLU activation function has been used. Moreover, the batch size has been to 100 and the Adam optimizer has been used. Also, the dropout was set to 0.55, the learning rate was set to 0.001 and the network was trained for 43 epochs, using the loss of the validation set calculated via cross-entropy as an early stopping method, in order to avert overfitting.

The aforementioned architecture has been selected through validation set tuning based on two factors: a) the need to build sufficiently rich representations to allow for effective classification; and b) the restriction of the number of parameters so as to allow flexibility, e.g., for easy deployment of the model in low-cost platforms or mobile devices to perform inference on the edge, in real-life applications of the herein proposed approaches.

3.4 Experimental Protocol and Results

For the experimental evaluation of our approach we used the PKU-MMD dataset [163] that has been presented in subsection 2.4.1.1. As it has already been mentioned, PKU-MMD is a large-scale benchmark dataset, focusing on human action understanding and containing approx. 20K action instances from 51 categories, while 66 human subjects have participated in the data collection process. Moreover, each action has been recorded by 3 camera angles, using the Microsoft Kinect v2 camera. For each action we use the provided extracted 3D positions of skeletons, discarding RGB, depth and infrared information.

Our experiments are divided into two parts. In the first part our goal was to assess whether the proposed approach may be used for ambient assistive living scenarios and more specifically for the recognition of ADLs. Therefore, we selected 11 out of the 51 classes of PKU-MMD, which we believe are the most close to ADLs or events in such a scenario. The selected classes are: *eat meal snack*, *falling*, *handshaking*, *hugging other person*, *make a phone call answer phone*, *playing with phone tablet*, *reading*, *sitting down*, *standing up*, *typing on a keyboard* and *wear jacket*. In Fig. 3.10 we illustrate sample signal and activity images from these 11 classes and for all types of transforms. Note the visual difference of these images which may not be significant to the human observer in some cases, especially in the cases of FFT and DCT, yet allows the CNN to learn the differences between two classes. In the second part, we performed experiments with the whole dataset, i.e., with all 51 classes.

The evaluation protocol we followed is as follows: we first performed experiments per camera position; we remind that in this case both training and testing sets derived from the same viewpoint. Then, we performed cross-view experiments, where different viewpoints were used for training and for testing. As we have already mentioned, the goal of these experiments was to test the robustness of the proposed approach in terms of transformation (e.g., a translation and a rotation), which could correspond to abrupt viewpoint changes which typically occur in real-life situations. Finally, we performed cross-subject experiments, where subjects were split in training and testing groups, i.e., any actor "participated" only into one of the groups. The goal of this experiment was to test the robustness of our approach into intra-class variations. In real-life situations this is expected to happen when a system is trained e.g., within a laboratory environment and is deployed into a real ambient-assistive living environment where different subjects are observed. Note that in all cases we measured classification accuracy. Detailed results are depicted in Tables 3.2 and 3.3 for 11 and 51 classes, respectively.

As it may be observed, in the case of 11 classes, DST showed best accuracy for the majority of single- and cross-view experiments, followed by DCT. However, in the cross-subject case, DFT showed best accuracy, once again followed by DCT. In single-view experiments, and in case of the middle (M) camera, accuracy varied between 0.83 – 0.89, in case of the left (L) camera, it varied between 0.76 – 0.84 and in case of the right (R) camera, it varied between 0.84 – 0.89. In all cases and bearing in mind the early stage of research, they are considered as satisfactory. Similarly, in cross-subject experiments, accuracy varied between 0.81 – 0.85, which was also a satisfactory result. However, in

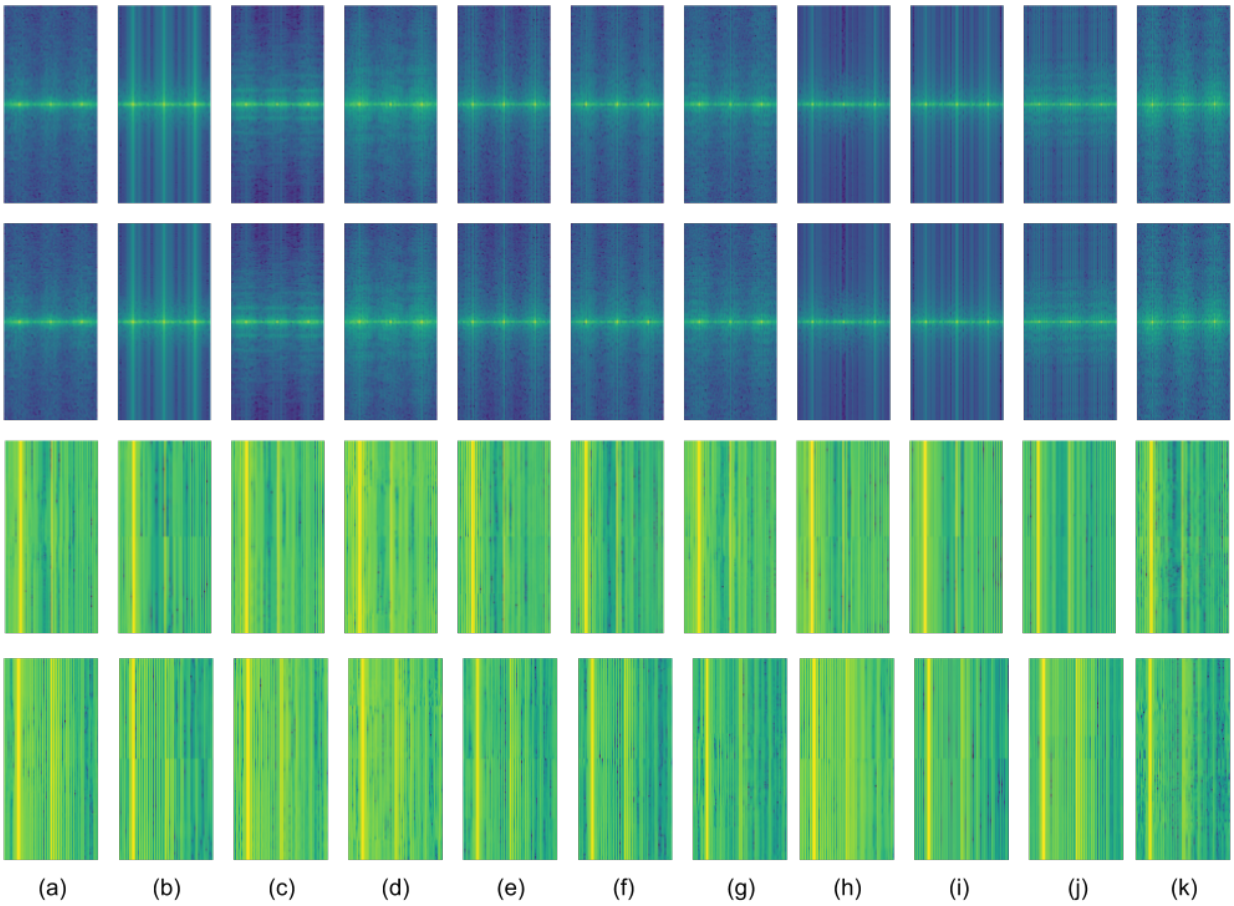


Figure 3.10: Examples of activity images from 11 classes and for the 4 transforms used. 1st row: DFT; 2nd row: FFT; 3rd row: DCT; 4th row: DST. a) eat meal/snack; b) falling; c) handshaking; d) hugging other person; e) make a phone call/answer phone; f) playing with phone/tablet; g) reading; h) sitting down; i) standing up; j) typing on a keyboard; k) wear jacket. DFT and FFT images have been processed with log transformation for visualization purposes. Figure best viewed in color.

the cross-view case we noticed a severe drop of performance, compared to the previous experiments. Specifically, when using one camera for training and another for testing, our observations may be divided into two main aspects: a) in cases of "neighbouring" cameras, i.e., L, M and R, M, accuracy varies between 0.56 – 0.72, being inferior to cross-view and cross-subject cases; b) in cases of "extreme" viewpoint changes, i.e., when using L and R cameras for training/testing or vice versa, accuracy significantly drops between 0.32 – 0.43. Finally, in case of using two cameras for training and the third for testing, notably when L and R cameras are used for training, accuracy is between 0.77 – 0.81, while in the other two cases, it lies significantly lower, i.e., between 0.57 – 0.62.

Continuing with the case of 51 classes, our observations are similar, yet all accuracies are significantly lower compared to the case of 11 classes, as expected. Specifically,

Table 3.2: Experimental results denoting accuracy of the proposed approach in the 11 selected classes of the PKU-MMD dataset. M, L and R denote the middle, left and right camera angles, respectively. Bold denotes best result.

Experiment	Train	Test	DFT	FFT	DCT	DST
Single-view	M	M	**0.89**	0.84	0.83	0.86
	L	L	0.76	0.82	0.78	**0.84**
	R	R	0.84	0.85	**0.89**	0.87
Cross-view	M	L	0.62	0.61	0.63	**0.64**
	M	R	0.58	0.61	**0.65**	0.63
	L	M	0.65	0.66	0.64	**0.72**
	L	R	0.41	0.38	0.32	**0.43**
	R	M	0.56	0.59	**0.64**	0.63
	R	L	0.32	0.37	0.33	**0.39**
	M,L	R	0.60	0.60	0.62	0.62
	M,R	L	0.60	0.57	0.59	**0.60**
	L,R	M	0.77	0.77	0.81	**0.82**
Cross-subject	M,L,R	M,L,R	**0.85**	0.79	0.83	0.81

DCT and DST exhibited once again best accuracy in the majority of cases, apart from the extreme cross-view cases where L camera was used for training and R for testing or vice versa, where DFT showed best accuracy, followed by FFT. Specifically, in single-view experiments, and in case of the middle (M) camera, accuracy varied between 0.46 – 0.66, in case of the left (L) camera, it varied between 0.46 – 0.66 and in case of the right (R) camera, it varied between 0.51 – 0.69. Notably, DFT and FFT exhibited significantly inferior performance to DCT and DST. Similarly, in cross-subject experiments, accuracy varied between 0.50 – 0.64, while once again DFT and FFT exhibited inferior performance. In the case of 51 classes and contrary to the one of 11 classes, performance in the cross-subject case was comparable to the single-view case. Finally and in the cross-view case, when using one camera for training and another for testing, our observations may be again divided into two main aspects: a) in cases of "neighbouring" cameras, i.e., L, M and R, M, accuracy varies between 0.30 – 0.39, being inferior to cross-view and cross-subject cases; b) in cases of "extreme" viewpoint changes, i.e., when using L and R cameras for training/testing or vice versa, accuracy significantly drops between 0.13 – 0.22. Finally, in case of using two cameras for training and the third for testing, notably when L and R cameras are used for training, accuracy is between 0.44 – 0.55, while in the other two cases, it lies significantly lower, i.e., between 0.32 – 0.34.

3.5 Conclusions

In this Chapter our primary goal was to propose a novel approach for 2D representation of 3D motion of skeletal joints, so as to recognize human activity using CNNs. To this, we first presented a simple arm gesture recognition experiment, whose goal was to verify whether the proposed representation had the potential to be applied into a real-life prob-

Table 3.3: Experimental results denoting accuracy of the proposed approach in the 51 classes of the PKU-MMD dataset. M, L and R denote the middle, left and right camera angles, respectively. Bold denotes best result.

Experiment	Train	Test	DFT	FFT	DCT	DST
Single-view	M	M	0.46	0.49	**0.66**	0.65
	L	L	0.46	0.52	0.60	**0.65**
	R	R	0.55	0.51	**0.69**	0.67
Cross-view	M	L	0.33	0.34	0.33	**0.36**
	M	R	0.32	0.31	**0.36**	0.32
	L	M	0.33	0.30	0.35	**0.39**
	L	R	**0.22**	**0.22**	0.13	0.14
	R	M	0.32	0.34	**0.36**	0.34
	R	L	0.20	0.19	0.13	0.14
	M,L	R	0.33	**0.34**	0.33	**0.34**
	M,R	L	0.32	0.33	0.32	**0.34**
	L,R	M	0.44	0.44	**0.55**	0.52
Cross-subject	M,L,R	M,L,R	0.50	0.52	**0.64**	0.63

lem. Upon the initial, encouraging results we opted to apply this methodology into large scale, challenging human activity recognition datasets.

Specifically the proposed methodology is based on a novel representation of skeletal 3D motion which uses spectral image transformations of signal images, resulting upon a) concatenation of raw 1D signals corresponding to 3D motion of skeletal joints' coefficients; and b) application of a transform to the created image. For classification we used a CNN architecture.

We evaluated the proposed approach using a state-of-the-art and challenging dataset, which consisted of sequences corresponding to 51 human actions. These sequences had been captured with 3 Kinect v2 cameras, under different camera angles and the skeletal joints of the human actors involved had been extracted. We performed experiments involving either a single camera (single-view) or more than one (cross-view). We also performed cross-subject experiments to evaluate the robustness of the approach. We mainly focused on a subset of 11 actions which in our opinion are the most close to real-life ADLs. However, we also experimented with the whole dataset. Our results indicate that the proposed approach may be successfully applied to human action recognition in real-like conditions, yet a drop of performance is expected when significant changes of viewpoint occur.

4. GEOMETRIC DATA AUGMENTATION FOR HAR

4.1 Introduction

In Chapter 3 we presented a novel human activity recognition approach that was applied on segmented video data and was based on a deep CNN model. This model was trained on 2D image representations of 3D skeletal motion, which was presented in Subsection 3.3. However, this approach demonstrated reduced performance in case where training and testing data derived from different cameras, i.e., in the cross-view evaluation. To deal with this drop of performance, in this Chapter we enhance this methodology by applying a geometric rotation transform to action examples. Its goal is twofold: a) to augment the available dataset by increasing both the number and the diversity of the data samples and b) to "align" action examples that originate from different cameras, so as to compensate for the corresponding viewpoint change.

As we have already discussed in subsection 2.2, data augmentation is a generic methodology that aims to expand the size and/or the diversity of some data set, in an effort to improve training and subsequently increase the performance in evaluation. This is typically achieved upon creating "artificial" data samples, which are quite similar to those of the original set, yet not identical to any of them. In the context of computer vision problems, usually data are images, thus data augmentation aims to construct synthetic images, based on the properties and limitations of the given task. Data augmentation approaches in HAR problems have employed local averaging and sampling [258], transformations such as rotation, scaling, jittering [117], and data warping [91]. However little work has been demonstrated when working with skeletal data such as the ones we use in the context of this Thesis. In that case, augmentation approaches include direct application of geometric transformations [169] or RNNs [325] in raw skeletal sequences. Nevertheless, and as we have previously mentioned in subsection 2.2, such approaches are typically hard to implement, and are mostly domain-dependent, as they are specific to a particular problem/task.

The novelty of this Chapter is a data augmentation scheme that is applied on raw skeletons, so as to enhance training data set with artificial examples. Moreover, we assess the effect of data augmentation to the classification performance of the HAR methodology presented in Chapter 3. We show that by incorporating artificially rotated skeletons to the training dataset, the accuracy of a deep approach within a multi-camera setup may be boosted. Of course, due to the methodology we use for the construction of image representations of skeletal data, popular data augmentation strategies such as rotations and random crops may not be applied; such methods could severely affect the spectral properties of activity images, by removing or distorting important information. Thus, rotation of activity samples of skeletal motion upon geometric processing them which led to the creation of artificial, valid examples has been instead adopted.

The rest of this Chapter is organized as follows: Section 4.2 presents related research works that use a data augmentation step on human skeletons. The proposed data aug-

mentation methodology is presented in detail in Section 4.3. The experimental protocol we followed is presented in Section 4.4, wherein results are also presented and discussed. Finally, conclusions are drawn is Section 4.5.

4.2 Related Work

As we have already mentioned in section 2.2.1, data augmentation aims to expand the size and/or the diversity of a given data set, by constructing some kind of synthetic images and by considering any properties/limitations of the problem at hand. Although a plethora of research works has addressed the problem of HAR, as presented in Section 1.4, only few research efforts have addressed the problem of viewpoint invariance. In the context of HAR, data augmentation has been previously used with inertial measurements extracted from wearable sensors by Eyobu and Han [258], who applied local averaging as a down-sampling technique and shuffling. Also, using similar data, Kalouris et al. [117] applied a set of domain specific transformations, such as rotation, scaling, jittering etc. Hernandez et al. [91] worked with hand points and applied data warping on the magnitude and the temporal location of motion signals.

Moreover, the aforementioned problem of viewpoint invariance has also not been adequately addressed by intense research efforts. Liu et al. [169] proposed the direct application of geometric transformations to raw skeletal sequences. They created a 5D joint representation by concatenating 3D space coordinates, time and joint label. Then, in order to create a 2D image, they projected 2 of the aforementioned 5 dimensions, while the remaining 3 were mapped to R, G and B color values. Obviously, the resulting images were pseudo-colored. Moreover, we should emphasize that the proposed data augmentation approach has been partially inspired by the work of Zhang et al. [325]. Therein, a view adaptive RNN, was used in order to apply geometric transformations to raw skeletons captured under several views, towards the selection of more “consistent” viewpoints.

4.3 Methodology

In this Section we present the proposed methodology for data augmentation and viewpoint alignment. Our approach is based on the 3D rotation transformation, which is directly applied on the 3D skeletal data, so as to artificially modify camera viewpoint. Specifically, our main goals are to assess the effects of both data augmentation and viewpoint alignment in the human recognition task which was presented in Chapter 3.

4.3.1 Geometric Transformations on 3D skeletons

To begin with, let us consider a skeleton, comprising a set of 3D joints, as illustrated in Fig. 4.4c. In this example, a camera is directly facing the skeleton. This is the ideal

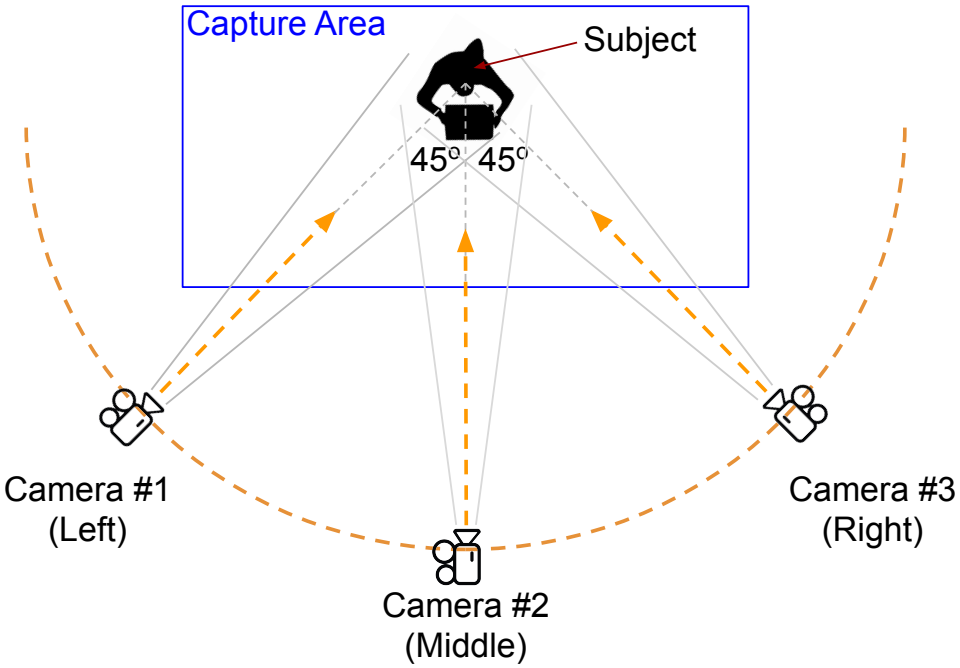


Figure 4.1: Camera setup of the PKU-MMD dataset. Three cameras are placed in front of the subject, capturing her/him under three viewpoints, under angles $\theta = 0° \pm 45°$ Cameras $\#1$, $\#2$, $\#3$ will be denoted as L, M, R, respectively.(This Figure is the same as Fig. 2.24, and is herein used for convenience.)

case in HAR tasks. However, in practical applications, subjects do not necessary face the camera during the execution of activities. As it has been previously discussed in Subsection 2.4.1, contemporary large scale datasets, such as the PKU-MMD [163], which we will use throughout the experimental evaluation of this approach, adopt a multi-camera setup, aiming to simultaneously capture the subject under different viewpoints. Specifically, in the case of PKU-MMD a camera setup that is composed of three cameras is used. One camera is placed directly at the front of the test subject, while the other two are placed at its left and right, respectively, as it may be seen in Fig. 4.1.

To deal with viewpoint alignment, and given the aforementioned camera setup, an obvious approach would be to apply an appropriate transformation on one camera, so that a pair of cameras is aligned, i.e., they have the same viewpoint. Under the assumption that all cameras are placed at the same distance to the test subject, i.e., at the perimeter of an imaginary circle, each camera may be aligned to any other with a simple geometric rotation transformation. Such a transformation by an angle θ, in the y-axis may described by [272]:

$$\mathbf{R}_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} . \tag{4.1}$$

Specifically, given a joint $\mathbf{J}_i$, rotation by an angle θ, in the y-axis transforms it to $\mathbf{J}'_i$, as

follows:

$$\mathbf{J}'_i = \mathbf{R}_y(\theta) \cdot \mathbf{J}_i \ , \tag{4.2}$$

where

$$\mathbf{J}_i = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \text{ and } \mathbf{J}'_i = \begin{bmatrix} x'_i \\ y'_i \\ z'_i \end{bmatrix} \tag{4.3}$$

4.3.2 Skeleton Alignment and Data Augmentation

Therefore, to align any two given skeletons captured from different cameras, simultaneously, we may rotate each 3D joint of one of them by an angle θ, about the y-axis, complying to the Cartesian 3D coordinate system used by Kinect v2. Of course, the angle θ of the rotation, depends on the initial camera position setup and on which two cameras are used for training and testing. For example, to align a camera placed at the subject's left, at an angle θ_L and a camera placed at the subject's right, at an angle θ_R, we should apply $\mathbf{R}_y(\theta_R - \theta_L)$. Specifically, in the setup illustrated in Fig. 4.1, to align right camera with middle camera, the transform needed is $\mathbf{R}_y(\theta_L - \theta_M) = \mathbf{R}_y(45^\circ - 0^\circ) = \mathbf{R}_y(0^\circ)$. We will use this skeleton alignment approach, to assess whether the classification accuracy of our approach in the cross-view case may be improved when skeletons are aligned prior to classification. A visual overview of the proposed alignment approach is illustrated in Fig. 4.2.

For data augmentation, our goal is to assist the training procedure of our network, by augmenting the data set used. To this, since for each skeleton sample we are now able to produce 4 rotated instances, then, given a certain camera viewpoint, we aim to provide more reliable recognition of actions by training with an augmented data set, which includes all aligned images resulting from the aforementioned rotated instances. More details regarding the evaluation protocols we followed will be presented in subsection 4.4. A visual overview of the proposed augmentation approach is illustrated in Fig. 4.3, while in Fig. 4.4 we illustrate a skeleton which is rotated by all angles used throughout our experiments.

4.3.3 CNN Architecture

For classification we have used the CNN architecture that has been firstly presented in subsection 3.3.3 and is illustrated in Fig. 3.9. As it has been shown in Chapter 3, this architecture was successfully applied to the problem of HAR using image representations of skeletal sequences. Therefore, since we also base the herein presented approach to the same representations, we also use the same network application of the evaluation of the data augmentation approach. As in Chapter 3, the ReLU activation function has been used, the batch size has been to 100, the Adam optimizer has been used, the dropout was

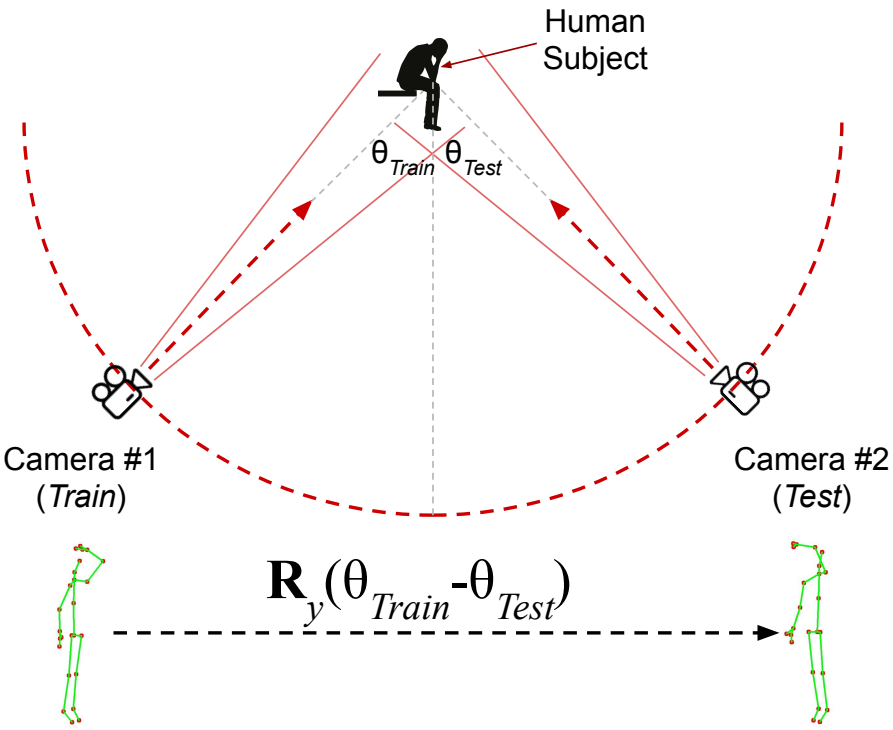


Figure 4.2: A visual overview of the proposed data alignment methodology.

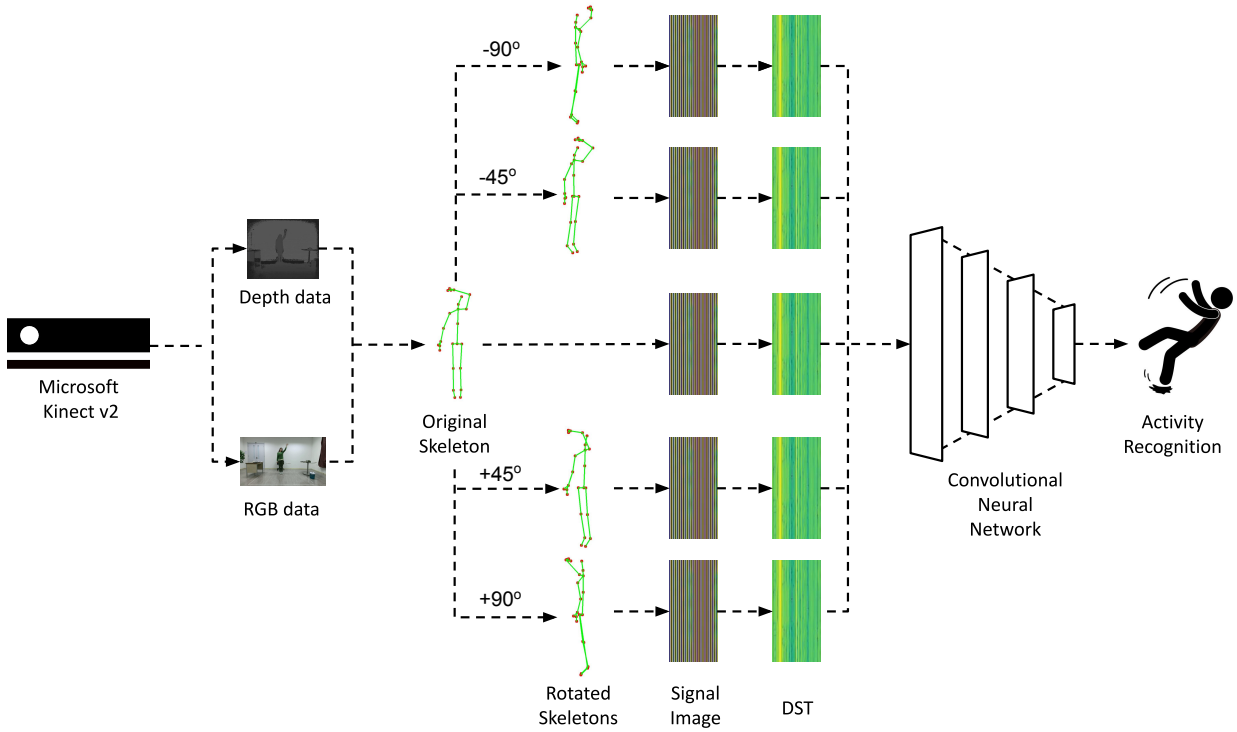
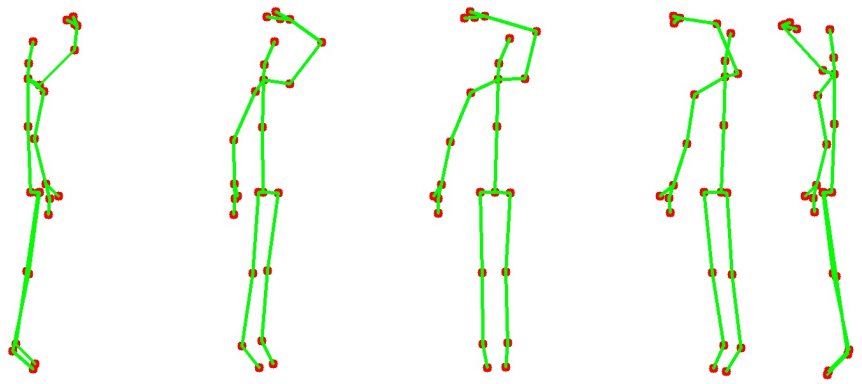


Figure 4.3: A visual overview of the proposed data augmentation methodology.



(a) (b) (c) (d) (e)

Figure 4.4: skeleton rotated by angle θ: a) $\theta = 90°$, b) $\theta = 45°$, c) $\theta = 0°$ (raw skeleton), d) $\theta = -45°$, e) $\theta = -90°$. For illustrative purposes, depth information, i.e., z-coordinate has been discarded.

set to 0.55, the learning rate was set to 0.001. The network was trained using the loss of the validation set calculated via cross-entropy as an early stopping method.

4.4 Experimental Protocol and Results

Based on the experimental results that were presented in Chapter 3, in this Chapter we opt to evaluate the herein proposed approaches using the DST transformations of activity images, since this demonstrated overall best performance. Therefore we create activity images simply by transforming the raw signal image by applying the DST transform. Contrary to Chapter 3 we include the extra step of rotation for aligning skeletal information prior to applying a transform. Note that due to data augmentation we use up to 5 times more data for training and testing.

As in Chapter 3, we also investigate whether the proposed approach may be used for ambient assistive living scenarios and more specifically for the recognition of ADLs. Therefore, we performed experiments using the following 11 classes of PKU-MMD, as follows: *eat meal snack*, *falling*, *handshaking*, *hugging other person*, *make a phone call*, *answer phone*, *playing with phone-tablet*, *reading*, *sitting down*, *standing up*, *typing on a keyboard* and *wear jacket*. We then perform experiments with the whole dataset, i.e., with all 51 classes. In all cases we worked only based on the skeletal data, discarding RGB, depth and infrared information.

In this Chapter, the evaluation protocol we followed concerns the following cases:

i. *Cross-view experiments*: at least two different camera viewpoints were used for training and only one viewpoint for testing or 2 different viewpoints were used, one for training and one for testing, e.g., *L* used for training, *M* or *R* used for testing.

ii. *Cross-subject experiments*: subjects were split in training and testing groups, i.e., each one was a member of exactly one of these groups.

We remind that the third case are single-view experiments, wherein both training and testing sets derived from the same viewpoint e.g., *L* used for training and testing purposes. However, obviously such a case is not applicable, since it does not involve skeleton alignment.

The main difference between the evaluation conducted in Chapter 3 and the herein presented one lies in the training and testing sets that have been used. Based on the camera setting of the PKU-MMD data, and the coordinate system used by the Microsoft Kinect v2 camera, we opted for signal image rotations as discussed in subsection 4.3.2. More specifically, we performed rotations using $\theta \in \{\pm 45^{\circ}, \pm 90^{\circ}\}$. This means that *L* signal images rotated by $+45^{\circ}$ and $+90^{\circ}$, align to *M*, *R*, respectively, *M* signal images -45° and $+45^{\circ}$ align to *L*, *R*, respectively, while *R* signal images rotated by -45°, -90° align to *M*, *L*, respectively. Obviously, the number of available images is multiplied by 5.

Our goal is to use this augmented data set, so as to accomplish an increase in performance, when comparing to the results of Chapter 3, which are herein used as a baseline, for comparison purposes. In other words, given a certain camera viewpoint, we aim to provide more reliable recognition of actions upon training with an augmented data set, which includes all aligned images resulting from the aforementioned rotations. However, the following obvious question rises from the aforementioned statement: What if the given camera setting is *unknown*, while we have no knowledge regarding which camera viewpoint(s) has(have) been used for training and from which viewpoint originates a given testing sequence? Bearing this question in mind, our decision was to expand our protocol by applying all the aforementioned rotations to all the original signal images. More specifically, our experiments were organized as follows: we used the augmented set of samples deriving from a single or multiple cameras for training. For testing, we performed separate experiments using original samples and augmented ones for the remaining cameras.

Moreover, we further heuristically modified the proposed approach as follows: upon performing all the aforementioned alignments, for each case a probability per each activity is estimated. We then select as the predicted activity the one that demonstrates the maximum value, among all alignment cases. Therefore, in this case the output of the proposed approach is the most “confident” decision of all classifiers.This modified evaluation approach is illustrated in Fig. 4.5.

Furthermore, in Table 4.1 we present cross-view results of the herein proposed methodologies, compared to the approach of Chapter 3, which is herein considered as a baseline. Also, in Table 4.2 we present cross-subject results. For each experiment, we calculated the classification accuracy. Note that each experiment was performed 10 times, and in all Tables we depict mean value of these accuracies.

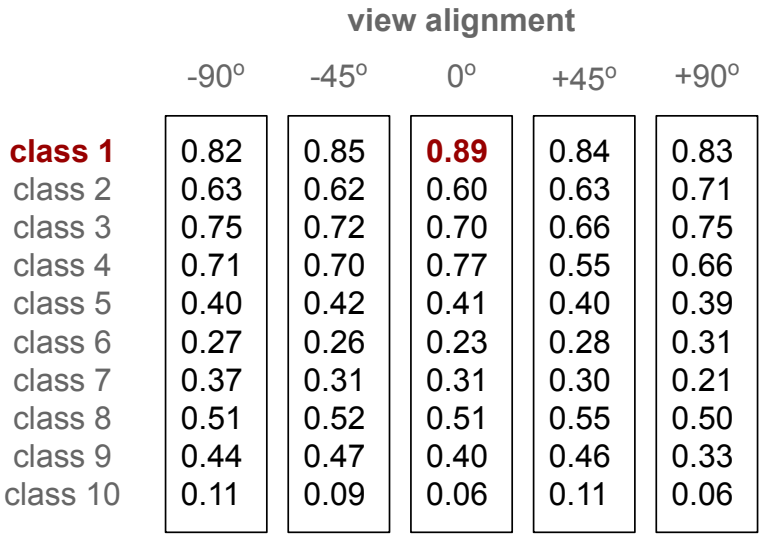


Figure 4.5: A visual overview of the heuristically modified methodology. In this artificial example we consider an imaginary testing sample, 10 possible classes and the 5 alignments presented in subsection 4.4. All classification confidences are first estimated (i.e., for all 10 classes). Then, we consider as classification output the class with the maximum confidence score, among all, i.e., in our example the imaginary testing sample is classified as "class 1" with 0.89 confidence.

Table 4.1: Cross-view experimental results of the geometric data augmentation approach presented in this Chapter. In all cases, results denote classification accuracy. By B, VA1 and VA2 we denote the baseline approach (i.e., the approach of Chapter 3), the data augmentation and view alignment approach and its modification wherein the output of the most confident classifier is considered, respectively. Differences (Diff.) are calculated as $\%$ percentages. Note that in B the original samples have been used for training and testing, while in VA1 and VA2 the augmented samples have been used for training and testing.

Train	Test	Part	B	VA1						Diff. B, VA1 (%)	VA2	Diff. VA1, VA2 (%)
				$-90°$	$-45°$	$0°$	$+45°$	$+90°$	mean			
L	M	11	0.72	0.82	0.86	0.93	0.93	0.86	0.88	+22.2	0.91	+3.4
		51	0.39	0.56	0.61	0.72	0.76	0.66	0.66	+69.2	0.76	+15.2
L	R	11	0.43	0.59	0.65	0.85	0.78	0.55	0.68	+59.1	0.81	+19.1
		51	0.14	0.34	0.33	0.64	0.49	0.27	0.41	+192.9	0.57	+39.0
M	L	11	0.64	0.84	0.84	0.85	0.90	0.91	0.87	+35.6	0.92	+5.8
		51	0.36	0.59	0.58	0.60	0.72	0.73	0.64	+77.8	0.74	+15.6
M	R	11	0.63	0.83	0.87	0.90	0.89	0.87	0.87	+38.4	0.89	+2.3
		51	0.32	0.68	0.61	0.75	0.71	0.63	0.68	+112.5	0.76	+11.8
R	L	11	0.39	0.59	0.53	0.69	0.83	0.86	0.70	+79.5	0.85	+21.4
		51	0.14	0.28	0.25	0.38	0.52	0.65	0.42	+200.0	0.55	+31.0
R	M	11	0.63	0.93	0.82	0.88	0.93	0.87	0.89	+40.6	0.94	+5.6
		51	0.34	0.75	0.57	0.64	0.71	0.61	0.66	+94.1	0.72	+9.1

Beginning with the results of Table 4.1, we should notice the following: in all cases, the

Table 4.2: Cross-subject experimental results of the geometric data augmentation approach presented in this Chapter. In all cases, results denote classification accuracy. By B and VA1 we denote the baseline approach (i.e., the approach of Chapter 3) and the data augmentation and view alignment approach, respectively. Differences (Diff.) are calculated as $\%$ percentages. Note that in B the original samples have been used for training and testing, while in VA1 the augmented samples have been used for training and testing.

Part	B	VA1	Diff. (%)
11	0.81	0.95	+17.28
51	0.63	0.76	+20.63

mean accuracy of VA1 experiments is significantly higher than the corresponding accuracy of the baseline experiment, for both 11 and 51 classes of the PKU-MMD dataset. Notably, in the extreme cases of experiments using L and R cameras for training and testing, or vice versa, performance has increased between 59.1% and 80.9%. However, despite this increase of performance, in those cases the achieved accuracy still remains lower, compared to the other cases of experiments, being equal to 0.68 and 0.41 when using L for training and R for testing in cases of 11 and 51 classes, and equal to 0.70 and 0.42, respectively, when using R for training and L for testing. For comparison reasons, the lowest accuracy among all other experiments was 0.65 when training with M and testing with L for the 51 cases. Moreover, in every case of VA2 experiments, an increase of performance was observed. Once again, in the extreme cases of experiments using L and R cameras for training and testing, or vice versa, performance has increased between 19.1% and 31.0%.

Continuing with the cross-view experiments of Table 4.2, we may observe that performance in case of 11 classes reached 0.95, upon an increase of 17.28%, while a slight increase of 4.11% was also observed in case of 51 classes, reaching a value of 0.76.

Generally, experimental cross-view results indicate that: a) in every case, a significant boost of performance is achieved, compared to the baseline approach of Chapter 3; b) in most cases, best accuracy is achieved when the testing viewpoint is "aligned" with the training one, e.g., when L is used for training, while M_{+45° for testing; and c) when we use the most confident classifier decision to measure accuracy, we constantly observe an increase of accuracy compared to the mean accuracy of the view alignment approach. Moreover, a significant boost of performance is also observed in cross-subject case. Therefore, we may argue that experiments justify our expectations regarding both data augmentation and viewpoint alignment.

4.5 Conclusions

In this Chapter we presented a novel approach which aims to recognize human actions in video sequences based on a representation of skeletal 3D motion, using spectral image transformations, while classification is performed by a CNN. Contrary to Chapter 3,

herein the model was trained on images that resulted upon a) concatenation of raw 1D signals corresponding to 3D motion of skeletal joints; and b) application of a transform to the created image. A further pre-processing, i.e., a geometric rotation was imposed and significantly improved the accuracy while providing robustness to our approach. This approach was evaluated using a state-of-the-art, challenging human action dataset, for cross-view and cross-subject cases, so as to assess its robustness and experimental results indicated that it may be successfully applied to real-like tasks, such as the ones encountered e.g., in assistive living environments. We also experimentally verified that the proposed alignment step leads to an increase of accuracy. More specifically, we demonstrated a boost of performance in the cross-view and single-view cases, achieving results comparable to when the same camera viewpoint is used for training and testing.

5. DOMAIN ADAPTATION FOR HAR

5.1 Introduction

Domain adaptation [15] refers to a broad class of techniques which fall into the research area of *transfer learning*. In general, supervised learning techniques rely, fundamentally, on the assumption that both train and test datasets are drawn from the same distribution. However, as it has already been mentioned in subsection 2.2.2, some distributional change between train and test data breaks this assumption even if the underlying conditional distribution of labels corresponding to features of the input space $P(Y|X)$ is the same for both sets. This is known as "domain"/"covariate" shift. As a result the model is expected to under-perform on the test set [208]. Domain adaptation aims to develop techniques for mitigating the domain/covariate shift problem. In the area of computer vision, factors such as illumination, viewpoint changes and even different acquisition devices render the data collection process prone to domain/covariate shift and for this reason many research efforts on domain adaptation have focused on this area, yielding a plethora of techniques [223].

In Chapter 4 we presented and evaluated a view augmentation approach which was applied to the problem of human activity recognition. However, although contemporary recognition approaches typically aim to surpass the aforementioned problem of lack of training data through augmentation [280] of the available ones, i.e., by constructing "synthetic" data, in a way that the global distribution of the input features is better represented e.g., by adding some kind of noise, translating/rotating the input data, cropping the scene or by using some domain specific heuristic [48, 151], such hand-crafted data synthesizers are typically hard to implement, while in most cases they are problem-specific and/or domain-dependent. One could argue that for any given application, a large corpus of data can be collected ad hoc in order to perform supervised training. However, the collection process is time-consuming and expensive, with the main bottleneck being the process of data annotation. Typically, a realistic use case is to have very few labeled data in an otherwise unlabeled dataset. Also, data augmentation is often criticized as being largely problem dependent and for this reason techniques should be viewed on a per problem basis.

Domain adaptation on the other hand leads to more general methods that can easily be applied across different problems. In [324], kernel PCA is used to transfer knowledge between domains in a semi-supervised manner in the context of video action recognition, by extracting key frames using a shot boundary detection algorithm. Many adversarial neural network techniques have been utilized for domain adaptation including the works in [3,278] where a fully unsupervised approach is followed to tackle problems in image classification and natural language processing, and works like [28, 29, 31, 103, 326] where the standard domain adaptation framework is extended to cover more challenging knowledge transfer problems within these fields. These works highlight the flexibility of adversarial techniques for knowledge transfer, since they demonstrate applications in various problems with little

to no adjustments.

In this Chapter, we propose a different approach for addressing the aforementioned issues, which is based on adversarial domain adaptation algorithms. More specifically, we show that adapted representations can be retrieved *automatically* to perform inference on a *sparsely* labeled dataset, using a model that has been trained on a related labeled dataset. That is, the novelty of this Chapter is the introduction of a technique for tackling in a semi-supervised setting, the problem of classifying actions from video obtained with one form of measurement bias using a model trained on data where a different form of measurement bias is present. As such, our technique can be utilized to generalize classifiers to new forms of measurement bias. We perform experiments where we demonstrate that generalization between different viewpoints can be achieved without data augmentation.To the best of our knowledge, the herein presented methodology is the first to consider regularizing model training using a domain confusion term in the objective for producing classifiers that are robust to altered measurement environments and subjects for action recognition from video data. In addition, we apply our proposed approach to tackle different viewpoint scenarios. This is an application that is very important in e.g., assisted living environments since ensuring similar viewpoints for different sites is impractical or even impossible.

The rest of this Chapter is organized as follows: in Section 5.2 we present related work in the broader area of domain adaptation. Then, in Section 5.3 we present the proposed domain adaptation methodology. The experimental protocol we followed is presented in Section 5.4, wherein experimental results are also presented and discussed. Finally, conclusions are drawn in Section 5.5.

5.2 Related Work

5.2.1 Domain Adaptation

As it has been already discussed, domain adaptation is a sub-field of transfer learning which aims to mitigate the domain/covariate shift problem when training a classifier. To make this more precise, consider, as it is typical in domain adaptation literature [41], a classification problem as a tuple (D, T), where D is called the *domain* and T the *task*. The domain is in turn a tuple $(\mathcal{X}, P(X))$, where $\mathcal{X}$ is the space in which the covariates (i.e., the input feature representations) take values and $P(X)$ is the distribution of the covariates over $\mathcal{X}$. The task is another tuple $(\mathcal{Y}, P(Y|X))$, where $\mathcal{Y}$ is the *label space* and $P(Y|X)$ is the conditional distribution of labels given values of the covariates. In standard supervised learning we are typically interested in approximating $P(Y|X)$, i.e. obtain a predictor (classifier) for T. In domain adaptation, the setup usually consists of two problems (D_s, T_s), (D_t, T_t), respectively called the source and target problems, such that $T_s = T_t$ but $D_s \neq D_t$, i.e., sharing the same labels, while their features and their distributions are both different. In this work we focus on homogeneous domain adaptation where we further assume that $\mathcal{X}_s = \mathcal{X}_t$, but $P(X_s) \neq P(X_t)$, i.e., input features are the

same, while their distributions are different.

The domain adaptation problem is to utilize (D_s, T_s) or a predictor for T_s so as to obtain or improve a predictor for T_t. This is particularly useful to consider in a learning scenario where the (unlabeled or sparsely labeled) test set is available during training. In general two general approaches are followed; the first involves importance sampling techniques, while the second involves learning a representation for source and target data in which the covariate-shift problem is resolved, i.e. where the distribution of source and target data is the same [41]. Deep learning algorithms for obtaining such representations have been widely explored yielding many successful methods [296], especially in the field of computer vision [41]. In particular, adversarial neural networks have proven to be an effective tool for mitigating the covariate-shift problem and have been successfully employed in multiple works [3, 217, 278]. Such methods, are flexible and well suited for the domain adaptation task because they allow for the automatic extraction of representations, where the distributions of source and target data are the same, using simple iterative optimization algorithms. In this work we focus on adversarial domain adaptation techniques for our evaluation of domain adaptation in computer vision tasks.

Adversarial domain adaptation schemes were inspired by the distribution alignment approach presented in [77], within the context of generative adversarial networks. A general abstract scheme that breaks down such methods is presented in [278]. Typically, two representation extracting networks $M_s,\ M_t$ are defined, along with a domain discriminator network D and a source classifier C which discriminates source data instances based on the representation extracted from M_s. The source networks M_s and C are either jointly trained on source data before the adaptation procedure or trained along with the M_t and D during adaptation in an alternating manner. In practice, the former approach is more susceptible to poor local minima e.g., mode collapse, while the second approach is very unstable during training and difficult to converge. For a contrast of the two approaches see e..g, [278] and [3] respectively. Experimentally, we found the former approach to be more practical, with better results in our task.

5.2.2 Transfer Learning for HAR

The notion of transfer learning [199], i.e., of storing knowledge which has been gained upon the solution of a problem and then of its use in a different, yet related one, has great potential in aiding the action recognition problem.

Xu et al. [313] proposed the use of autoencoders on high-level representations obtained through semantic transfer. In particular, video data was represented by action bank features [228]. Action banks for source and target datasets were then brought to lie in the same space by applying PCA to a fixed number of components. A latent representation was computed by training the autoencoders to reconstruct the class centroid over both datasets for each instance in the source and target domain. Unlike our method, this training procedure requires labels for the target domain in order to be able to train the target autoencoder. Moreover, Yusuf and Koniusz [269] proposed a CNN architecture on ker-

nel based feature maps extracted from skeletal data and leverage a supervised domain adaptation technique to increase the robustness of the learnt classifier. Note that again, target labels are required and this method cannot be used for labeling an unlabeled or sparsely labeled dataset. The particular supervised adaptation technique utilized is the So-HoT algorithm [131] which attempts to align second order statistics across datasets.

Another interesting work attempts to transfer knowledge for the video action recognition task from still images, which are typically more available. Zhang et al. [324] manipulated features extracted through kernel PCA to effectively transfer knowledge between heterogeneous domains. The video data was analyzed into key frames using a shot boundary detection algorithm. The method benefits from both labeled and unlabeled video data and thus lies within the frame of semi-supervised learning as does our method. Finally, similarly to our work, Hachiya et al. [85] tackled homogeneous adaptation for action recognition in the semi-supervised setting. However, rather than video, they used accelerometer data and extracted a representation based on orientation invariant statistics. Moreover, while we use a feature learning approach for adapting classifiers (adversarial discriminative adaptation), in the aforementioned work, importance sampling based adaptation was utilized.

5.3 Methodology

In this Chapter our goal is to provide a domain adaptation approach [41] tailored to the problem of HAR, in order to decrease the test error of a classifier when it is trained on data acquired using a different viewpoint (i.e., sampled from a different distribution) than the test set, i.e., in the cross-view evaluation setup (see subsection 2.4.2. Specifically, we propose an adversarial-based approach, using neural networks [3, 278, 296]. We herein remind that the main idea of these algorithms is similar to the one of Generative Adversarial Networks [77]. Particularly, the goal is to learn two embeddings M_s, M_t for source and target data, i.e., of the aforementioned different viewpoints, respectively, into some latent space $\mathcal{L}$, such that the distributions of target and source data in $\mathcal{L}$ are the same. We further require that the representation of data in $\mathcal{L}$ is rich enough to support classification of source instances. Note that typically no labelled data for the target domain is available during training, while the source domain data is annotated.

To align distributions of target and source data, we use a domain discriminator network D which is trained to discriminate the domains in the latent space for fixed M_s and M_t. D is kept fixed, while its gradients are used to update the parameters of M_t. This process is repeated until both networks reach their optimal performance, i.e., an equilibrium is reached. At that point, it can be shown that the distributions of source and target latent representations are aligned [77, 278]. The proposed approach is inspired from Adversarial Discriminative Domain Adaptation (ADDA) [278], which has been described in subsection 2.2.3. In particular, ADDA is adapted to a semi-supervised setting, i.e., we assume that a small part of the target domain is labelled while the remaining labels are unknown. This assumption holds with a lot of problems that need to be resolved in practice. Annotating a

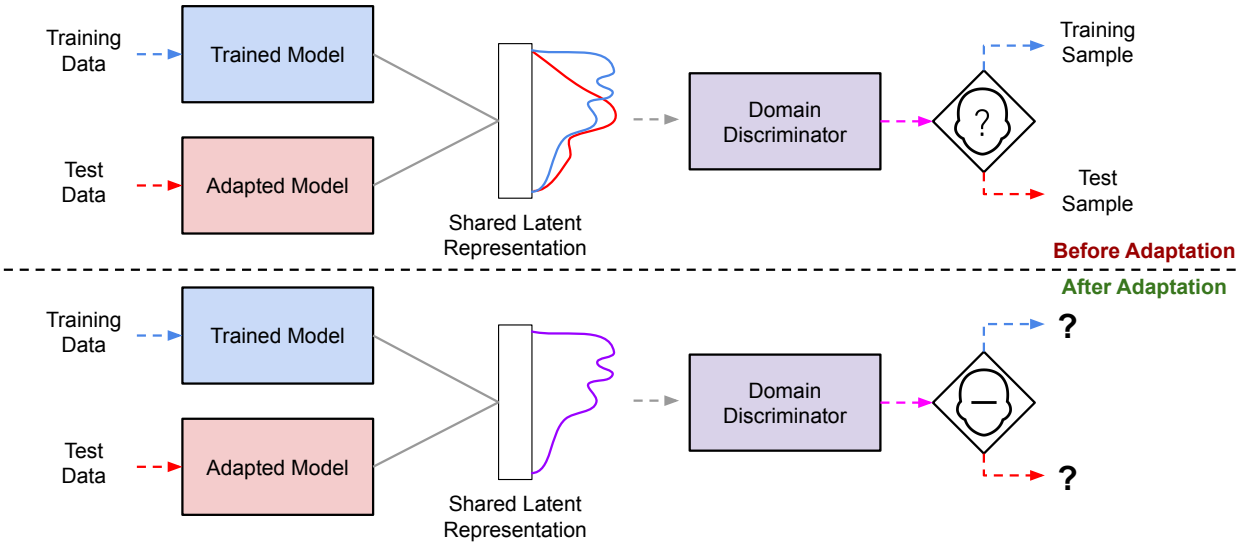


Figure 5.1: Visual illustration of the effects of domain adaptation. On top the adapted model is equal to the trained model. Because source and target data have different distributions, the resulting distributions in latent space are different and the domain discriminator can be trained to classify instances in latent space. After the adaptation procedure is finished, the adapted model has been trained so that the distribution of its output, when its input is distributed as the test data, is equal to the source data distribution in latent space. The domain discriminator can no longer discriminate the domains.

large dataset is the main bottleneck in constructing training sets, however labeling a small portion of it is often considered feasible. Therefore, the proposed approach aims to boost the performance of a source data classifier given a large corpus of sparsely labeled data from the target domain. In our case we consider that all activity samples from a given viewpoint (source) and also a small portion of samples from another viewpoint (target) are annotated and build on top of a source classifier C obtained through any supervised learning approach on the source data, which remains fixed with pre-computed parameters (e.g., obtained through standard supervised learning on the source domain), while the target representer M_t is trained. The role of the labeled data of the target domain is to provide supplementary information to the model during training.

In contrast to ADDA, the loss function herein used to train the target representer network $\mathcal{L}_{M_t}$ is given by

$$\mathcal{L}_{M_t} = -E_{x_t \sim X_T}[\log(D(\mathbf{M}_t(x_t)))] - \lambda E_{x_t, y_t \sim X_T}[\log(C(\mathbf{M}_t(x_t)))] \,, \tag{5.1}$$

where the second expectation is replaced with the empirical average over the labeled subset of the target domain. Note that λ in Eq. 5.1 is a trade-off parameter, whose value is empirically selected; for $\lambda = 0$ we get the typical ADDA loss, while for $\lambda \to \infty$ we obtain a typical supervised optimization task. Since a very small portion of the target data set is labeled, typical supervised learning would not be a robust approach to tackle the classification problem.

Therefore, the training procedure as adopted in the experimental section of this Chapter takes the following form: Firstly, the source network C is trained in a standard supervised learning manner using the source training data. The weights of M_s are then fixed. In turn, we initialize the target representation network M_t with the weights of M_s. This is a standard step in adversarial domain adaptation literature which helps avoid poor local minima [278]. The domain discriminator is initialized randomly and must be chosen with more parameters than the classifier network C [20], given the principle that for effective domain adaptation and a particular representation, it should be harder to discriminate domains than discriminating classes. At each step during training, a batch of source and target instances is sampled and the corresponding representations are computed using M_s and M_t. The domain discriminator D is trained in a standard supervised manner to discriminate between instances from source and target domain. Once D is updated, its weights are fixed, and the target representation network M_t is trained in a similar manner but using the reversed gradients of D.

In Fig. 5.2 we illustrate an overview of the proposed approach, as applied to the task of HAR, using a cross-view setup. In brief, given two camera viewpoints (i.e., L and R), raw visual data are captured, while e.g., an actor performs a given action. From the RBG and depth data captured, skeletal sequences are extracted. Upon concatenation and interpolation of those sequences, as described in Chapter 3, a signal image is formed. Using the Discrete Sine Transformation, an activity image is then formed. The collection of captured action instances form two datasets, namely D_s and D_t. The former corresponds to the source domain, while the latter to the target domain and is further split to a labelled (D_t^l) and an unlabelled (D_t^u) subset. D_s is used for training the source model S, consisting of the source representer M_s and the classifier C. M_s is the input of the viewpoint adaptation procedure, which is a process combining domain confusion and supervised training. The former utilizes D_s and D_t^u while the latter uses D_t^l. The output of the viewpoint adaptation is the target representer model M_T. Now, when the target camera captures a raw data sample, following the aforementioned process its activity image is adapted using M_T and the action is recognized by C, fed with the adapted sample. Note that for classification we used the same CNN architecture as in Chapters 3 and 4.

Note that this process can be shown to minimize the Jensen-Shannon divergence [39], i.e., maximizing the similarity between the distribution of source and target instances in latent space, i.e. the distributions of the outputs of M_s and M_t, if at each step the discriminator D is trained to optimality. For this reason, it is a good practice to perform multiple updates on the parameters of D for each update of parameters of M_t. However care must be taken since if D can perfectly discriminate between the two domains, there is no gradient feedback to continue training [10]. For this reason one needs to carefully adjust the number of iterations for D before each update of M_t. After convergence, the source classifier network C can be used to classify target images mapped into the latent space by M_t. This is because the covariate shift problem has been mitigated in the latent space, and by assumption $T_s = T_t$ and hence $P(Y|X)$, which C approximates, is the same for both domains.

Both source and target classifiers follow the exact architecture that has been used in

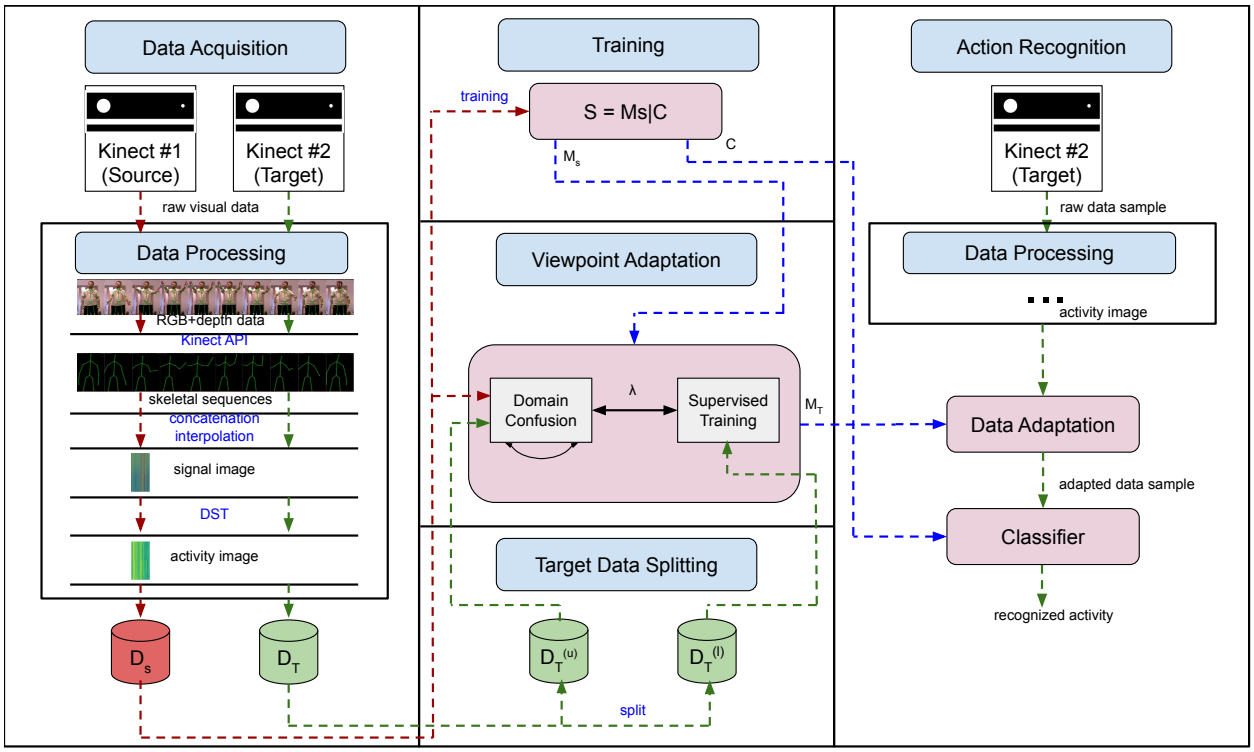


Figure 5.2: A visual overview of the proposed domain adaptation approach.

Chapters 3 and 4. While the source classifier has been trained using the dataset and the approach of 3, using Discrete Sine Transform to create the activity images used as input, since it demonstrated best performance, as discussed in Chapter 4. For training both networks, the ReLU activation function was used, the batch size was set to 100, the Adam optimizer was used, the dropout was set to 0.55 and the learning rate was set to 0.001, using the loss of the validation set calculated via cross-entropy as an early stopping method. Moreover, the discriminator used is a simple fully-connected NN, with 2 hidden layers, which uses as input the extracted feature vector of an activity image, i.e., the output of the last pooling layer of the trained classifier and its output denotes whether the input belongs to the source or the target domain.

Concluding, the proposed approach may be utilized without access to any target data labels. However, having such labeled target data can help improve the performance of the target classifier as it may help guide the adaptation procedure and aid in avoiding poor local minima. In particular, a method for incorporating target label knowledge into standard adversarial domain adaptation schemes is explored in [217].

5.4 Experimental Protocol and Results

In this Section, firstly we present the evaluation protocol that we have followed for the evaluation of the proposed domain adaptation approach, which we also compare with the

data augmentation approach presented in Chapter 4. We then present and discuss the corresponding results and we compare with baseline approaches.

More specifically, as in previous Chapters, we aim to assess whether and how the proposed strategies may be used to assist human activity recognition. To this, we once again experimented with the 11 classes of PKU-MMD that correspond to ADLs, as in Chapters 3 and 4. Note that since evaluation using all 51 classes would require hardware which was not currently available at the moment of experimentation, we limited our evaluation only to the aforementioned case. We remind that the selected 11 classes are: *eat meal snack*, *falling*, *handshaking*, *hugging other person*, *make a phone call answer phone*, *playing with phone tablet*, *reading*, *sitting down*, *standing up*, *typing on a keyboard* and *wear jacket*.

In this Chapter and contrary to Chapters 3 and 4, we limit our evaluation to cases where different camera viewpoints are used for training and testing, e.g., samples from *L* camera view are used for training while samples from *R* camera view are used for testing. We herein remind that the goal of such experiments is to test the robustness of a given approach in terms of typical geometric transformations (i.e., a rotation), which could be regarded as a simulation of a real-life case of abrupt viewpoint changes, typically occurring when subjects are not limited to a relatively small area or are not required to face directly a given camera.

The evaluation protocol we have used for the case of domain adaptation is as follows: one of the camera views is selected as source data, while another as target data. At the following, we shall use the typical notation $X \rightarrow Y$, which denotes adaptation between source X and target Y. For the sake of the presentation, the same notation shall be used also in the augmentation case, denoting training set X and testing set Y. Therefore, we evaluate our approach for all possible combinations, i.e., $L \rightarrow R, L \rightarrow M, R \rightarrow M, R \rightarrow L, M \rightarrow L$ and $M \rightarrow R$. Also, considering the semi-supervised setting that we have presented in subsection 5.3, in this case it is required to use a small, labeled subset of the target data; these examples serve as labeled target instances and are utilized during training both for providing a supervision signal and for unsupervised adversarial training. Of course, this subset is excluded when calculating validation accuracy both in this case and also in the case of augmentation. In every combination, the percentage of the labeled target data varies between $0\%, 1\%, 5\%$ and 10%, while a source model (S) is trained in a standard supervised way on the source domain. Experimental results of domain adaptation are presented in Table 5.1.

The evaluation protocol we have used for the case of augmentation is the same as in Chapter 4. We remind that it is as follows: we apply rotation transformations on signal images, using $\theta \in \{\pm 45^\circ, \pm 90^\circ\}$ (See Eq. 4.1). We also remind that *L* signal images, when rotations of -45° and -90° are applied, align to *M*, *R*, respectively. Similarly *M* signal images when rotations of 45° and -45° are applied align to *L*, *R*, respectively, while *R* signal images when rotations of 45°, 90° are applied align to *M*, *L*, respectively. Obviously, the aforementioned process results to a multiplication of the number of available images by a factor of 5. We may now use this augmented data set instead of the original one for training, aiming to achieve a significant increase in the classification performance of our model. In other words, our goal could be alternatively stated as follows: under a given

Table 5.1: Experimental results. VA and DA denote view adaptation and domain adaptation approaches, respectively. Figures represent average accuracy percentages over ten runs. In each setup best accuracy achieved is indicated with bold. For each viewpoint adaptation scenario, e.g. "Left to Middle" (indicated $L \rightarrow M$) the four numbers (0, 1, 5 and 10) indicate the percentage of target data instances which are labelled. S is the source model, built by a simple cross-view experiment. T_{rand} and T_{w_S} denote models whose initial weights have been taken randomly and from S, respectively. Note that with no labeled data in the target domain (i.e. 0) these models cannot be trained and hence these columns are omitted.

	Baselines									**VA**	**DA**			
	S	T_{w_S}				T_{rand}					**0**	**1**	**5**	**10**
		0	**1**	**5**	**10**	**0**	**1**	**5**	**10**					
$L \rightarrow M$	0.85	-	0.85	0.86	0.92	-	0.61	0.73	0.77	**0.93**	0.84	0.86	0.89	0.92
$L \rightarrow R$	0.41	-	0.60	0.70	0.75	-	0.51	0.61	0.71	**0.85**	0.50	0.63	0.78	0.82
$M \rightarrow L$	0.83	-	0.83	0.87	0.90	-	0.51	0.68	0.79	0.85	0.83	0.84	0.87	**0.91**
$M \rightarrow R$	0.78	-	0.80	0.85	0.90	-	0.51	0.70	0.76	0.90	0.83	0.83	0.86	**0.91**
$R \rightarrow L$	0.44	-	0.60	0.76	0.81	-	0.51	0.65	0.77	0.69	0.53	0.65	0.80	**0.86**
$R \rightarrow M$	0.85	-	0.85	0.86	0.91	-	0.61	0.70	0.77	0.88	0.85	0.85	0.88	**0.92**
mean	0.69	-	0.76	0.82	0.87	-	0.54	0.68	0.76	0.85	0.73	0.78	0.85	**0.89**

camera viewpoint, provide more reliable recognition of human actions upon training with an augmented data set; the latter comprises of all aligned images that resulted from the aforementioned rotation transformations. Experimental results of data augmentation are presented in Table 5.1.

We should herein note that to the best of our knowledge, the herein proposed approach is the first that a) applies data augmentation in raw skeletal data; and b) applies an adversarial domain adaptation strategy. Therefore, in order to compare the aforementioned strategies, we used the following baselines: a) the standard (source) model S; b) a model with random initial weights, i.e., T_{rand}; and c) a model with initial weights taken from S, i.e., T_{w_S}. This last approach represents a widely utilized class of transfer learning techniques known as fine tuning methods [41]. Results of baseline approaches are also presented in Table 5.1.

Regarding the results presented in Table 5.1, we should emphasize the following: view adaptation was the best strategy when training with the L dataset. When tested with M, it achieved an accuracy of 0.93, which was slightly better, compared to domain adaptation using 10% labeled data samples which achieved an accuracy of 0.92. Moreover, when tested with R, it achieved an accuracy of 0.85, while domain adaptation using 10% labeled data samples achieved an accuracy of 0.82. In all other cases, domain adaptation with 10% labeled examples was the best option. The most notable example was the case of training with R and testing with L camera, wherein domain adaptation achieved an accuracy of 0.92, while view augmentation was limited to 0.69.

The results of this study have indicated that both of the herein applied techniques offer substantial improvement to the generalization capabilities of HAR classifiers in the cross-view

scenario. In particular, comparing these methods with T_{rand} model results, which corresponds to "vanilla" supervised learning without any cross domain knowledge transfer and when labelled data are scarce, we observe a significant performance boosts. Moreover, the source domain S often outperforms the T_{rand} model (when the viewpoint change is mild, e.g. $L \rightarrow M$) highlighting the potential benefit of cross-domain data when training viewpoint robust classifiers. The benefit of utilizing data augmentation and domain adaptation is made much more apparent in more extended viewpoint changes (e.g. $L \rightarrow R$ as opposed to $L \rightarrow M$), where simply using a source model trained on a viewpoint different than the test set gives poor performance.

We further observe that data augmentation typically performs much better than domain adaptation when no target labels are available. In particular, the accuracy of models trained with data augmentation techniques are much less dependent on the underlying cross-view adaptation than the corresponding domain adaptation results. Domain adaptation yields almost no improvement for easy viewpoint changes while it yields satisfying improvement for harder viewpoint changes, albeit much less significant than the ones achieved through augmentation. In the semi-supervised setting the domain adaptation approach substantially benefits from the presence of a supervision signal which guides the adaptation process. The performance of adapted models in this setting is much more consistent between different cross-view adaptation instances and on average outperforms the plain data augmentation approach.

Although in the semi-supervised setting domain adaptation provides the best accuracy out of the examined methods, the associated hyper-parameter tuning may become an important issue when deploying such methods for real-life applications, generating a substantial computational overhead. In particular, we observed significant sensitivity to learning rates, the trade-off parameter between the domain confusion and the supervision signal term in the adversarial objective function and the number of updates on the domain discriminator weights per update of the target representation network updates. On the other hand, producing synthetic data to capture the effect of changing the viewpoint is relatively cheap and may be preferred for many practical HAR tasks. Moreover, the percentage of labeled data played a crucial role in domain adaptation results. In cases where target domain labelled data is less abundant, an augmentation approach for cross view activity recognition is observed to be a more appropriate choice altogether.

Finally, in case of cross-subject, we found that source and target errors were similar without any adaptation indicating that no covariate shift is present in the this setting. This indicates that covariate shift due to different subjects has been effectively mitigated through the use of skeletal data and interpolation.

5.5 Conclusions

In this Chapter we addressed the problem of lack of a sufficient amount of labeled data that are necessary when training a model that would be used for human activity recognition. In particular, in many real-life applications, the data collection and annotation process may

be a very costly and/or slow process. To overcome this limitation, we proposed the use of domain adaptation, aiming to mitigate the covariate shift problem given that training and evaluation sets derive from the same distribution. Specifically, we experimented with an adversarial domain adaptation approach, under the assumption that part of the testing data were labeled and excluded from evaluation and performed comparisons with the previously presented data augmentation methodology. Our approach was based on the HAR methodology presented in Chapter 3. We used a 2D image representation of these data, which relies on spectral images that have been obtained through the application of DST on raw data. Note that domain adaptation (and also data augmentation) is agnostic to the way this representation is formed. Since the experimental setup comprised three cameras, i.e., three different viewpoints, our goal was to transfer knowledge from one viewpoint to another. In other words, we have trained a model using samples captured under a single viewpoint (source model). Evaluation used only samples from a different viewpoint. Since the angle is significantly different, a drop of performance is expected, making them impractical for real-life applications. However, the application of both transfer approaches enabled a significant increase of performance in this cross-view scenario. We evaluated the proposed approach using a popular action recognition dataset as a source, focusing on a subset of 11 actions which we believe are the most close to real-life ADLs.

Our experiments showed that when target data are fully unlabelled, data augmentation offers a significant increase over the source model. However, a small amount of labeled data makes domain adaptation approach to exhibit clearly improved performance overall, making it practical for real-life applications. For example, consider an assistive living scenario, where a single camera is used to monitor the behaviour of a human subject. Typically, in such scenarios the space where subjects act are not strictly limited. Therefore, the viewpoint may significantly change. Now, consider an activity recognition model that has been pre-trained in laboratory conditions. In case that the subject is able to participate in a data collection process, it should be more effective to adopt a domain adaptation approach. Of course, if this is not possible, then data augmentation is able to offer an adequate alternative, while it is able to be immediately applied.

We should herein note that the domain adaptation step introduces a significant overhead in any classification methodology, since it is a time consuming task. However, it consists an offline step, therefore, it may still be used in real-life applications, since it does not have any effect in a given deep neural network architecture in terms of complexity. Moreover, it requires careful hyper-parameter tuning. In particular, the design choices including standard network topology, batch size, learning rates etc., require careful tuning because training occurs as an interplay between multiple networks whose training capacity affects the training of the other networks. In addition, one needs to tune the trade-off parameter between the domain confusion term and the supervision signal term in the objective function. Large values of the parameter may lead to overfitting while lower values may hinder training. Moreover, as mentioned above, it is important to tune the number of training iterations of the domain discriminator network per training iteration of the target representation network. Typically, grid search should be applied but we note that tuning these parameters may introduce a substantial computational overhead.

6. DATA FUSION APPROACHES FOR HAR

6.1 Introduction

In this Chapter we present a novel approach for HAR that fuses multiple modalities, incorporating RGB, depth and visual representations of 3D skeletal information. We build on the works that have been presented in previous Chapters 3 and 4 and propose a late fusion methodology that uses as input RGB and depth video sequences. The former are "summarized" into a single "dynamic" image [17]. Both data representations are used to create 3D skeletal sequences of the subjects. These are then used to create (a) 4 visual representations of skeletal data upon applying well-known image transforms to the spectral space as in Chapter 3; and (b) a pseudo-colored image representation of skeletal data [285]. For each of the aforementioned modalities, a convolutional neural network is trained and is used as feature extractor. Additionally, we also investigate whether a set of handcrafted features, initially proposed for the problem of arm gesture recognition [206], which are herein applied to the whole skeleton.

We should note that earlier HAR approaches were based on the extraction of handcrafted features from raw visual data. These features are algorithmically extracted, capture visual properties of postures and/or motion and are used to train traditional machine learning approaches, such as neural networks or support vector machines [234]. However they have been criticized since they exhibit significant drop of performance and lack of generalization when applied to large-scale datasets. Moreover they are not robust to viewpoint changes. Of course, as with every other field of application, they require specific knowledge of the domain of application. Therefore, in many cases handcrafted features are considered to be obsolete and are rarely used in research works, being almost completely replaced by deep approaches.

Specifically, at the following we consider the cases of fusion of dynamic and pseudo-colored images with (a) handcrafted features; and (b) activity images. In both cases, the viewpoint augmentation scheme presented in Chapter 4 is applied prior to the creation of both image transforms and pseudo-colored image representations of skeletal motion. Moreover, extracted features are fused, scaled and used as input to an SVM classifier. The proposed method is evaluated on three publicly available human motion datasets, namely PKU-MMD dataset [163], SYSU 3D Human-Object Interaction (HOI) dataset [102] and UTKinect-Action3D dataset [311]. The novelties of the herein proposed approaches are as follows: (a) we extract deep features and propose an early fusion of different image transforms that correspond to skeletal motion; (b) we apply the viewpoint augmentation scheme (see Chapter 4) prior to create image transforms and pseudo-colored image representations of skeletal motion; (c) we incorporate dynamic images to exploit the raw RGB information of motion; (d) we propose fusion of skeletal motion and raw RGB features using only CNNs; and (e) we investigate whether handcrafted features are able to boost the performance of deep approaches.

The rest of this Chapter is organized as follows: subsection 6.2 presents recent research

works closely related to the proposed approach. Specifically, focus is given on handcrafted features used for skeletal motion and multimodal fusion methods. Then, in subsection 6.3 we first describe the data modalities that are used within this work and then the whole classification methodology. The experimental setup, evaluation and results are presented in subsection 6.4. Finally conclusions are drawn in subsection 6.5.

6.2 Related Work

In this Section we present a brief overview of recent research works on HAR that (a) are based on the extraction of handcrafted features; (b) utilize information from multiple modalities; and (c) fuse handcrafted and deep features.

6.2.1 Hand-crafted Features

Handcrafted features are extracted from raw data using a predefined algorithm. Typical approaches in the field of HAR exploit angles between joints and joint distances or properties of joint trajectories. A notable work has been proposed by Xia et al. [311] and is based on 3D joint location histogram within a modified spherical coordinate system. Moreover, the authors built posture vocabularies upon clustering of the histogram vectors. Similarly, Keceli and Can [124] proposed histograms of angles and displacements between a set of predefined joints in the 3D space. Gowayyed et al. [79] used a histogram of oriented displacements, extracted under 3 viewpoints, so as to create 3D features. Yang and Tian [316] combined activity information including static pose, properties of motion and overall dynamics and combined these three representations so as to create a compact description of frames. Finally, Pazhoumand-Dar et al. [209] first created relative skeleton motion trajectories and then selected a subset of these features using the longest common sub-sequence algorithm.

6.2.2 Multimodal Fusion Methods

Several research works dealing with the fusion of more than one data modalities have been proposed during the last few years. A notable early work is the one of Simonyan et al., [247] who trained a CNN for capturing spatial features from raw video frames and another for capturing motion features from dense optical flow. A late fusion approach was adopted, using an SVM. Another early work is the one of Chaaraoui et al., [33] who combined body pose estimation and 2D shapes, to obtain skeletal and silhouette-based features, which were then combined by early fusion.

The majority of more recent works is based on the use of both RGB and depth data, [33, 55, 132, 286]. Other modalities that have been used as input in multimodal fusion approaches are thermal data [87], inertial measurements [55, 108], audio data [159] and RFID data [159]. In those works, each data modality is independently processed, in order

to extract features. In many cases, features are handcrafted [55], yet in the majority of works deep features are extracted, aiming to capture spatial [247] and/or temporal [247] properties of human motion. To this goal, most approaches rely only in CNN architectures [33, 132, 167, 247, 286] that are trained as feature extractors and aim to extract features from either from raw video frames or from intermediate 2D feature representations. Although deep features have almost completely dominated the research area of HAR, it has been shown that handcrafted features may capture complementary motion properties, thus, their fusion may boost performance [132]. In several cases, especially when dealing with small datasets, pre-trained networks in the same or in a similar domain or dataset [286] or transfer learning approaches [167] are used. More recent works may additionally use LSTM networks so as to extract temporal features [87, 108, 159, 336], while in few cases only LSTM networks are used [264].

Moreover, several approaches dealing with the fusion of several representations have been proposed. Basly et al. [14] combined deep learning methods with traditional classifier hand-crafted features extractors. For feature extraction, they used a pre-trained CNN approach-based residual neural network (ResNet) model. The resulting feature vector was then fed as an input to an SVM classifier. Simonyan et al. [247] trained two spatial and temporal CNNs. The softmax scores for each model were combined with a late fusion approach, i.e., by training a multi-class linear SVM. Ehatisham-Ul-Haq et al. [55] proposed a multimodal feature-level fusion approach for robust human action recognition. Their features include densely extracted histogram of oriented gradient (HOG) features from RGB/depth videos and statistical signal attributes from wearable sensors data. K-nearest neighbor and support vector machine classifiers were used for training and testing the proposed fusion model for HAR. Chaaraoui et al. [33] combined body poses estimation and 2D shape, in order to improve human action recognition. Using efficient feature extraction techniques, skeletal and silhouette-based features low-dimensional, real-time features were obtained. These two features were then combined by means of feature fusion. Finally, Vernikos et al. [286] presented an approach for the recognition of human activity that combined handcrafted features from 3D skeletal data and contextual features learned by a trained deep CNN. They trained a CNN using a dataset for action recognition and used the output of the last fully-connected layer as a contextual feature extractor. Then, an SVM was trained upon an early fusion step of both features.

Typically, fusion is implemented as concatenation of these feature vectors (i.e., early fusion) and recognition involves a traditional machine learning classifier, such as a support vector machine, [55, 132, 247, 286], or a clustering approach such as k-means [33, 55]. Yet, late fusion, i.e., fusion of classifier decisions is also used [204, 247]. However, the majority of works rely on a single representation per data modality. Specifically, it has not been investigated whether different representations of skeletal joint motion may carry complementary information, thus their fusion could provide a performance boost on recognition approaches. Moreover, raw RGB data are often ignored or used only for skeleton extraction, yet, in this work we demonstrate that they may significantly assist recognition in several cases.

6.2.3 Fusion of handcrafted and deep features

A few approaches for fusion of handcrafted features with features learnt by a deep architecture have been recently proposed in the field of HAR. Typically, these approaches fuse features from the last pooling layer of a CNN with the handcrafted ones by concatenating them into a single vector and then use a traditional machine learning classifier, e.g., a support vector machine for classification. Khan et al. [127] proposed fusion of learnt features of a pre-trained CNN model, namely the VGG19 architecture [246] and multiview handcrafted features that are based on horizontal/vertical gradients and directional features. In another work, Khan et al. [128] extracted motion, geometric and shape features from regions of interest and fused them with features of pre-trained AlexNet architecture [134]. Udeen and Lee [279] extracted deep spatial features of Inception-Resnet-v2 architecture [266] and fused them with a novel feature descriptor that captures spatio-temporal and shape features. Finally, Vernikos et al. [286] trained a CNN using a dataset for action recognition and used it as feature extractor for a gesture recognition task, upon fusing its learnt features with handcrafted inter-joint distance and joint angle features.

6.3 Methodology

6.3.1 Data Modalities

6.3.1.1 Pseudo-colored Image Representation of Skeletal Data

As it has already been discussed in Chapter 3, another approach to use the 3D skeletal information as input to a CNN, is to create image representations of skeletal data. However, when creating such representations, one should consider capturing and preserving spatio-temporal properties of skeleton trajectories, so that the resulting representations could be able to discriminate among different activities. Moreover, they should also comply to the graph structure skeleton representation. To this, we use the representation of Vernikos et al. [285] which has been successfully applied to the problem of HAR.

Specifically, this representation aims to capture inter-joint distances as they vary through the duration of the activity. These are then used to create pseudo-colors within an artificial RGB image. Note that it is based on the 3D trajectories of skeletal joints. Let $x(n)$, $y(n)$ and $z(n)$ denote the sequences of coordinates of each of the N available joints, at the n-th frame F_n of the video sequence depicting the activity. Through the duration of any activity, a set of $3 \times N$ signals is collected for a given video sequence. To address the problem of temporal variability between actions and between users, as it has been described in subsection 3.3.1, a linear interpolation step in the exact same way as in the case of signal images is imposed. Then, from each of the aforementioned sequences difference between consecutive frames is calculated. To create pseudo-colored images, x, y, z coordinates are assigned to R, G, B color channels of the pseudo-colored image, respectively.

Particularly, the process of creating a pseudo-colored image $\mathbf{I}$ is as follows: Let $x_i(n), i =$



Figure 6.1: Sequences of actors performing the activities *handwaving* (top) and *kicking something* (bottom). Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the PKU-MMD dataset [163] and have been trimmed for illustration purposes.

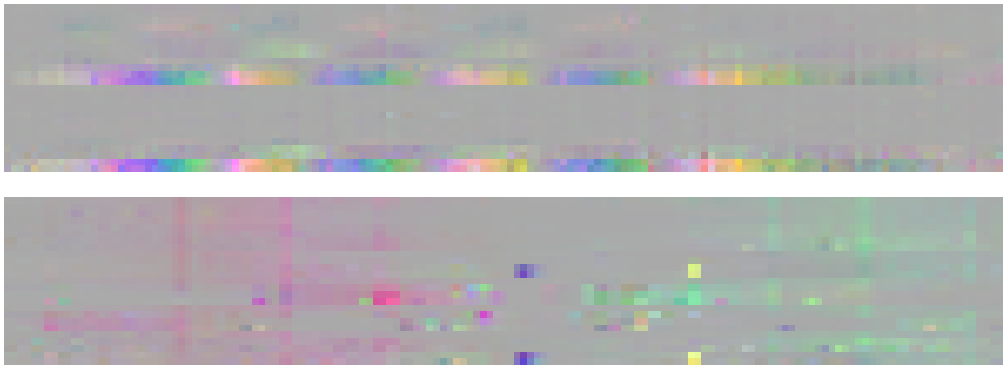


Figure 6.2: The pseudo-colored images for the activities *handwaving* (top) and *kicking something* (bottom) that are illustrated in Fig. 6.1.

$1, \ldots, N$ denote the x-position of the i-th joint in the n-th frame. Let r, g, b denote the red, green and blue channel of $\mathbf{I}$, respectively. Pixel values of $r(i,n)$, $g(i,n)$, $b(i,n)$ are calculated as:

$$\begin{aligned} r(i,n) &= x_i(n+1) - x_i(n) \\ g(i,n) &= y_i(n+1) - y_i(n) \\ b(i,n) &= z_i(n+1) - z_i(n) \end{aligned} \tag{6.1}$$

where $n = 1, \ldots, T_a$ and $i = 1, \ldots, N$. As it is exhibited, the way these pseudo-colored images are formed, leads to preserving both the temporal and the spatial properties of the skeleton trajectories. Obviously, the dimension of the pseudo-colored image is $N \times T_a \times 3$, which in our case is $25 \times 159 \times 3$. In Fig. 6.1 we illustrate two sequences of activities performed by actors, while in Fig. 6.2 we illustrate the corresponding pseudo-colored images.

6.3.2 Dynamic Images for Activity Representation

A dynamic image is a typical RGB image, which by construction aims to summarize the appearance and dynamics of a given video sequence. [17]. Specifically, the idea that lies behind dynamic images [62] is known as "rank pooling" and aims to represent a video sequence as a ranking function $S(\bullet)$ of its frames $F_1, \ldots, F_{T_a}$. By using $S(\bullet)$ a feature vector $\psi(F_i)$ from each frame F_i is extracted. Let

$$V_n = \frac{1}{n} \sum_{i=1}^{t} \psi(F_n) \tag{6.2}$$

denote the time average of the aforementioned features ranging from F_1 to F_n. The ranking function associates each moment n with a score $S(n)$, given a parameter set $\mathbf{d}$ and assigning larger scores to later frames. Scores are learned by solving a convex optimization problem which defines a function $\rho(\bullet)$ mapping a given video sequence comprising T frames to a vector $\mathbf{d}^*$, used as feature descriptor. Using the RankSVM formulation [251], $\mathbf{d}^*$ may be estimated as follows:

$$\mathbf{d}^* = \rho(F_1, \ldots, F_n; \psi) = \arg\min_{\mathbf{d}} E(\mathbf{d}) \ , \tag{6.3}$$

where

$$E(\mathbf{d}) = \frac{\lambda}{2}\|\mathbf{d}\|^2 + \frac{2}{T(T-1)} \times \sum_{q>t} \max\{0, 1 - S(q|\mathbf{d}) + S(t|\mathbf{d})\} \ . \tag{6.4}$$

Although $\psi(\bullet)$ may be any feature extractor, Bilen et al. [17] opted for simply using raw RGB pixel values and reported remarkable results. However, the most important aspects of such an approach are a) $\mathbf{d}^*$ may be interpreted as an RGB image as it has the exact same number of elements; and b) this image is obtained by rank pooling, thus it may be regarded as a summary of the whole video sequence. Note that the pixels in the produced dynamic images tend to focus on salient information rather than the background; this is considered the reason for them being appropriate for the problem of HAR, wherein typically the background is static and a subject consists a non-static part within the video sequence. This is clearly depicted in Fig. 6.3 where dynamic images that correspond to the activities of Fig. 6.1 are illustrated. In this image it is clear that information regarding the background has been "discarded," while emphasis has been given to the subject and her moving arm.

6.3.3 Handcrafted Features

In this work we use a set of handcrafted features that have been proposed by Paraskevopoulos et al. [206]. Although they have been initially designed for the problem of hand gesture recognition, upon initial experimental evaluation we ended up that they are appropriate for the problem of HAR. These features assume a set of joints organized in a hierarchical structure, wherein a parent–child relationship is implied. These joints move in a 3D



Figure 6.3: The dynamic images for the activities *handwaving* (left) and *kicking something* that are illustrated in Fig. 6.1.

space, over time. Therefore, they may be extracted by the aforementioned set of skeletal sequences. At the following we briefly present these features.

Specifically, for a given joint J, let J_c and J_p be its child and parent joints, respectively. Let $\mathbf{F} = \{F_i\},\ i = 1, 2, \ldots N$ denote the set of frames of a given video sequence depicting an activity. Moreover, let $\mathbf{v}_i^J$ be a vector that corresponds to the 3D coordinates of J within F_i. More specifically, $\mathbf{v}_i^J = \left(v_{x,i}^J, v_{y,i}^J, v_{z,i}^J\right)$ are the aforementioned x, y and z coordinates. Also, let $\mathcal{V}^{\mathcal{J}}$ be the set of all vectors $\mathbf{v}_i^J$. By $B(\mathcal{V}^{\mathcal{J}})$ we denote the 3D bounding box of $\mathcal{V}^{\mathcal{J}}$, by $a_{B(\mathcal{V}^{\mathcal{J}})}$ and $b_{B(\mathcal{V}^{\mathcal{J}})}$ the lengths of its horizontal and vertical sides, respectively. We summarize the extracted features in Table 6.1.

6.3.4 Fusion and Classification

The methodologies that are presented in this Chapter are based on the fusion of raw RGB and 3D skeletal motion data sequences. The former are used to create dynamic images, as described in subsection 6.3.2, which may be regarded as condensed representations of activities, i.e., each represented by a single 2D image. The latter are used to create four representations of the spectral content of the skeletal motion, which are based on well-known image transforms and have been presented in Chapter 3 and also the pseudo-colored representation of skeletal motion, presented in subsection 6.3.1.1. Moreover, we also use the skeleton sequences to extract hand-crafted features, computed using joints' co-ordinates and by using the methodology described in section 6.3.3. As we have already mentioned, from 2D image representations we extract deep features using CNNs, therefore these features are learned. Specifically, CNNs are trained and then each image

Table 6.1: Proposed features, extracted from the skeletal joints. For features marked with *, respective angles a_{pc}, b_{pc}, c_{pc} are calculated as: $a_{pc}^2 = \left(v_x^J - v_x^{J_c}\right)^2 + \left(v_y^J - v_y^{J_c}\right)^2$, $b_{pc} = v_x^J$, $c_{pc}^2 = \left(v_x^{J_p}\right)^2 + \left(v_y^J - v_y^{J_p}\right)^2$. Note that for the formulation of the triangle, a reference point with coordinates $(v_{x,i}^{J_p}, 0, 0)$ is used. By d we denote the Euclidean distance and F^J is the number of frames for each gesture, which is also used as a feature.

Feature name	Frames involved	Equation
Spatial angle	F_2, F_1	$\arccos \frac{\mathbf{v}_2^J \cdot \mathbf{v}_1^J}{\|\mathbf{v}_2^J\| \cdot \|\mathbf{v}_1^J\|}$
Spatial angle	F_N, F_{N-1}	$\arccos \frac{\mathbf{v}_N^J \cdot \mathbf{v}_{N-1}^J}{\|\mathbf{v}_N^J\| \cdot \|\mathbf{v}_{N-1}^J\|}$
Spatial angle	F_N, F_1	$\arccos \frac{\mathbf{v}_N^J \cdot \mathbf{v}_1^J}{\|\mathbf{v}_N^J\| \cdot \|\mathbf{v}_1^J\|}$
Total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{N} \arccos \left(\frac{\mathbf{v}_i^J \cdot \mathbf{v}_{i-1}^J}{\|\mathbf{v}_i^J\| \|\mathbf{v}_{i-1}^J\|} \right)$
Squared total vector angle	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \arccos \left(\frac{\mathbf{v}_i^J \cdot \mathbf{v}_{i-1}^J}{\|\mathbf{v}_i^J\| \|\mathbf{v}_{i-1}^J\|} \right)^2$
Total vector displacement	F_N, F_1	$\left\|\mathbf{v}_N^J - \mathbf{v}_1^J\right\|$
Total displacement	$F_1, \ldots, F_N$	$\sum_{i=1}^{n} \left\|\mathbf{v}_i^J - \mathbf{v}_{i-1}^J\right\|$
Maximum displacement	$F_1, \ldots, F_N$	$\max_{i=2,\ldots,N} \left(\left\|\mathbf{v}_i^J - \mathbf{v}_{i-1}^J\right\|\right)$
Bounding box diagonal length	$F_1, \ldots, F_N$	$\sqrt{a_{B(\mathcal{V}^J)}^2 + b_{B(\mathcal{V}^J)}^2}$
Bounding box angle	$F_1, \ldots, F_N$	$\arctan \frac{b_{B(\mathcal{V}^J)}}{a_{B(\mathcal{V}^J)}}$
Initial angle	F_1	$\measuredangle \mathbf{v}_1^J \mathbf{O} \mathbf{v}_1^{J_p}$ or $\measuredangle \mathbf{v}_1^J \mathbf{O} \mathbf{v}_1^{J_c}$
Final angle	F_N	$\measuredangle \mathbf{v}_N^J \mathbf{O} \mathbf{v}_N^{J_p}$ or $\measuredangle \mathbf{v}_N^J \mathbf{O} \mathbf{v}_N^{J_c}$
Mean angle	$F_1, \ldots, F_N$	$\frac{1}{N} \sum_{i=1}^{N} \measuredangle \mathbf{v}_i^J \mathbf{O} \mathbf{v}_i^{J_p}$ or $\frac{1}{N} \sum_{i=1}^{N} \measuredangle \mathbf{v}_i^J \mathbf{O} \mathbf{v}_i^{J_c}$
Max angle	$F_1, \ldots, F_N$	$\max_{i=1}^{N} \measuredangle \mathbf{v}_i^J \mathbf{O} \mathbf{v}_i^{J_p}$ or $\max_{i=1}^{N} \measuredangle \mathbf{v}_i^J \mathbf{O} \mathbf{v}_i^{J_c}$

is then fed to an appropriately trained CNN. The second from the end dense layer of each network is used as a feature vector. All types of features in each experiment are concatenated into a single feature vector. For classification, they are normalized and then upon a PCA step they are given as input to an SVM.

Specifically, we consider the following cases of fusion:

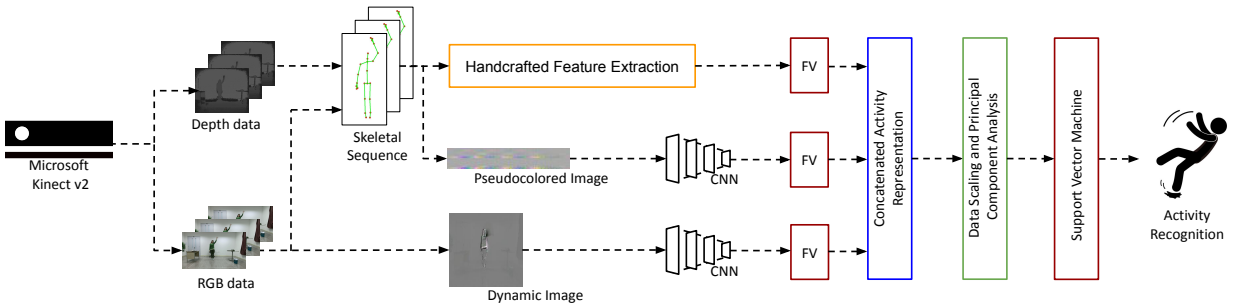


Figure 6.4: A visual overview of the approach of fusion between handcrafted features, pseudocolored and dynamic images.

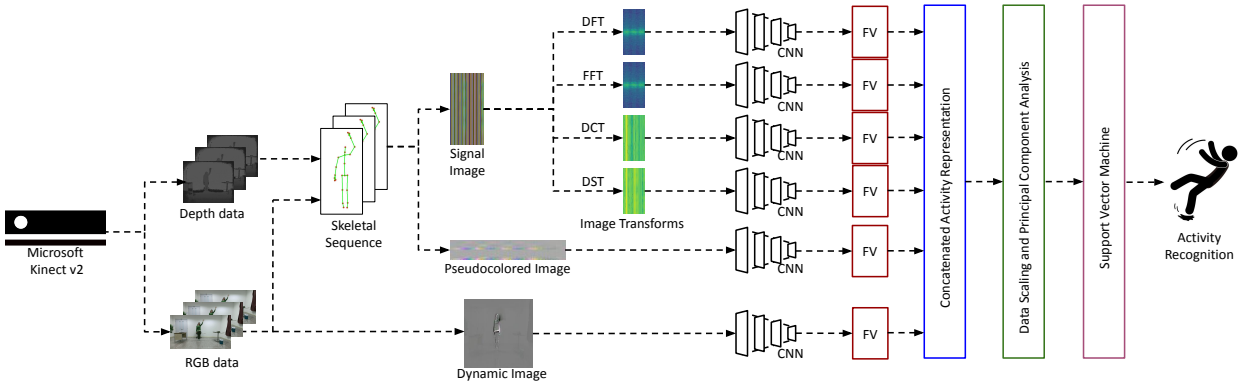


Figure 6.5: A visual overview of the proposed approach of fusion between activity, pseudocolored and dynamic images.

- fusion of dynamic images, pseudocolored images and handcrafted features;
- fusion of dynamic images, pseudocolored images and activity images.

Visual overviews of both strategies are illustrated in Figs. 6.4 and 6.5.

The architecture of the CNN that has been used throughout our experiments with DFT, FFT, DCT and DST images is the same that has been used throughout the experiments of Chapters 3 – 5 and is illustrated in Fig. 5.1. In case of the pseudo-colored image representations, the CNN architecture was based on the aforementioned one and has been experimentally modified to better fit this representation. It is illustrated in detail in Fig. 6.6a. It consists of a convolutional layer with 16 kernels of size 3×3, followed by a pooling layer. Then, 2 convolutional layers with 32 kernels of size 3×3 and a pooling layer follow, succeeded by a convolutional layer with 64 kernels of size 3×3 followed by a pooling layer. Each pooling layer uses “max-pooling” to perform 2×2 subsampling. Then, once again a flatten and two dense layers follow as in the aforementioned network.

Finally, in case of dynamic images, the CNN architecture that was adopted is based on the well-known VGG16 [246] and is illustrated in detail in Fig. 6.6b. First, 2 convolutional layers filter the input image with 64 kernels of size 3×3. A pooling layer follows. Then, 2

convolutional layers filter the input image with 128 of size 3×3, followed by another pooling layer. Then 3 convolutional layers filter the resulting image with 256 kernels of size 3×3 and are again followed by a pooling layer. Then, 2 triplets convolutional layer follow, filtering the resulting image with 512 kernels of size 3×3, each followed by a pooling layer. Then, a flatten layer transforms the output image of size 7×7 of the fifth pooling into a vector, with is then used as input to a dense layer of size 256 using dropout. Finally, a second dense layer produces the output of the network. All pooling layers use "max-pooling" to perform 2×2 subsampling.

Note that as previously mentioned, the second dense layer is omitted when the CNN is used for feature extraction. Also, the first and the second convolutional networks are trained using the available dataset, while the third convolutional network is pre-trained with Imagenet [45]. The convolutional layers are "freezed" and training of the dense layers using the dynamic images continues.

Therefore, upon using the aforementioned networks as feature extractors, a feature vector per representation is formed. The early fusion step of this work consists of concatenation of these 3 or 6 feature vectors in each case, accordingly. This way, combined representations of sizes 476 or 896 are created, respectively. Upon scaling and PCA, only the components that correspond to 95% of total variance are kept, resulting to reduced feature vectors which are then fed to the SVM. Note that due the way PCA works, the size depends on the dataset, therefore, as expected it varies, per experiment. Finally, for classification an SVM with an RBF kernel has been used.

Moreover, in the second series of experiments, i.e., when activity, pseudocolored and dynamic images are fused, the view augmentation methodology that has been presented in Chapter 4 has been incorporated to the training process.

6.4 Experimental Protocol and Results

In order to experimentally evaluate the proposed multimodal HAR approaches, we used PKU-MMD, as in previous Chapters, and also SYSU 3D Human-Object Interaction (HOI) [102] and UTKinect-Action3D [311] datasets, which have been previously presented in Subsection 2.4.1. We herein remind that both these datasets consist of 3D human motion. The former focuses on interactions between humans and objects, while the latter consists of 10 simple activities. Contrary to PKU-MMD which has been recorded using a multi-camera setup, comprising 3 Microsoft Kinect v2 cameras and capturing the subject simultaneously from 3 different angles, SYSU 3D HOI and UTKinect-Action 3D have been recorded using a single Microsoft Kinect v1 camera, directly facing the subject. In all datasets the following visual data modalities are provided: (a) raw RGB video sequences; (b) depth sequences; (c) infrared radiation sequences; and (d) positions in the 3D space of the extracted human skeleton joints, varying over time.

We have conducted experiments considering individually the following modalities: (a) pseudocolored images (PC); (b) dynamic images (DI); (c) handcrafted features (HF); (d)

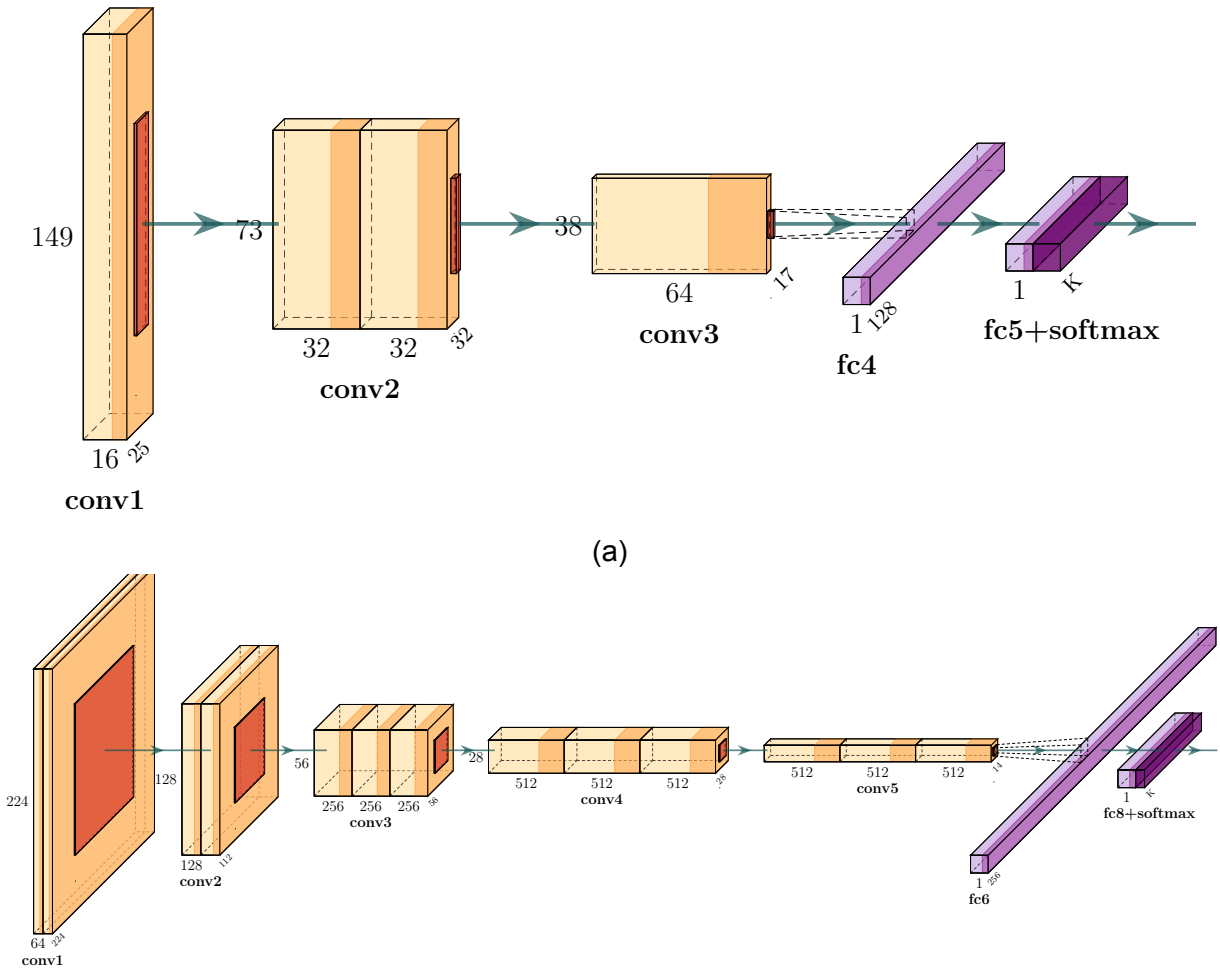


Figure 6.6: The deep convolutional network architectures that have been used in this work and for the cases of (a) pseudo-colored images; and (b) dynamic images.

discrete Fourier transform (DFT); (f) fast Fourier transform (FFT); (g) discrete cosine transform (DCT); and (h) discrete sine transform (DST). We also tested the following combinations using the late fusion strategy that has been presented in Subsection 6.3.4: (a) pseudocolored and dynamic images; (b) pseudocolored images and handcrafted features; (c) dynamic images and handcrafted features; (d) pseudocolored, dynamic images and handcrafted features; (e) all spectral transformations (TF), (f) all spectral transformations with pseudocolored images; and (g) all spectral transformations with dynamic and pseudocolored images.

In case of PKU-MMD, experiments are divided into three parts: a) *Single-view*; b) *Cross-view*; and c) *Cross-subject* experiments, as in Chapters 3 – 5. In case of Single-View and Cross-Subject experiments 87.5% of data have been used for training, while the remaining 12.5% for testing. In the former case, the dataset was randomly split , while in the latter case, the split that has been imposed by the authors of the dataset has been used. More-

over, in case of Cross-Subject experiments, in all cases all available data from a given viewpoint were used. Moreover, in case of SYSU 3D HOI, the authors define a typical protocol (setting 1), wherein 50% of samples are used for training and 50% for testing and a cross-subject protocol (setting 2), wherein 50% of subjects are used for training and 50% for testing, without any overlap. Finally, the evaluation protocol of UTKinect-Action3D dataset is much simpler, wherein a leave one sequence out cross validation is indicated by its authors.

6.4.1 Experimental Setup and Implementation Details

Experiments were performed on a personal workstation with an AMD Ryzen™5 1600 6-core processor on 3.20 GHz and 16GB RAM, using NVIDIA™Geforce GTX 1060 GPU with 6 GB GDDR5 VRAM and Ubuntu 20.04 (64 bit). The deep architecture has been implemented in Python, using Keras 2.4.3 [38] with the Tensorflow 2.5 [1] backend. All data pre-processing and processing steps have been implemented in Python 3.9 using NumPy,[1] SciPy[2] and OpenCV.[3] For training the CNN, the ReLU activation function has been used. Moreover, the batch size has been to 8 and the Adam/SGD optimizers(Adam in skeleton models,SGD in Dynamic model) has been used. Also, the dropout was set to 0.5, the learning rate was set to 0.001 and the network was trained for 150 epochs, using the loss of the validation set calculated via cross-entropy as an early stopping method, in order to avert overfitting. The parameters of the SVM were $C = 100$ and $\gamma = 0.001$ and have been selected upon grid search.

The four CNNs that are used to extract features from the DFT, FFT, DCT and DCT images comprise $2,164,339$ trainable parameters; training in case of the most demanding PKU-MMD dataset requires on average approx. 35 min., while extraction of features requires approx. 0.002 sec. The CNN that is used to extract features from the pseudo-colored images comprises $170,611$ trainable parameters; training in case of the most demanding PKU-MMD dataset requires on average approx. 25 min., while extraction of features requires approx. 0.001 sec. The VGG16 network that is used to extract features from the dynamic images comprises $21,150,579$ trainable parameters; training in case of the most demanding PKU-MMD dataset requires on average approx. 50 min., while extraction of features requires approx. 0.004 sec. Finally, the SVM that is used for classification requires approx. 420 sec for training and approx. 0.008 sec for classification of a given fused feature vector. Note that the aforementioned times may vary in every repetition of the experiment due to the early stopping.

[1] `https://numpy.org/`
[2] `https://scipy.org/`
[3] `https://opencv.org/`

Table 6.2: Experimental results of the fusion approach presented in this Chapter. Numbers denote accuracy. Bold numbers indicate best result, per case. Also "PC," denote pseudocolored images (subsection 6.3.1.1), "DI" denote dynamic images (subsection 6.3.2), "HF" denote handcrafted features (subsection 6.3.3). Moreover, "DFT," "FFT," "DCT" and "DCT" denote the Discrete Fourier Transform, the Fast Fourier Transform, the Discrete Cosine Transform and the Discrete Sine Transform, respectively (Chapter 3), while "TF" denotes their fusion. "L," "R" and "M" denote Left, Right and Middle cameras, while "CV," "CS" and "SV" denote cross-view, cross-subject and single-view cases.

Exp.	Viewpoint		DFT	FFT	DCT	DST	PC	DI	HF	PC+DI	PC+HF	DI+HF	PC+DI+ HF	TF	TF+PC	TF+PC+ DI
	Train	Test														
	L,R	M	0.44	0.44	0.55	0.52	0.81	0.59	0.62	0.81	0.83	0.71	0.85	0.92	**0.98**	**0.98**
	L,M	R	0.33	0.34	0.33	0.34	0.72	0.50	0.54	0.71	0.75	0.61	0.75	0.86	**0.95**	0.94
	R,M	L	0.32	0.33	0.32	0.34	0.74	0.51	0.56	0.74	0.77	0.61	0.77	0.86	**0.95**	0.94
	M	L	0.33	0.34	0.33	0.36	0.63	0.47	0.54	0.64	0.70	0.57	0.71	0.85	**0.94**	0.92
CV	M	R	0.32	0.31	0.36	0.32	0.62	0.50	0.52	0.62	0.70	0.55	0.68	0.85	**0.93**	0.92
	R	L	0.20	0.19	0.13	0.14	0.55	0.28	0.48	0.50	0.64	0.42	0.57	0.78	**0.90**	0.87
	R	M	0.32	0.34	0.36	0.34	0.65	0.39	0.55	0.61	0.73	0.53	0.69	0.87	**0.95**	0.94
	L	R	0.22	0.22	0.13	0.14	0.54	0.30	0.47	0.52	0.61	0.45	0.59	0.74	**0.89**	0.86
	L	M	0.33	0.30	0.35	0.39	0.67	0.47	0.57	0.68	0.72	0.60	0.74	0.86	**0.95**	0.93
CS	L,R,M	L,R,M	0.50	0.52	0.64	0.63	0.74	0.80	0.58	0.81	0.76	0.74	0.82	0.85	0.92	**0.96**
	L	L	0.46	0.52	0.60	0.65	0.61	0.78	0.55	0.82	0.67	0.78	0.81	0.83	0.91	**0.95**
SV	R	R	0.55	0.51	0.69	0.67	0.55	0.80	0.55	0.79	0.68	0.78	0.81	0.82	0.92	**0.97**
	M	M	0.46	0.49	0.66	0.65	0.65	0.86	0.58	0.83	0.70	0.82	0.85	0.85	0.93	**0.97**

6.4.2 Experimental Results

Experimental results of all cases using the PKU-MMD dataset are depicted in Table 6.2. As it may be observed, in almost every case, fusion of more than one modalities outperforms corresponding single modalities. Notably, HF are able to boost the performance of PC and DI, upon fusion of each with them. However, the fusion of HF, PC and DI, in several cases did not provide a clear performance boost in all cases, while in several CV setups it showed decreased accuracy compared to fusion of HF with PC. Throughout our experiments, we also noticed that when the HF were fused with TF, it did not lead to an increment of accuracy, while fusion of all available features, performance was significantly lower, therefore, we did not include such cases in Table 6.2.

Continuing with the experimental results , it is evident that in cross-view setup, best accuracy is achieved when using all image transformations, fused with the PC. On the other hand, in cross-subject and single view setups, all available representations apart from HF are necessary in order to achieve best accuracy. Notably, and contrary to previous approaches that used only a single representation of skeletal data such as the view augmentation approach presented in Chapter 4, or the work of Vernikos et al. [285], it is herein demonstrated that even when the viewpoint used for training differs significantly from the one used for testing, the proposed fusion approach is able to demonstrate comparable performance to all other cases. In those cases, i.e., R-L and L-R, achieved accuracies were 0.90 and 0.89, respectively, which were slightly lower than other cases, indicating the lack of performance gap as in previous Chapters. In less challenging cross-view cases, performance ranged between 0.93–0.98 As expected from previous Chapters, high accuracy values are achieved both in single view (i.e., 0.95–0.97) and cross-view (i.e., 0.96)

cases.

Moreover, it should be clear that fusion of several representations is able to significantly boost the performance. For example, in cross-subject case the achieved accuracy using a single representation ranged between 0.58–0.80, while fusion allowed for an increase of approx. 20%. Also, note that while DI exhibit poor performance in cross-view cases, thus are unable to boost performance when fused with other modalities, they exhibit strong performance in single-view and cross-subject cases, which is the best per means of a single representations. Unsurprisingly, in those cases they are able to provide a strong performance boost. This was expected due to the fact that the data augmentation approach has been used only in cases of image transformations and pseudo-colored images. Furthermore, another reason for this, is the loss of visual information in case of viewpoint changes; in the PKU-MMD dataset skeletons always comprise 25 joints. It should also be emphasized that as demonstrated in Table 6.2, the fusion of all transforms (except from HF, as previously discussed) leads to improved accuracy compared to the cases of using a single one, meaning that the information extracted is complementary.

For the sake of comparison using the PKU-MMD dataset five state-of-the-art works, which to the best of our knowledge exhibit highest performances in the PKU-MMD dataset have been used. More specifically, the experimental results of this work in case of fusion of image transforms (TF) with pseudocolored images (PC) and dynamic images (DI) are compared to the ones of Li et al. [151], Li et al. [154], Li et al. [156] and also to the more recent works of Guo et al. [84] and Sun et al. [262].

Comparative results are depicted in Table 6.3. Note that in this comparison we used mAP@50 instead of accuracy. The authors of PKU-MMD propose the use of the following two tasks: (a) cross-subject, which is the same task as the one that we have already demonstrated; and (b) cross-view, wherein the case that is considered is the one that L and R viewpoints are used for training, while M is used for testing purposes. As it may be seen in Table 6.3, the proposed approach demonstrated improved performance in the cross-view case, while it shows inferior performance, yet comparable, to the results of Li et al. [156] which reported best results in the cross-subject case.

In case of the SYSU 3D HOI dataset, we compared our work using the same setup as in PKU-MMD with five state-of-the-art works which exhibit strong performance in both settings, namely to the ones of Zhang et al. [328], Zhang et al., [325] Zhang et al., [327] Ke et al., [123] and Hu et al. [102]. Comparative results are depicted in Table 6.4, where it may be seen that the proposed approach demonstrated best performance in both settings of the dataset. Moreover, in case of the UTKinect-Action3D dataset, we compared our work also using the same setup as with PKU-MMD and SYSU 3D HOI, with seven state-of-the-art works which exhibit strong performance, namely to the ones of Paoletti et al., [201] Koniusz et al., [130] Gao et al., [68] Tang et al. [268], Kao et al. [119], Avola et al. [11] and Zhang et al. [330]. Comparative results are depicted in Table 6.5, where it may be seen that the proposed approach demonstrated best performance.

Table 6.3: Comparison of the proposed approach to state-of-the-art research works using PKU-MMD dataset. Numbers indicate mAP (%). Note that Average Precision (AP) is calculated as the area under the precision-recall curve, while Mean Average Precision (mAP) extends the concept of AP by calculating the average AP values across multiple object classes. Bold indicate best results per task. CV, CS denote cross-view and cross-subject cases, respectively.

Experiment	**Methodology**					
	Proposed	[156]	[154]	[151]	[84]	[262]
CV	**95.1**	94.4	94.2	93.7	-	94.6
CS	92.1	**92.9**	92.6	90.4	87.8	93.2

Table 6.4: Comparison of the proposed approach to state-of-the-art research works using SYSU 3D HOI dataset. Numbers indicate accuracy (%). Bold indicate best results per task.

Experiment	**Methodology**					
	Proposed	[328]	[325]	[327]	[123]	[102]
Setting 1	**87.3**	86.9	85.1	85.7	-	79.6
Setting 2	**88.7**	86.5	84.8	85.7	83.3	84.9

Table 6.5: Comparison of the proposed approach to state-of-the-art research works using UT-Kinect Action3D dataset. Numbers indicate accuracy (%).

Methodology							
Proposed	[201]	[130]	[68]	[268]	[119]	[11]	[330]
99.6	99.5	99.2	98.5	98.5	96.0	95.7	95.6

6.5 Conclusions

In this Chapter, a multimodal fusion approach which targeted the problem of human activity recognition from video data was presented. Specifically, it was based on moving RGB and skeletal data, both captured by a depth camera. These modalities were used to create (a) 4 2D image representations, which were based on popular spectral transformations, i.e., the Discrete Fourier Transform, the Fast Fourier Transform, the Discrete Cosine Transform and the Discrete Sine Transform; (b) a pseudo-colored image which is formed in order to capture inter-joint differences over time; and (c) a dynamic image which is used to provide a single-frame visual “summary” of a video sequence. Moreover, a set of handcrafted features was also extracted from moving skeletal data. Experiments have been mainly performed using a dataset of human motion activities, which was recorded with a multi-camera setup and conducted a three-fold evaluation, i.e., a single view case where the same viewpoint was used for training/testing, a cross-view case where different viewpoints were used for training/testing and cross-subject case, where different subjects were used for training/testing.

Additional experiments were performed using two smaller-scale single camera datasets.

A CNN was trained for each case, which then was used for feature extraction. Thus, in order to classify a given activity sequence, all the aforementioned 2D images were first extracted and then the 6 convolutional neural networks were used so as to extract features. The latter were fused in several scenarios and fed into a support vector machine for classification. The experimental evaluation indicated that the proposed approach may be successfully used for HAR in all the aforementioned cases. Moreover, it has been shown that the fusion of several representations is able to boost performance in the cross-subject and single-view cases, however in case of cross-view the dynamic images are not necessary, since they cause a small, yet significant drop of performance, while in several cases handcrafted features are able to complement deep features. Also, upon comparison of our approach with state of the art approaches, superior performance in the cross-view case and comparable performance in the cross-subject case have been demonstrated.

7. DATA AUGMENTATION USING OCCLUSION FOR HAR

7.1 Introduction

As it well-known, one of the main challenges in several recognition tasks which are approached by using a deep learning architecture, is the size of the training data set [23]. Specifically, using small datasets for training deep learning models can lead to overfitting, i.e., models learn noise and fail to generalize to new data. Limited diversity in these datasets may not capture real-world variability, resulting in poor model performance and increased output variance. Furthermore, small datasets often provide insufficient information for deep learning models and robust model evaluation, leading to oversimplified models that do not accurately represent complex behaviors. To mitigate these issues, several approaches may applied, such as data augmentation and transfer learning, or even the choice of much simpler models.

In this Chapter we propose a data augmentation technique that is based on the creation of artificial occluded samples of activities. Specifically, contrary to Chapter 4, where we augmented the training data set by incorporating artificially rotated skeletons within the training process, in this Chapter we artificially remove body parts from the skeletons, so as to simulate the effect of occlusion. This way the network is trained with a significantly enhanced data set both in terms of size and diversity.

In the context of human activity recognition, occlusion refers to the partial or complete blockage of a person, which can prevent the activity from being fully visible and accurately recognized by a trained recognition model [71]. Possible causes for this effect include other objects such as furniture, or people being present in the scene, positioned between the camera and the subject, which is commonly referred to as “external occlusion,” or even by the subject’s own body parts, e.g., one arm obscuring the other during an action, which is referred to as “self-occlusion.” As expected, this disrupts the continuity of motion and obscures vital visual cues, which are necessary for accurate classification of actions. This becomes notably challenging in real-world scenarios where occlusion occurs [71, 287]. The impact of occlusion varies based on its extent, duration, and the importance of the occluded body parts in the action being performed [287]. For example, in an activity like “kicking something,” the movement of legs is crucial and their occlusion can lead to significant errors in recognition. In contrast, the same activity could still be recognizable even with occlusion of both arms.

As the field of human activity recognition continues to evolve, addressing occlusion effectively remains a key area of research. The development of more sophisticated models that can handle various types of occlusions not only improves the accuracy of action recognition systems but also expands their applicability in real-world environments where occlusions are a common occurrence. However, in this work we attempt to exploit the effect of occlusion as a means for training more robust recognition models. Note that contrary to typical research works, the novelty of the methodology that will be presented in this Chapter is the use of artificially occluded samples to augment the training data set, but

with the goal of improving the performance in the recognition of *non-occluded* samples.

The rest of this Chapter is organized as follows: in Section 7.2 we present related work regarding occlusion and data augmentation, in the context of HAR. Then, in Section 7.3 we present the proposed methodology for data augmentation using artificially created occluded samples. The proposed approach is evaluated and the corresponding results are discussed in Section 7.4 and finally, conclusions are drawn in Section 7.5.

7.2 Related Work

7.2.1 Occlusion in HAR

As it has already been mentioned, recent related work on occlusion focuses on tackling the problem of recognition of activities when one/more body parts are not fully visible. A study on the effect of occlusion on HAR has been presented by Giannakos et al. [71] who artificially created occluded skeletons, by removing structured body parts, i.e., corresponding to arms and legs, prior classification. At the following we present research works that deal with occlusion in the context of HAR, mainly focusing on the way body/skeleton parts are occluded.

Typical research works are based on extracted skeletons. Since to the best of our knowledge there do not exist any publicly available datasets which contain occluded samples, the majority of works create artificially occluded data samples. A popular approach is the removal of structured body parts, as proposed in [71]. Angelini et al. [7] considered the cases of persistent and short-time occlusions, i.e., when occlusion takes place within the whole or a part of the action. In the first case they proposed the use of action prototypes as a means of filling missing information. In the second case they used interpolation. They simulated occlusion by removing structured sets of skeleton joints, corresponding to body parts. Ghafoor et al. [70] experimented with missing both random and structured sets joints by using a temporal dilated CNN, utilizing temporal data to estimate missing joints and accordingly increase overall system robustness. Vernikos et al. [287] proposed a CNN-based method trained on 2D representations of 3D skeletal motion, including artificially occluded data samples and showed that this way performance in recognizing activities under occlusive conditions of structured body parts, may be significantly improved. Finally, Yang et al. [315] considered the body skeleton as a graph and simulated occlusion by removing sub-graphs. They also introduced a novel pose encoder incorporating a sub-graph data augmentation mechanism during training, which simulates occlusions, into a video pose encoder based on Graph Convolutional Networks (GCNs).

Other approaches are using still images/videos with occluded body parts, prior to the extraction of skeleton joints. Bian et al. [19] considered a multi-camera setup. In some cases, parts of the skeleton were not visible from some of the cameras. They introduced a novel multi-view spatial-temporal graph contrastive learning method, to develop view-invariant representations, robust to occlusions. Li et al. [162] extracted skeletons from images where body parts were occluded and proposed a recognition method that trans-

forms the occlusion problem into a process of imputing missing values in a feature matrix. Similarly, Gu et al. [83] simulated occlusions both by creating artificial occlusion masks and also by using incomplete skeletons. They then used temporal gated convolutions to recover missing body information. Moreover, Cheltha et al. [35] used images depicting partially occluded human subjects and then extracted incomplete skeletons. They addressed occlusions by incorporating multiple hypothesis tracking and recurrent neural networks, while Iosifidis et al. [109] adopted a multi-camera setup, wherein cameras were surrounding the subject, assuming that occlusion does not have an effect on all of them. Finally, Li et al. [157] worked on depth images by employing an action graph to model the dynamics of actions and utilized a bag-of-3D points to describe postures. They simulated occlusions by ignoring specific body areas.

7.2.2 Data Augmentation for HAR

Moreover, during the last few years, a plethora of research works focusing on data augmentation for HAR have emerged. At the following we present research works that deal with data augmentation on extracted skeletons. We begin with works applied in the spatial dimension of skeletal data. The work of Chen et al. [37], proposed a simple augmentation strategy consisting of simple transformations such as scalings, translations, rotations, and addition of random noise. Similarly, Park et al. [207] and Wang et al [300] also applied rotations as a means to artificially create different viewpoints of the skeleton. Moreover, Rao et al. also used rotations, shears, reverses (i.e., “flips”), gaussian noise, gaussian blur and also masks applied either to sets of joints or to joint coordinates, while Li et al. [151] used 3d rotations and Gaussian noise.

Additionally, several research efforts work in the temporal dimension of skeletons. Xin et al. [312] proposed temporal data augmentation techniques such as time reversing, interpolation, shifting, and warping. Li et al. [151] used video crops to trim actions. Chen et al. [37] used interpolation to simulate different durations of the same activity, as if they were performed by different actors, while Huynh-The et al. [104] proposed random elimination and addition of frames, so as to create artificial activity samples of different durations.

Notably, during the last few years, several works that are based on generative architectures such as generative adversarial networks have been presented. Such networks are able to create artificial skeleton samples/sequences, which can be used for augmentation of the training data set. Tu et al. [277] and Meng et al. [183] proposed an LSTM network, able to generate skeleton sequences which have similar features to the original ones. Shen et al. [239] proposed the Imaginative GAN, which is able to approximate the underlying distribution of the input skeletal data and then it samples new skeletons from this distribution. Wang et al. [306] proposed a method that is based on contrastive learning and uses both coordinates and velocities of skeletons, being able to provide augmented skeleton sequences.

The proposed approach, presented at the following has been partially inspired by the works of Vernikos et al. [288] and Angelini et al. [7] who included occluded samples in the

training process to improve classification performance of occluded samples. We remind that contrary to those approaches, our goal is to use data augmentation using occluded samples so as to improve classification performance of *non-occluded* samples.

7.3 Methodology

Similarly to previous Chapters, the herein proposed augmentation training strategy uses as input 3D trajectories of human skeletons, which are ultimately represented by activity images that have been created using the discrete sine transform (DST) (see subsection 3.3.2). Contrary to previous approaches we also include occluded samples in the training process, i.e., samples where structured set of joints corresponding to body parts have been removed.

7.3.1 Occlusion of Skeletal Data

In real-life scenarios, occlusion is one of the factors that hinder optimal performance of HAR approaches. Thus, as described in subsection 7.2.1, several research efforts include occluded samples in the training process, so as to provide robustness in recognition. To understand the notion of occlusion, let us consider an assistive living scenario, wherein one or more cameras placed in the subject's environment are used to capture its appearance and motion, so as to recognize her/his behaviour. Of course, in such a scenario, the living space of the subject may typically contain furniture or other objects. In several cases, actions may take place behind them, thus partial or even full occlusion may occur. During the former, some of the subject's body parts are partly or fully occluded. Occlusion may also occur when other people are present in the room. In all cases, occlusion is responsible for loss of visual information, which even in the case of simple actions may be crucial for recognition; e.g., considering the activity "handshaking," occlusion of arms would render a trained model unable to recognize it.

As we have already mentioned, amongst the goals of this Thesis is to assess whether the inclusion of occluded samples in the training process, i.e., the augmentation of the training data with artificially created occluded samples set is able to improve classification performance. However, publicly available, datasets such as the ones we use for the evaluation of this work, have been created in ideal conditions, i.e., the subject is captured in a well-illuminated, empty space to ensure visibility of the full body. This results to absence of occlusion samples.

Therefore, for the experimental evaluation of our work, we need to simulate the effect of occlusion. Given the paradigms of the works of [71, 83, 288], we chose to remove distinct body parts of the skeleton, i.e., structured sets of the skeleton joints. Specifically, in case of PKU-MMD dataset which has been captured using Kinect v2, and skeletons comprise 25 joints, we remove arms (each comprising a shoulder, an elbow, a wrist, a hand, a thumb and a handtip) and legs (each comprising a hip, a knee, an ankle and a

foot); the rest of the joints (i.e., head, spine shoulder, spine mid and spine base) form the torso, which we consider as non-occluded in all cases, as we experimentally verified through a series of initial experiments and given that Giannakos et al. [71] and Vernikos et al. [288] reached the same conclusion. SYSU-3D-HOI and UTKinect-Action3D, skeletons comprise 20 joints, differentiating in the arms (each comprising a shoulder, an elbow, a wrist and a hand) and the torso (comprising head, neck, spine shoulder, spine mid and spine base). In both cases and despite the aforementioned minor differences, we form 5 body parts, as illustrated in Fig. 2.17.

Moreover, also as in [71, 288], we assume that occlusion affects the whole duration of the activity. Also, intuitively, we consider the cases of occlusion of a) an arm; b) a leg; c) both arms; d) both legs; and e) an arm and a leg of the same side, i.e., a total of 8 cases. We assume that in a case where two parts are occluded, they should probably belong to the same side. We consider two cases of augmentation: augmenting training dataset using training samples wherein a) one or both arms have been occluded (resulting to $6\times$ original size); and b) all 8 aforementioned occlusion cases are considered (resulting to $9\times$ original size).

A visual overview of the proposed augmentation approach is illustrated in Fig. 7.1, while in Fig. 7.3 we illustrate an example of a full skeleton and all 8 cases of artificial occlusion. Moreover, in Fig. 7.2 we illustrate an example of an activity with the full skeleton and with occlusion of both arms. In this example it should be evident that the occlusion of these two body parts causes the loss of significant visual information rendering the performed activity unrecognizable.

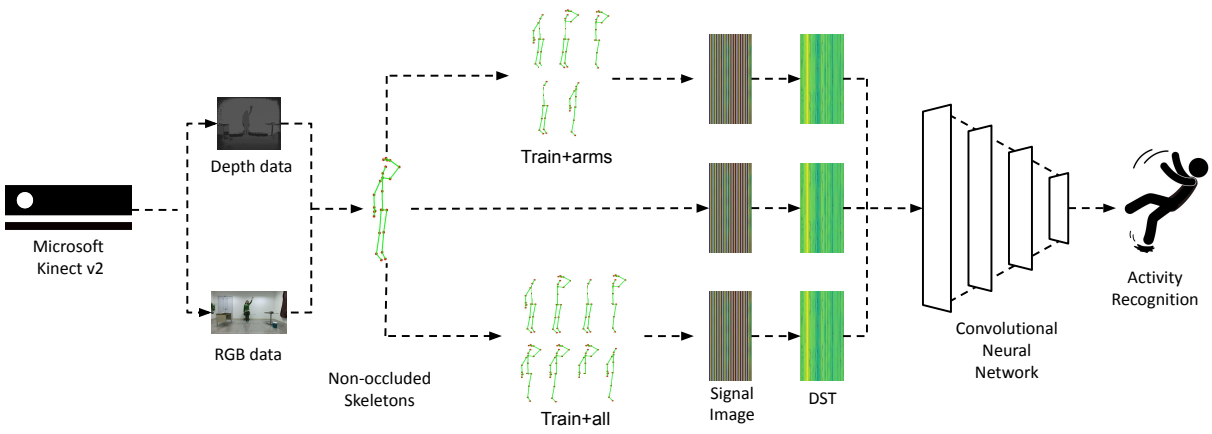


Figure 7.1: A visual overview of the proposed data augmentation methodology using occluded samples.

7.3.2 CNN Architecture

For classification we have once again used the CNN architecture that has been firstly presented in subsection 3.3.3 and is illustrated in Fig. 3.9. This architecture was successfully

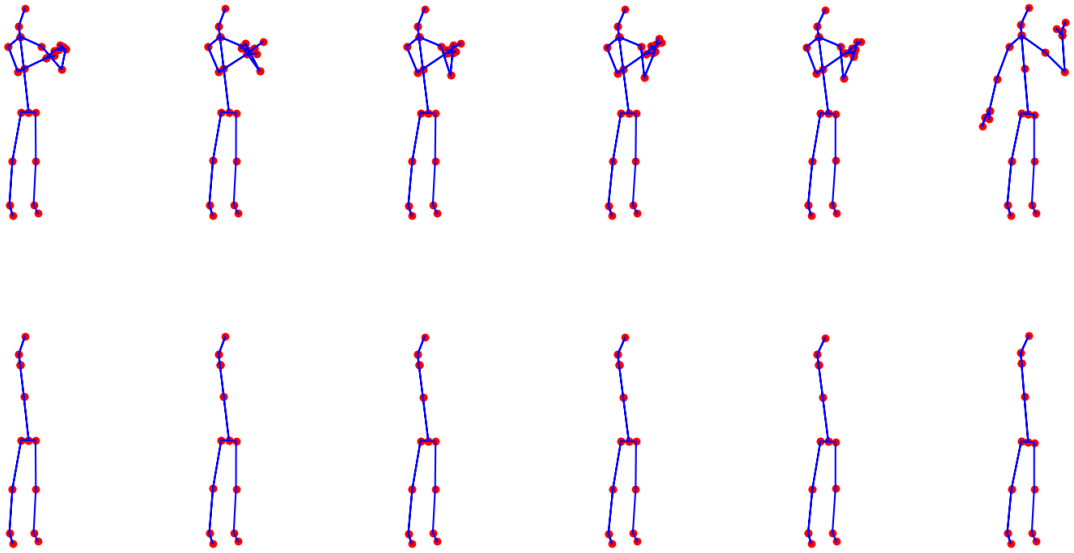


Figure 7.2: Example skeleton sequences of the activity *make phone call/answer phone* from the PKU-MMD dataset, captured by Microsoft Kinect v2. First row: original skeletons, including all 25 joints (i.e., without any occlusion); Second row: joints corresponding to both arms have been discarded (i.e., the skeleton is partially occluded).

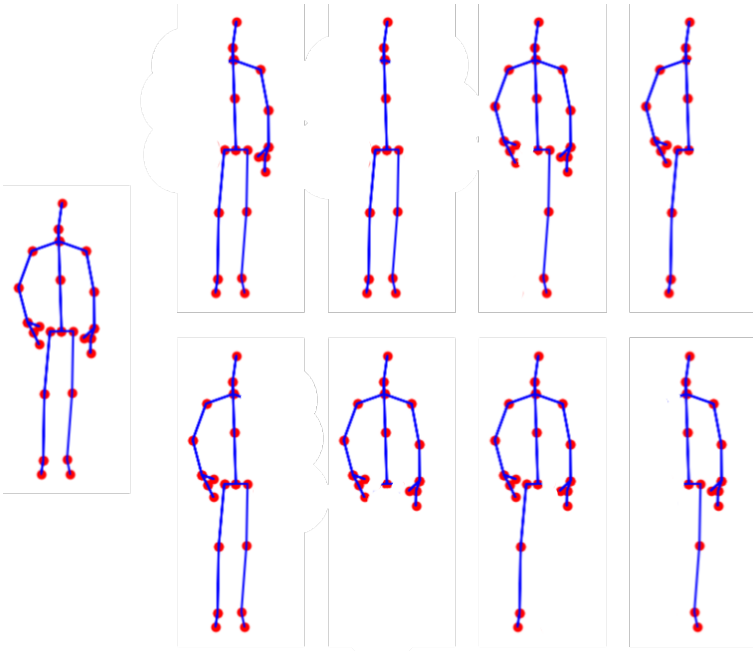


Figure 7.3: A full skeleton and the 8 cases of occlusion. First row: right arm, both arms, right leg, left arm and left leg; Second row: left arm, both legs, left leg, right arm and right leg.

applied to several problems of HAR using image representations of skeletal sequences (see Chapters 3–6). Therefore, we also use the same network for the evaluation of the herein proposed data augmentation approach. As in Chapter 3, the ReLU activation function has been used, the batch size has been to 32, the Adam optimizer has been used, the dropout was set to 0.5, the learning rate was set to 0.001. The network was trained using the loss of the validation set calculated via cross-entropy as an early stopping method and for 150 epochs.

7.4 Experimental Protocol and Results

Based on the experimental results that were presented in Chapter 3, in this Chapter we opt to evaluate the herein proposed approaches using the DST transformations of activity images, since this transform has demonstrated overall best performance. Therefore we create activity images simply by transforming the raw signal image by applying the DST transform. Note that due to data augmentation we use up to 8 times more data for training.

In order to experimentally evaluate the proposed multimodal HAR approaches, we once again used PKU-MMD, as in previous Chapters, and also SYSU 3D Human-Object Interaction (HOI) [102] and UTKinect-Action3D [311] datasets as in Chapter 6. All datasets have been previously presented in Subsection 2.4.1. We also remind that all these datasets consist of 3D human motion. Specifically, SYSU 3D-HOI focuses on interactions between humans and objects, UTKinect-Action3D comprises simple actions, while both are using a single-camera setup, contrary to the three-camera setup of PKU-MMD. Therefore, in case of PKU-MMD we consider single-view, cross-view and cross-subject evaluation cases (see subsection 2.4.2). Also in case of SYSU-3D-HOI we consider both single-view (setting 1) and cross-subject (setting 2), as suggested by the evaluation protocol imposed by the dataset's authors.

As in Chapter 3, we also investigate whether the proposed approach may be used for ambient assistive living scenarios. Therefore, we performed experiments using the following 11 classes of PKU-MMD, as follows: *eat meal snack*, *falling*, *handshaking*, *hugging other person*, *make a phone call*, *answer phone*, *playing with phone-tablet*, *reading*, *sitting down*, *standing up*, *typing on a keyboard* and *wear jacket*. Throughout our experiments with all datasets, we worked only based on the skeletal data, discarding RGB, depth and infrared information.

For the experimental evaluation of the proposed methodology, we considered the following training strategies:

a. a baseline approach, denoted as "Full," wherein the CNN is trained using only non-occluded samples, i.e., using the exact methodology presented in Chapter 3;

b. augmentation of the dataset of case [a.], denoted as "Full+Arms," with simulated occlusion examples for the cases of removal of structured sets of skeletal joints, corresponding to arms, i.e., (a) left arm; (b) right arm; (c) both arms; (d) left arm+left

leg; (e) right arm+right leg;

c. further augmentation of the dataset of case [b.], denoted as “Full+All,” with simulated occlusion examples for the cases of removal of structured sets of skeletal joints, corresponding to (a) left leg; (b) right leg; (c) both legs.

In all the aforementioned experiments we evaluate the performance of classification of non-occluded samples, using a CNN network (see subsection 7.3.2), trained with activity images. Experimental results for all datasets are depicted in Tables 7.1 – 7.3. In all cases we extract the mean accuracy.

Table 7.1: Results on PKU-MMD dataset. “Full” denotes the case of training dataset comprising non-occluded samples only; “Full+Arms” denotes the case of augmenting “Full” with samples where one or both arms have been occluded; “Full+All” denotes the case of augmenting “Full+Arms” with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies.

Experiment	Train	Test	Full	Full+Arms	Full+All
Single-view	M	M	0.86	**0.95**	0.85
	L	L	0.84	0.87	**0.88**
	R	R	0.87	**0.94**	0.90
Cross-view	M	L	0.64	**0.84**	0.83
	M	R	0.63	0.81	**0.86**
	L	M	0.72	0.81	**0.83**
	L	R	0.43	0.55	**0.56**
	R	M	0.63	0.85	**0.89**
	R	L	0.39	0.40	**0.45**
	M,L	R	0.62	**0.84**	0.82
	M,R	L	0.60	**0.78**	0.77
	L,R	M	0.82	0.94	**0.95**
Cross-subject	M,L,R	M,L,R	0.81	**0.88**	0.75

Table 7.2: Results on SYSU dataset. “Full” denotes the case of training dataset comprising non-occluded samples only; “Full+Arms” denotes the case of augmenting “Full” with samples where one or both arms have been occluded; “Full+All” denotes the case of augmenting “Full+Arms” with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies.

Experiment	Full	Full+Arms	Full+All
Single-view	0.58	0.62	**0.69**
Cross-subject	0.56	0.55	**0.60**

Table 7.3: Results on UTKinect-3D dataset. "Full" denotes the case of training dataset comprising non-occluded samples only; "Full+Arms" denotes the case of augmenting "Full" with samples where one or both arms have been occluded; "Full+All" denotes the case of augmenting "Full+Arms" with samples where arms are always present but one or both arms/legs of the same side have been occluded. Numbers denote Accuracy, while numbers in bold indicate best performance between the three training strategies.

Experiment	**Full**	**Full+Arms**	**Full+All**
Single-view	0.85	**0.90**	0.85

In case of the PKU-MMD dataset, for single-view experiments, we should note the following: in case of M and R cameras, augmentation case "Full+Arms" exhibited best accuracy in both cases, with values 0.95 and 0.94, respectively. However, in case of L camera, "Full+All" exhibited marginally better accuracy than the one of "Full+Arms," i.e., equal to 0.94. In all cases, augmentation demonstrated significantly increased accuracy compared to the baseline. Moreover, comparing average accuracy between all single-view cases, we could argue that "Full+Arms" demonstrated better overall performance. In case of cross-view experiments, in most cases "Full+All" demonstrated better performance, even though in several cases the differences were marginal. However, once again the improvement imposed by the augmentation step over the baseline approach was clear in all cases. Comparing average accuracy between all cross-view cases, we could argue that "Full+Arms" again demonstrated better overall performance. Finally, in case of cross-subject, interestingly "Full+Arms" exhibited best performance, i.e., equal to 0.88, while "Full+All" was significantly lower than the baseline case.

In case of SYSU dataset, and in the single-view experiments, we noticed a significant increase of performance. Specifically, in case of single-view experiments (Setting 1), without augmentation the accuracy was 0.58, reaching 0.69 in case of "Full+All," i.e., exhibiting an increase of 19%. Similarly, in case of cross-subject experiments (Setting 2), without augmentation the accuracy was 0.56, reaching 0.60 in case of "Full+Arms," i.e., exhibiting an increase of 7%. Finally, in case of UTKinect-3D dataset, without augmentation the accuracy was 0.85, reaching 0.90 in case of "Full+Arms," i.e., exhibiting an increase of 5.9%. Interestingly, "Full+All" showed equal accuracy with the baseline case.

7.5 Conclusions

In this Chapter, a data augmentation approach which was based on the artificial occlusion of body parts, and targeted the problem of human activity recognition from video data was presented. Specifically we used 3D skeletons from which we manually removed one or two body parts. We performed experiments where the training data set was firstly augmented using samples where at least one arm was occluded and we further augmented it using samples where at least one leg was occluded. Experiments have been performed using three datasets of human motion activities, which were recorded with single and multi-

camera setups. In the latter case, we conducted a three-fold evaluation, i.e., a single view case where the same viewpoint was used for training/testing, a cross-view case where different viewpoints were used for training/testing and cross-subject case, where different subjects were used for training/testing. Actions were represented using activity images created by applying the Discrete Sine Transform on raw motion data (see Chapter 3) and a CNN was trained for each dataset and for each augmentation case. The experimental evaluation indicated that the proposed approach may be successfully used for HAR in most the aforementioned cases, as it is able to provide a significant performance boost over the baseline approaches.

8. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this Thesis we presented our research efforts within the broad area of human activity recognition using visual data, focusing on a) the representation of human motion; and b) training strategies for deep learning architectures. Specifically, our work was based on human motion that was modeled using a structured set of skeletal joints, extracted from depth data. Moreover, we considered the problem of segmented activity recognition, i.e., the case when the temporal activity boundaries are known. We first proposed an approach for the 2D representation of human motion. Then, we extended this approach so as to tackle the problem of cross-view recognition, i.e., of abrupt viewpoint changes that are unseen in training, while augmenting the training data set. We continued by experimenting with a domain adaptation approach, as a means to align skeleton data distributions captured under different viewpoints. We then experimented with fusion of multiple visual modalities so as to assess whether they provide complementary feature representations and we concluded with an approach for augmentation of the training data set with artificially occluded skeleton sequences, in an effort to increase the size and the diversity of the training data set.

8.1 Conclusions

Specifically, in the first research Chapter of this Thesis (Chapter 3) we presented an approach for the 2D representation of 3D skeletal data. These data have been captured using the Kinect RGB+D camera and consist from the 3D positions of a set of skeleton joints, as they move in space and over time, during an action. We first created “signal” images by concatenating raw signals corresponding to joint motion and then we applied four well-known spectral transforms, so as to capture the spectral properties of this motion. This way, the 3D skeleton motion was represented with 2D images, which were then used to train convolutional neural networks which classified human activities. We showed that these representations may be successfully applied to the problem of human activity recognition and were able to differentiate between different activities within experimental evaluation using a challenging publicly available and large scale human motion datasets and within single-view, cross-view and cross-subject experimental setups. However, we noticed a significant performance drop in cross-view, compared to the other setups which was clearer within the more abrupt changes.

Therefore, in the second research Chapter of this Thesis (Chapter 4) we shifted our focus into enhancing the previous approach so as to be more robust into abrupt viewpoint changes. We presented a geometric data augmentation approach which was applied to raw skeleton data and aimed to enhance the training data set with artificially rotated examples. Specifically, considering a multi-camera setup, our goal was to improve the performance in cases where a dataset is trained using a given camera, but evaluation takes place using another camera. Typically, performance is expected to fall when changes of viewpoint are small, but in more abrupt cases this may render the algorithm unusable in

real-life situations. To this, we artificially created “rotated” skeleton sequences by applying a 3D rotation transformation, so as to allow training with viewpoints previously unseen. We demonstrated that this augmentation step is able to provide a significant boost of performance in cross-view setups, while also improving the cross-subject case. Moreover, we came up with a simple heuristic, wherein a testing example is processed prior to classification so as to be “aligned” to the viewpoint that was used for training of the model. We experimentally verified that this heuristic was able to provide a further increase of performance in both setups.

However, since the collection, the annotation and even the augmentation of a large data set may be a time consuming and expensive task, in the third research Chapter of this Thesis (Chapter 5) we experimented with a domain adaptation approach, extending the works previously presented. The goal of domain adaptation is to mitigate the domain shift problem, i.e., the cases which occurs when the statistical characteristics of captured data change between different domains, and typically causes under-performance of models trained in one domain when applied in another. In our work, as domains we consider the different viewpoints, i.e., we work within a cross-view setup, i.e., a case wherein a classifier is trained using samples captured under a given viewpoint and is evaluated using samples captured another one. We assume that the second viewpoint is unlabelled, apart from a small part of it, i.e., we work in a semi-supervised setting. We propose an adversarial domain adaptation algorithm and we show that generalization between different viewpoints may be achieved without data augmentation, thus we effectively tackle the cross-view shift.

Our experiments show that when target data are fully unlabelled, data augmentation offers a huge increase over the source model. However, a small amount of labeled data makes domain adaptation approach to exhibit clearly improved performance overall, making it feasible for real-life applications. For example, consider an assistive living scenario, where a single camera is used to monitor the behaviour of a human subject. Typically, in such scenarios the space where subjects act are not strictly limited. Therefore, the viewpoint may significantly change. Now, consider an activity recognition model that has been pretrained in laboratory conditions. In case that the subject is able to participate in a data collection process, it should be more effective to adopt a domain adaptation approach. Of course, if this is not possible, then data augmentation is able to offer an adequate alternative, while it is able to be immediately applied.

In the fourth research Chapter of this Thesis (Chapter 6) we presented multimodal fusion approach, based on both RGB and skeletal data. As in Chapter 3 we used these modalities to create the four 2D image representations of the spectral properties of motion and we additionally used a pseudo-colored image which is formed so as to capture inter-joint differences over time and also dynamic images which were used to provide a single-frame visual “summary” of a video sequence. We also experimented with a set of hand-crafted features, extracted mainly from angles and displacements between joints and performed experiments with fusion of two or more modalities, so as to assess the complementarity of features. We show that the fusion of all visual representations is able to boost performance in the cross-subject and single-view cases, however in case of cross-view the

dynamic images are not necessary, since they cause a small, yet significant drop of performance. Also, upon comparison of our approach with state of the art approaches, using three datasets, superior performance in the cross-view case and comparable performance in the cross-subject case have been demonstrated.

One of the main advantages of this fusion approach is that it is not tied to a single modality. Instead it may be used with more than one modalities. Depending on the available hardware, different types of cameras may be used, while the extraction of skeletal data may be performed with any available methodology, although it is preferred to use 3D skeletal data. Also, a different deep network is used for feature extraction for each representation, allowing for faster training and extraction times. As demonstrated, the approach could be also used with single-camera datasets, without any modification, apart from training data and with RGB datasets (i.e., without depth information), although some performance loss is expected. The main limitation is the need for a multi-camera setup, in case of cross-view experiments. Also, as the majority of contemporary research works, it does not deal with spatial and/or temporal occlusion. However, as it will be later discussed, this could be a possible future research direction.

In the final research Chapter of this Thesis (Chapter 7) we once again present a data augmentation approach, so as to create a larger, more diverse training data set which is based on the addition of artificially created occluded activity samples. Since to the best of our knowledge there do not exist any publicly available datasets that involve occlusion, and following recent related works, we artificially create occluded samples by manually removing structured sets of skeletal joints that correspond to body parts (i.e., to arms and legs). We incorporate these samples into the training dataset and we demonstrate that this approach is able to significantly increase the performance in all experimental cases and for three datasets.

We should note that since all the herein presented approaches are based on skeleton data, a limitation of all our works is that these data should be somehow available, either using some specific hardware such as the Microsoft Kinect and its API, or some skeleton extraction library such as OpenPose [30], Posenet [125]. In such cases, since depth information is not available, a performance drop should be expected. Although they could be extracted without the need for specialized video capturing hardware, their extraction is by far more computationally expensive; its use for real-time applications may be impractical without using a modern GPU. Of course, reliable skeleton extraction is prone to illumination/viewpoint changes and occlusion. In this work we deal only with the viewpoint changes and under several assumptions that are quite “loose” compared to real applications, i.e., we consider a limited number of viewpoints, which is not the case in a real-life scenario. Moreover, we focus on low-cost sensors, allowing the proposed methodologies to be more easily applicable in real-world scenarios.

We believe that the proposed approaches could benefit several HAR tasks which take place into dynamic environments. The proposed algorithms could be deployed e.g., within an assisted living environment (e.g., a smart home), for monitoring activities of daily living (ADLs) [141], or medical-related events that can be recognized by motion and may require some kind of attention or intervention. Another application could be the recognition of hu-

man motion within Augmented Reality environments and applications. In such a case, human motion may offer useful cues for assessing user engagement or satisfaction. The latter could be a possible application in the area of retail and marketing, upon the recognition of behaviors and interactions with products, or in the area of sports and fitness, where it could act as a means of assessing performance. Surveillance and security applications may also benefit, e.g., for the detection of suspicious activities or individuals. Moreover, it could benefit the broader area of human-computer interaction, by recognizing gestures and actions to control interfaces or interact with virtual objects or even provide assistive interaction approaches for people with disabilities, or incorporation of player movements into gameplay mechanics, in the area of gaming. Finally, another possible application could the analysis of classroom activities and student engagement within an education environment.

8.2 Future Research Directions

Possible future extensions of this work include investigation on methods for creating the signal image, possibly with the use of other types of sensor measurements such as wearable accelerometers, gyroscopes etc. and evaluation of the proposed approach on several other public datasets, and for other types of activities. Within this process, a temporal augmentation approach [137] applied on skeletal data could replace the interpolation step herein used. Moreover, a further continuation of this work could involve other types of image representations of signals such as recurrence plots, [178] which have been successfully used with deep CNNs for classification of time-series, [88] deep architectures, such as Visual Transformers [86], LSTMs and hybrid CNN-LSTM networks [259], or techniques that work with sequences, such as seq2seq [265] or time-series such as multivariate CNNs [164].

Since handcrafted features have been shown to boost recognition performance of deep approaches, [132] it would be interesting to experiment with other methodologies for feature extraction that are based on the geometry and the motion of skeletons. [11, 330] Also, the use of modern classifiers could be investigated for classification. For example, instead of using an SVM, other possible research directions could include experiments with approaches such as Neural Dynamic Classification, [218] Dynamic Ensemble Learning, [4] and Finite Element Machine for fast learning [213].

ABBREVIATIONS - ACRONYMS

1D	1-Dimensional
2D	2-Dimensional
3D	3-Dimensional
5D	5-Dimensional
ADDA	Adversarial Discriminative Domain Adaptation
ADL	Activity of Daily Living
AHE	Adaptive Histogram Equalization
AI	Artificial Intelligence
ANN	Artificial Neural Network
AR	Augmented Reality
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DL	Deep Learning
DST	Discrete Sine Transform
DT	Decision Tree
DTW	Dynamic Time Warping Algorithm
ERT	Extremely Randomized Tree
FFT	Fast Fourier Transform
FN	False Negative
FNN	Feed Forward Neural Network
FP	False Positivre
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GMM	Gaussian Mixture Model
GPU	Graphic Processing Unit

HAR	Human Action Recognition
HMM	Hidden Markov Model
HOG	Histogram of Oriented Gradient
HOI	Human-Object Interaction
JDM	Joint Distance Maps
JPG	JPEG-Joint Photographic Experts Group
JTM	Joint Trajectory Maps
KNN	K-Nearest Neighbor
LSTM	Long Short-Term Memory
ML	Machine Learning
MoCap	Motion Capture (systems)
NLP	Natural Language Processing
NN	Neural Network
PCA	Principal Component Analysis
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
ResNet	Residual Neural Network
RFID	Radio-Frequency IDentification
RGB	Red Green Blue
RGB-D	Red Green Blue and Depth data
RNN	Recurrent Neural Network
SDK	Software Development Kit
So-HoT	Second- or Higher-Order Scatter Tensors
SVM	Support Vector Machine
SPMF	Skeleton Posture-Motion Feature
TN	True Negative
TP	True Positive
TSSI	Tree Structure Skeleton Image
VGG	Very Deep Convolutional Network

APPENDIX A. SHORT CV

Eirini Mathe holds a Bachelor's degree (BSc) in Computer Engineering T.E. from the Technological Educational Institute of Central Greece (2014). She also holds a Master's degree (MSc) in Information Technologies in Medicine and Biology from the Department of Informatics and Telecommunications of the National and Kapodistrian University of Athens (2017). Since 2019 she is pursuing her PhD titled "Neural Network Training Techniques for Human Activity Recognition Using Visual Data" from the Department of Informatics of the Ionian University.

She has worked as a research assistant at the Institute of Informatics and Telecommunications of the National Center for Scientific Research - "Demokritos" (2018-2022), and as an external collaborator at the Ionian University (2019-present). Additionally, she has taught as an academic scholar in the Undergraduate Program of "Computer Engineering T.E." at the University of Thessaly (2019-2021). She has 5 publications in international scientific journals and 24 publications in international scientific conferences. Her research interests lie in the areas of computer vision, machine learning, and artificial intelligence.

APPENDIX B. LIST OF PUBLICATIONS

B.1 Journals

J05 I. Vernikos, E. Spyrou, I.-A. Kostis, **E. Mathe**, Ph. Mylonas, A Deep Regression Approach for Human Activity Recognition Under Partial Occlusion, International Journal of Neural Systems, 33 (09), August 2023.

J04 D. Koutrintzes, E. Spyrou, **E. Mathe**, Ph. Mylonas, A Multimodal Fusion Approach for Human Activity Recognition, International Journal of Neural Systems, 33 (01), January 2023.

J03 E. Spyrou, **E. Mathe**, G. Pikramenos, K. Kechagias and Ph. Mylonas, Data Augmentation vs Domain Adaptation: a case study in Human Activity Recognition, Technologies, 8(4), 55, MDPI, October 2020

J02 G. Pikramenos, **E. Mathe**, E. Vali, I. Vernikos, A. Papadakis, E. Spyrou and Ph. Mylonas, An Adversarial Semi-Supervised Approach to Action Recognition from Pose Information, Neural Computing and Applications, Springer, June 2020

J01 **E. Mathe** and E. Spyrou, User Evaluation of Low-cost, Non-invasive Methods for Heart-Rate Measurement, Journal of Usability Studies, vol.15, issue 3, UXPA, 2020

B.2 Conferences

C20 **E. Mathe**, D. Mouratidis, Y. Voutos, K.-L. Kermanidis, Ph. Mylonas, A. Kanavos, ShipQuest: a Bottom-Up Approach for Building a Modern Greek Maritime Ontology, 18th International Workshop on Semantic and Social Media Adaptation & Personalization (SMAP) 2023

C19. D. Mouratidis, **E. Mathe**, Y. Voutos, K. Stamou, K. Kermanidis, P. Mylonas and A. Kanavos, Domain-Specific Term Extraction: A Case Study on Greek Maritime Legal Texts. In Proc. of Hellenic Conference on Artificial Intelligence (SETN), 2022

C18. I. Vernikos, D. Koutrintzes, **E. Mathe**, E. Spyrou and Ph. Mylonas, Early Fusion of Visual Representations of Skeletal Data for Human Activity Recognition. In Proc. of Hellenic Conference on Artificial Intelligence (SETN), 2022

C17. I.A. Kostis, **E. Mathe**, E. Spyrou and Ph. Mylonas, Human Activity Recognition under Partial Occlusion. In Proc. of Int'l Conf. on Engineering Applications of Neural Networks (EANN), 2022

C16. D. Koutrintzes, **E. Mathe** and E. Spyrou. Boosting the Performance of Deep Approaches through Fusion with Handcrafted Features. In Proc. of Int'l Conf. on Pattern Recognition Applications and Methods (ICPRAM), 2022

C15. S. Stylianou-Nikolaidou, I. Vernikos, **E. Mathe** and E. Spyrou, A Deep Hybrid Architecture for Human Activity Recognition. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2021

C14. S. Stylianou-Nikolaidou, I. Vernikos, **E. Mathe**, E. Spyrou and Ph. Mylonas, A Novel CNN-LSTM Hybrid Architecture for the Recognition of Human Activities. In Proc. of Int'l Conf. on Engineering Applications of Neural Networks (EANN), 2021

C13. I. Giannakos, **E. Mathe**, E. Spyrou and Ph. Mylonas, A study on the Effect of Occlusion inHuman Activity Recognition. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2021

C12. **E. Mathe**, A. Tranou, E. Spyrou and S. Perantonis, Human Action Recognition with Deep Learning Techniques. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2020

C11. A. Papadakis, I. Vernikos, **E. Mathe** and E. Spyrou, Skeleton Geometric Transformation For Human Action Recognition Using Convolutional Neural Networks. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2020

C10. A. Papadakis, **E. Mathe** and E. Spyrou, Human activity recognition using 3D skeletal information. In Proc. of 1st Summit on Gender Equality in Computing (GEC), 2019

C9. I. Vernikos, **E. Mathe**, A. Mitsou and E. Spyrou, The VISOR project: Virtual coachIng Services for OldeR adults. In Proc. of 1st Summit on Gender Equality in Computing (GEC), 2019

C8. E. Spyrou, **E. Mathe**, I. Vernikos, P. Dedousis, A. Vlachou and Ph. Mylonas. The VISOR project: Virtual coachIng Services for OldeR adults. In Proc. of South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), 2019

C7. A. Papadakis, **E. Mathe**, E. Spyrou and Ph. Mylonas. A Geometric Approach for Cross-View Human Action Recognition using Deep Learning. In Proc. of Int'l Symposium on Image and Signal Processing and Analysis (ISPA), 2019

C6. I. Vernikos, **E. Mathe**, E. Spyrou, A. Mitsou, T. Giannakopoulos and Ph. Mylonas, Fusing Handcrafted and Contextual Features for Human Activity Recognition. In Proc. of Int'l Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), 2019

C5. I. Vernikos, **E. Mathe**, A. Papadakis, E. Spyrou and Ph. Mylonas, An Image Representation of Skeletal Data for Action Recognition using Convolutional Neural Networks. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2019

C4. A. Papadakis, **E. Mathe**, I. Vernikos, A. Maniatis, E. Spyrou and Ph. Mylonas, Recognizing Human Actions using 3D Skeletal Information and CNNs. In Proc. of Int'l Conf. on Engineering Applications of Neural Networks (EANN),2019

C3. **E. Mathe**, A. Maniatis, E. Spyrou and Ph. Mylonas, A Deep Learning Approach for Human Action Recognition using Skeletal Information. In Proc. of World Congress "Genetics,Geriatrics and Neurodegenerative Diseases Research" (GeNeDiS), 2018

C2. **E.Mathe** , A. Mitsou, , E.Spyrou and Mylonas, Arm Gesture Recognition using a Convolutional Neural Network. In 2018 13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP), 2018

C1. **E. Mathe** and E. Spyrou, User Evaluation of Unobtrusive Methods for Heart-Rate Measurement. In Proc. of International Conference on PErvasive Technologies Related to Assistive Environments (PETRA), 2018

BIBLIOGRAPHY

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). TensorFlow: a system for Large-Scale machine learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

[2] Agrawal, K. (2010). To study the phenomenon of the Moravec's paradox. arXiv preprint arXiv:1012.3148.

[3] Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., & Marchand, M. (2014). Domain-adversarial neural networks. arXiv preprint arXiv:1412.4446.

[4] Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. Neural Computing and Applications, 32(12), 8675-8690.

[5] Albrecht, T., & Muller, M. (2009). Dynamic time warping (dtw). Information Retreival for Music and Motion, 70-83.

[6] Anello, E., (2021). Visualizing the Feature Maps and Filters by Convolutional Neural Networks, Retrieved from `https://medium.com/dataseries/visualizing-the-feature-maps-and-filters-by-convolutional-neural-networks-e1462340518e`, 28 April, 2024.

[7] Angelini, F., Fu, Z., Long, Y., Shao, L., & Naqvi, S. M. (2019). 2D pose-based real-time human action recognition with occlusion-handling. IEEE Transactions on Multimedia, 22(6), 1433-1446.

[8] Ann, O. C., & Theng, L. B. (2014, November). Human activity recognition: A review. In 2014 IEEE international conference on control system, computing and engineering (ICCSCE 2014) (pp. 389-393). IEEE.

[9] Anuj, A., Mallick, T., Das, P. P., & Majumdar, A. K. (2015, December). Robust control of applications by hand-gestures. In 2015 Fifth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG) (pp. 1-4). IEEE.

[10] Arjovsky, M., & Bottou, L. (2017). Towards principled methods for training generative adversarial networks. arXiv preprint arXiv:1701.04862.

[11] Avola, D., Cascio, M., Cinque, L., Foresti, G. L., Massaroni, C., & Rodolà, E. (2019). 2-D skeleton-based action recognition via two-branch stacked LSTM-RNNs. IEEE Transactions on Multimedia, 22(10), 2481-2496.

[12] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2011). Sequential deep learning for human action recognition. In Human Behavior Understanding: Second International Workshop, HBU 2011, Amsterdam, The Netherlands, November 16, 2011. Proceedings 2 (pp. 29-39). Springer Berlin Heidelberg.

[13] Barr, A., Feigenbaum, E. A., & Cohen, P. R. (Eds.). (1981). The handbook of artificial intelligence (Vol. 3). HeurisTech Press.

[14] Basly, H., Ouarda, W., Sayadi, F. E., Ouni, B., & Alimi, A. M. (2020). CNN-SVM learning approach based human activity recognition. In Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9 (pp. 271-281). Springer International Publishing.

[15] Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Vaughan, J. W. (2010). A theory of learning from different domains. Machine learning, 79(1-2), 151-175.

[16] Berretti, S., Daoudi, M., Turaga, P., & Basu, A. (2018). Representation, analysis, and recognition of 3D humans: A survey. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 14(1s), 1-36.

[17] Bilen, H., Fernando, B., Gavves, E., Vedaldi, A., & Gould, S. (2016). Dynamic image networks for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3034-3042).

[18] Bhattacharya, S., Czejdo, B., & Perez, N. (2012, November). Gesture classification with machine learning using kinect sensor data. In 2012 Third International Conference on Emerging Applications of Information Technology (pp. 348-351). IEEE.

[19] Bian, C., Feng, W., Meng, F., & Wang, S. (2022). View-Invariant Skeleton-based Action Recognition via Global-Local Contrastive Learning. arXiv preprint arXiv:2209.11634.

[20] Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., & Wortman, J. (2008). Learning bounds for domain adaptation. In Advances in neural information processing systems (pp. 129-136).

[21] Bloom, V., Makris, D., & Argyriou, V. (2012, June). G3D: A gaming action dataset and real time action recognition evaluation framework. In 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (pp. 7-12). IEEE.

[22] Bradski, G., (2000) The OpenCV Library Dr. Dobb's Journal of Software Tools.

[23] Brigato, L., & Iocchi, L. (2021, January). A close look at deep learning with small data. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 2490-2497). IEEE.

[24] Buch, S., Escorcia, V., Shen, C., Ghanem, B., & Carlos Niebles, J. (2017). Sst: Single-stream temporal action proposals. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 2911-2920).

[25] Caba Heilbron, F., Escorcia, V., Ghanem, B., & Carlos Niebles, J. (2015). Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the ieee conference on computer vision and pattern recognition (pp. 961-970).

[26] Camgoz, N. C., Hadfield, S., Koller, O., & Bowden, R. (2016, December). Using convolutional 3d neural networks for user-independent continuous gesture recognition. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 49-54). IEEE.

[27] Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7291-7299).

[28] Cao, Z., Long, M., Wang, J., & Jordan, M. I. (2018). Partial transfer learning with selective adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2724-2732).

[29] Cao, Z., Ma, L., Long, M., & Wang, J. (2018). Partial adversarial domain adaptation. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 135-150).

[30] Cao, Z., Hidalgo, G., Simon, T., Wei, S. E., & Sheikh, Y. (2018). Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008, 1.

[31] Cao, Z., You, K., Long, M., Wang, J., & Yang, Q. (2019). Learning to transfer examples for partial domain adaptation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2985-2994).

[32] Celebi, S., Aydin, A. S., Temiz, T. T., & Arici, T. (2013, February). Gesture recognition using skeleton data with weighted dynamic time warping. In VISAPP (1) (pp. 620-625).

[33] Chaaraoui, A., Padilla-Lopez, J., & Flórez-Revuelta, F. (2013). Fusion of skeletal and silhouette-based features for human action recognition with RGB-D devices. In Proceedings of the IEEE international conference on computer vision workshops (pp. 91-97).

[34] Chai, X., Liu, Z., Yin, F., Liu, Z., & Chen, X. (2016, December). Two streams recurrent neural networks for large-scale continuous gesture recognition. In 2016 23rd international conference on pattern recognition (ICPR) (pp. 31-36). IEEE.

[35] Cheltha, J. N., Sharma, C., Dadheech, P., & Goyal, D. (2024). Overcoming Occlusion Challenges in Human Motion Detection through Advanced Deep Learning Techniques. International Journal of Intelligent Systems and Applications in Engineering, 12(17s), 497-513.

[36] Chen, B. (2010). Deep learning of invariant spatio-temporal features from video (Doctoral dissertation, University of British Columbia).

[37] Chen, J., Yang, W., Liu, C., & Yao, L. (2021). A data augmentation method for skeleton-based action recognition with relative features. Applied Sciences, 11(23), 11481.

[38] Chollet, F. (2015) Keras, https://github.com/fchollet/keras.

[39] Cover, T. M. (1999). Elements of information theory. John Wiley & Sons.

[40] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), 53-65.

[41] Csurka, G. (2017). A comprehensive survey on domain adaptation for visual applications. Domain adaptation in computer vision applications, 1-35.

[42] Dave, A., Russakovsky, O., & Ramanan, D. (2017). Predictive-corrective networks for action detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 981-990).

[43] De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., & Tuytelaars, T. (2016). Online action detection. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14 (pp. 269-284). Springer International Publishing.

[44] Debes, C., Merentitis, A., Sukhanov, S., Niessen, M., Frangiadakis, N., & Bauer, A. (2016). Monitoring activities of daily living in smart homes: Understanding human behavior. IEEE Signal Processing Magazine, 33(2), 81-94.

[45] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[46] Deng, J., Zhang, Z., Eyben, F.,& Schuller, B. (2014). Autoencoder-based unsupervised domain adaptation for speech emotion recognition. IEEE Signal Processing Letters, 21(9), 1068-1072.

[47] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[48] Ding, J., Chen, B., Liu, H., & Huang, M. (2016). Convolutional neural network with data augmentation for SAR target recognition. IEEE Geoscience and remote sensing letters, 13(3), 364-368.

[49] Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., & Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2625-2634).

[50] Du, Y., Fu, Y., & Wang, L. (2015, November). Skeleton based action recognition with convolutional neural network. In 2015 3rd IAPR Asian conference on pattern recognition (ACPR) (pp. 579-583). IEEE.

[51] Du, Y., Wang, W., & Wang, L. (2015). Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1110-1118).

[52] Du, Y., Fu, Y., & Wang, L. (2016). Representation learning of temporal dynamics for skeleton-based action recognition. IEEE Transactions on Image Processing, 25(7), 3010-3022.

[53] Du, W., Wang, Y., & Qiao, Y. (2017). Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In Proceedings of the IEEE international conference on computer vision (pp. 3725-3734).

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

[55] Ehatisham-Ul-Haq, M., Javed, A., Azam, M. A., Malik, H. M., Irtaza, A., Lee, I. H., & Mahmood, M. T. (2019). Robust human activity recognition using multimodal feature-level fusion. IEEE Access, 7, 60736-60751.

[56] Escalera, S., Baró, X., Gonzalez, J., Bautista, M. A., Madadi, M., Reyes, M., ... & Guyon, I. (2015). Chalearn looking at people challenge 2014: Dataset and results. In Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13 (pp. 459-473). Springer International Publishing.

[57] Escorcia, V., Caba Heilbron, F., Niebles, J. C., & Ghanem, B. (2016). Daps: Deep action proposals for action understanding. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14 (pp. 768-784). Springer International Publishing.

[58] Farahani, A., Voghoei, S., Rasheed, K., & Arabnia, H. R. (2021). A brief review of domain adaptation. Advances in data science and information engineering: proceedings from ICDATA 2020 and IKE 2020, 877-894.

[59] Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1933-1941).

[60] Feigenbaum, E. A. (1981). Expert systems in the 1980s. State of the art report on machine intelligence. Maidenhead: Pergamon-Infotech, 23.

[61] Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., & Hovy, E. (2021). A survey of data augmentation approaches for NLP. arXiv preprint arXiv:2105.03075.

[62] Fernando, B., Gavves, E., Oramas, J. M., Ghodrati, A., & Tuytelaars, T. (2015). Modeling video evolution for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5378-5387).

[63] Fernando, B., Anderson, P., Hutter, M., & Gould, S. (2016). Discriminative hierarchical rank pooling for activity recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1924-1932).

[64] Fernando, B., & Gould, S. (2016, June). Learning end-to-end video classification with rank-pooling. In International Conference on Machine Learning (pp. 1187-1196). PMLR.

[65] Fothergill, S., Mentis, H., Kohli, P., & Nowozin, S. (2012, May). Instructing people for training gestural interactive systems. In Proceedings of the SIGCHI conference on human factors in computing systems (pp. 1737-1746).

[66] Ganin, Y., & Lempitsky, V. (2015, June). Unsupervised domain adaptation by backpropagation. In International conference on machine learning (pp. 1180-1189). PMLR.

[67] Gao, J., Yang, Z., Chen, K., Sun, C., & Nevatia, R. (2017). Turn tap: Temporal unit regression network for temporal action proposals. In Proceedings of the IEEE international conference on computer vision (pp. 3628-3636).

[68] Gao, X., Hu, W., Tang, J., Liu, J., & Guo, Z. (2019, October). Optimized skeleton-based action recognition via sparsified graph regression. In Proceedings of the 27th ACM international conference on multimedia (pp. 601-610).

[69] Gers, F. A., & Schmidhuber, J. (2000, July). Recurrent nets that time and count. In Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (Vol. 3, pp. 189-194). IEEE.

[70] Ghafoor, M., & Mahmood, A. (2022). Quantification of occlusion handling capability of 3D human pose estimation framework. IEEE Transactions on Multimedia.

[71] Giannakos, I., Mathe, E., Spyrou, E., & Mylonas, P. (2021, June). A study on the Effect of Occlusion in Human Activity Recognition. In Proceedings of the 14th PErvasive Technologies Related to Assistive Environments Conference (pp. 473-482).

[72] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[73] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[74] Gkioxari, G., & Malik, J. (2015). Finding action tubes. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 759-768).

[75] Gonzalez-Sanchez, T., & Puig, D. (2011). Real-time body gesture recognition using depth camera. Electronics Letters, 47(12), 697-698.

[76] Gonzalez, R. C. (2009). Digital image processing. Pearson education india.

[77] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[78] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.

[79] Gowayyed, M. A., Torki, M., Hussein, M. E., & El-Saban, M. (2013). Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition. In Twenty-third international joint conference on artificial intelligence.

[80] Graves, A., Mohamed, A. R., & Hinton, G. (2013, May). Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing (pp. 6645-6649). IEEE.

[81] Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., & Smola, A. (2006). A kernel method for the two-sample-problem. Advances in neural information processing systems, 19.

[82] Gu, Y., Do, H., Ou, Y., & Sheng, W. (2012, December). Human gesture recognition through a kinect sensor. In 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO) (pp. 1379-1384). IEEE.

[83] Gu, R., Wang, G., & Hwang, J. N. (2021, January). Exploring severe occlusion: Multi-person 3d pose estimation with gated convolution. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 8243-8250). IEEE.

[84] Guo, T., Liu, H., Chen, Z., Liu, M., Wang, T.,& Ding, R. (2022, June). Contrastive learning from extremely augmented skeleton sequences for self-supervised action recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 36, No. 1, pp. 762-770).

[85] Hachiya, H., Sugiyama, M., & Ueda, N. (2012). Importance-weighted least-squares probabilistic classifier for covariate shift adaptation with application to human activity recognition. Neurocomputing, 80, 93-101.

[86] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., ... & Tao, D. (2020). A survey on visual transformer. arXiv preprint arXiv:2012.12556.

[87] Haque, M. A., Bautista, R. B., Noroozi, F., Kulkarni, K., Laursen, C. B., Irani, R., ... & Moeslund, T. B. (2018, May). Deep multimodal pain recognition: a database and comparison of spatio-temporal visual modalities. In 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) (pp. 250-257). IEEE.

[88] Hatami, N., Gavet, Y., & Debayle, J. (2018, April). Classification of time-series images using deep convolutional neural networks. In Tenth international conference on machine vision (ICMV 2017) (Vol. 10696, pp. 242-249). SPIE.

[89] Haykin, S. (1998). Neural networks: a comprehensive foundation. Prentice Hall PTR.

[90] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[91] Hernandez, V., Suzuki, T., & Venture, G. (2020). Convolutional and recurrent neural network for human activity recognition: Application on American sign language. PloS one, 15(2), e0228869.

[92] Hinton, G. E., & Zemel, R. (1993). Autoencoders, minimum description length and Helmholtz free energy. Advances in neural information processing systems, 6.

[93] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[94] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the national academy of sciences, 79(8), 2554-2558.

[95] How ReLU and Dropout Layers Work in CNNs. Retrieved from `https://www.baeldung.com/cs/ml-relu-dropout-layers`, 28 April 2024.

[96] Hou, Y., Li, Z., Wang, P., & Li, W. (2016). Skeleton optical spectra-based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology, 28(3), 807-811.

[97] Hou, R., Chen, C., & Shah, M. (2017). Tube convolutional neural network (T-CNN) for action detection in videos. In Proceedings of the IEEE international conference on computer vision (pp. 5822-5831).

[98] Hou, Y., Wang, S., Wang, P., Gao, Z., & Li, W. (2017). Spatially and temporally structured global to local aggregation of dynamic depth information for action recognition. IEEE Access, 6, 2206-2219.

[99] Htike, K. K., Khalifa, O. O., Ramli, H. A. M., & Abushariah, M. A. (2014, April). Human activity recognition for video surveillance using sequences of postures. In e-Technologies and Networks for Development (ICeND), 2014 Third International Conference on (pp. 79-82). IEEE.

[100] Huang, D. A., Fei-Fei, L., & Niebles, J. C. (2016). Connectionist temporal modeling for weakly supervised action labeling. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 137-153). Springer International Publishing.

[101] Huang, Z., Wan, C., Probst, T., & Van Gool, L. (2017). Deep learning on lie groups for skeleton-based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6099-6108).

[102] Hu, J. F., Zheng, W. S., Lai, J., & Zhang, J. (2015). Jointly learning heterogeneous features for RGB-D activity recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5344-5352).

[103] Hu, J., Tuo, H., Wang, C., Qiao, L., Zhong, H., & Jing, Z. (2019). Multi-Weight Partial Domain Adaptation. In BMVC (p. 5).

[104] Huynh-The, T., Hua, C. H., & Kim, D. S. (2019). Encoding pose features to images with data augmentation for 3-D action recognition. IEEE Transactions on Industrial Informatics, 16(5), 3100-3111.

[105] Huynh-The, T., Hua, C. H., Ngo, T. T., & Kim, D. S. (2020). Image representation of pose-transition feature for 3D skeleton-based action recognition. Information Sciences, 513, 112-126.

[106] Ibanez, R., Soria, A., Teyseyre, A., & Campo, M. (2014). Easy gesture recognition for Kinect. Advances in Engineering Software, 76, 171-180.

[107] Ijjina, E. P. (2016). Classification of human actions using pose-based features and stacked auto encoder. Pattern Recognition Letters, 83, 268-277.

[108] Imran, J., & Raman, B. (2020). Evaluating fusion of RGB-D and inertial sensors for multimodal human action recognition. Journal of Ambient Intelligence and Humanized Computing, 11(1), 189-208.

[109] Iosifidis, A., Tefas, A., & Pitas, I. (2012, August). Multi-view human action recognition under occlusion based on fuzzy distances and neural networks. In 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO) (pp. 1129-1133). IEEE.

[110] Jain, A. K. (1989). Fundamentals of digital image processing. Prentice-Hall, Inc..

[111] Jhuang, H., Serre, T., Wolf, L., & Poggio, T. (2007, October). A biologically inspired system for action recognition. In 2007 IEEE 11th international conference on computer vision (pp. 1-8). Ieee.

[112] Ji, S., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. IEEE transactions on pattern analysis and machine intelligence, 35(1), 221-231.

[113] Jiang, F., Zhang, S., Wu, S., Gao, Y., & Zhao, D. (2015). Multi-layered gesture recognition with Kinect. J. Mach. Learn. Res., 16(1), 227-254.

[114] Jiang, W., & Yin, Z. (2015, October). Human activity recognition using wearable sensors by deep convolutional neural networks. In Proceedings of the 23rd ACM international conference on Multimedia (pp. 1307-1310). Acm.

[115] Jones, E., Oliphant, E., Peterson, P. et al.,SciPy: Open Source Scientific Tools for Python. `http://www.scipy.org/` [Accessed 20 May 2019]

[116] Kalogeiton, V., Weinzaepfel, P., Ferrari, V., & Schmid, C. (2017). Action tubelet detector for spatio-temporal action localization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4405-4413).

[117] Kalouris, G., Zacharaki, E. I., & Megalooikonomou, V. (2019, July). Improving CNN-based activity recognition by data augmentation and transfer learning. In 2019 IEEE 17th International Conference on Industrial Informatics (INDIN) (Vol. 1, pp. 1387-1394). IEEE.

[118] Kang, G., Jiang, L., Yang, Y., & Hauptmann, A. G. (2019). Contrastive adaptation network for unsupervised domain adaptation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 4893-4902).

[119] Kao, J. Y., Ortega, A., Tian, D., Mansour, H., & Vetro, A. (2019, September). Graph based skeleton modeling for human activity analysis. In 2019 IEEE International Conference on Image Processing (ICIP) (pp. 2025-2029). IEEE.

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

[121] Ke, Q., An, S., Bennamoun, M., Sohel, F., & Boussaid, F. (2017). Skeletonnet: Mining deep part features for 3-d action recognition. IEEE signal processing letters, 24(6), 731-735.

[122] Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017). A new representation of skeleton sequences for 3d action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3288-3297).

[123] Ke, Q., Bennamoun, M., Rahmani, H., An, S., Sohel, F., & Boussaid, F. (2019). Learning latent global network for skeleton-based action prediction. IEEE Transactions on Image Processing, 29, 959-970.

[124] Keceli, A. S., & Can, A. B. (2014). Recognition of basic human actions using depth information. International Journal of Pattern Recognition and Artificial Intelligence, 28(02), 1450004.

[125] Kendall, A., Grimes, M., & Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In Proceedings of the IEEE international conference on computer vision (pp. 2938-2946).

[126] Keogh, A., Dorn, J. F., Walsh, L., Calvo, F., & Caulfield, B. (2020). Comparing the usability and acceptability of wearable sensors among older Irish adults in a real-world context: observational study. JMIR mHealth and uHealth, 8(4), e15704.

[127] Khan, M. A., Javed, K., Khan, S. A., Saba, T., Habib, U., Khan, J. A., & Abbasi, A. A. (2020). Human action recognition using fusion of multiview and deep features: an application to video surveillance. Multimedia tools and applications, 1-27.

[128] Khan, M. A., Sharif, M., Akram, T., Raza, M., Saba, T., & Rehman, A. (2020). Handcrafted and deep convolutional neural network features fusion and selection strategy: an application to intelligent human action recognition. Applied Soft Computing, 87, 105986.

[129] Kong, Y., & Fu, Y. (2022). Human action recognition and prediction: A survey. International Journal of Computer Vision, 130(5), 1366-1401.

[130] Koniusz, P., Wang, L., & Cherian, A. (2021). Tensor representations for action recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(2), 648-665.

[131] Koniusz, P., Tas, Y., & Porikli, F. (2017). Domain adaptation by mixture of alignments of second-or higher-order scatter tensors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4478-4487).

[132] Koutrintzes, D., Mathe, E., & Spyrou, E. (2022, February). Boosting the Performance of Deep Approaches through Fusion with Handcrafted Features. In ICPRAM (pp. 370-377).

[133] Krizhevsky, A., Sutskever, I.,& Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[134] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[135] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., & Serre, T. (2011, November). HMDB: a large video database for human motion recognition. In 2011 International Conference on Computer Vision (pp. 2556-2563). IEEE.

[136] Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. The annals of mathematical statistics, 22(1), 79-86.

[137] Kwon, T., Tekin, B., Tang, S., & Pollefeys, M. (2022). Context-aware sequence alignment using 4d skeletal augmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8172-8182).

[138] Lai, K., Konrad, J., & Ishwar, P. (2012, April). A gesture-driven computer interface using Kinect. In 2012 IEEE Southwest Symposium on Image Analysis and Interpretation (pp. 185-188). IEEE.

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

[140] Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. IEEE communications surveys & tutorials, 15(3), 1192-1209.

[141] Lawton, M. P., & Brody, E. M. (1969). Assessment of older people: self-maintaining and instrumental activities of daily living. The gerontologist, 9(3 Part 1), 179-186.

[142] Le, Q. V., Zou, W. Y., Yeung, S. Y., & Ng, A. Y. (2011, June). Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In CVPR 2011 (pp. 3361-3368). IEEE.

[143] Lea, C., Flynn, M. D., Vidal, R., Reiter, A., & Hager, G. D. (2017). Temporal convolutional networks for action segmentation and detection. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 156-165).

[144] LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

[145] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[146] LeCun, Y., Touresky, D., Hinton, G., & Sejnowski, T. (1988, June). A theoretical framework for back-propagation. In Proceedings of the 1988 connectionist models summer school (Vol. 1, pp. 21-28).

[147] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.

[148] Lee, M. P., Pryce, J. D., & Harrison, A. (1970). Prolog as a first programming language. WIT Transactions on Information and Communication Technologies, 7.

[149] Lee, I., Kim, D., Kang, S., & Lee, S. (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In Proceedings of the IEEE international conference on computer vision (pp. 1012-1020).

[150] Lev, G., Sadeh, G., Klein, B., & Wolf, L. (2016). Rnn fisher vectors for action recognition and image annotation. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI 14 (pp. 833-850). Springer International Publishing.

[151] Li, B., Dai, Y., Cheng, X., Chen, H., Lin, Y., & He, M. (2017, July). Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN. In 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW) (pp. 601-604). IEEE.

[152] Li, C., Wang, P., Wang, S., Hou, Y., & Li, W. (2017). Skeleton-based action recognition using LSTM and CNN. In Proc. of IEEE ICME Workshops.

[153] Li, C., Hou, Y., Wang, P., & Li, W. (2017). Joint distance maps based action recognition with convolutional neural networks. IEEE Signal Processing Letters, 24(5), 624-628.

[154] Li, C., Zhong, Q., Xie, D., & Pu, S. (2018). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. arXiv preprint arXiv:1804.06055.

[155] Li, S. Z., Yu, B., Wu, W., Su, S. Z., & Ji, R. R. (2015). Feature learning based on SAE–PCA network for human gesture recognition in RGBD images. Neurocomputing, 151, 565-573.

[156] Li, T., Fan, L., Zhao, M., Liu, Y., & Katabi, D. (2019). Making the invisible visible: Action recognition through walls and occlusions. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 872-881).

[157] Li, W., Zhang, Z., & Liu, Z. (2010, June). Action recognition based on a bag of 3d points. In 2010 IEEE computer society conference on computer vision and pattern recognition-workshops (pp. 9-14). IEEE.

[158] Li, W., Wen, L., Chang, M. C., Nam Lim, S., & Lyu, S. (2017). Adaptive RNN tree for large-scale human action recognition. In Proceedings of the IEEE international conference on computer vision (pp. 1444-1452).

[159] Li, X., Zhang, Y., Zhang, J., Chen, S., Marsic, I., Farneth, R. A., & Burd, R. S. (2017). Concurrent activity recognition with multimodal CNN-LSTM structure. arXiv preprint arXiv:1702.01638.

[160] Li, Y., Lan, C., Xing, J., Zeng, W., Yuan, C., & Liu, J. (2016). Online human action detection using joint classification-regression recurrent neural networks. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14 (pp. 203-220). Springer International Publishing.

[161] Li, Y., Miao, Q., Tian, K., Fan, Y., Xu, X., Li, R., & Song, J. (2016, December). Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In 2016 23rd international conference on pattern recognition (ICPR) (pp. 25-30). IEEE.

[162] Li, Z., & Li, D. (2022). Action recognition of construction workers under occlusion. Journal of Building Engineering, 45, 103352.

[163] Liu, C., Hu, Y., Li, Y., Song, S., & Liu, J. (2017, October). PKU-MMD: A large scale benchmark for skeleton-based human action understanding. In Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities (pp. 1-8).

[164] Liu, C. L., Hsaio, W. H., & Tu, Y. C. (2018). Time series classification with multivariate convolutional neural network. IEEE Transactions on industrial electronics, 66(6), 4788-4797.

[165] Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016). Spatio-temporal lstm with trust gates for 3d human action recognition. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14 (pp. 816-833). Springer International Publishing.

[166] Liu, J., Wang, G., Hu, P., Duan, L. Y., & Kot, A. C. (2017). Global context-aware attention lstm networks for 3d action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1647-1656).

[167] Liu, J., Akhtar, N., & Mian, A. (2017, November). Viewpoint invariant RGB-D human action recognition. In 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 1-8). IEEE.

[168] Liu, J., Akhtar, N., & Mian, A. (2019, June). Skepxels: Spatio-temporal image representation of human skeleton joints for action recognition. In CVPR workshops (pp. 10-19).

[169] Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognition, 68, 346-362.

[170] Liu, X., Zheng, Y., Du, Z., Ding, M., Qian, Y., Yang, Z., & Tang, J. (2023). GPT understands, too. AI Open.

[171] Liu, Z., Zhang, C., & Tian, Y. (2016). 3D-based deep convolutional neural network for action recognition with depth sequences. Image and vision computing, 55, 93-100.

[172] Long, M., Zhu, H., Wang, J., & Jordan, M. I. (2017, July). Deep transfer learning with joint adaptation networks. In International conference on machine learning (pp. 2208-2217). PMLR.

[173] Luo, Z., Peng, B., Huang, D. A., Alahi, A., & Fei-Fei, L. (2017). Unsupervised learning of long-term motion dynamics for videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2203-2212).

[174] Ma, S., Sigal, L., & Sclaroff, S. (2016). Learning activity progression in lstms for activity detection and early detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1942-1950).

[175] Mahasseni, B., & Todorovic, S. (2016). Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3054-3062).

[176] Majumder, S., Mondal, T., & Deen, M. J. (2017). Wearable sensors for remote health monitoring. Sensors, 17(1), 130.

[177] Mangera, R., Senekal, F., & Nicolls, F. (2014). Cascading neural networks for upper-body gesture recognition. Avestia Publishing.

[178] Marwan, N. (2008). A historical review of recurrence plots. The European Physical Journal Special Topics, 164(1), 3-12.

[179] Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. arXiv preprint arXiv:1511.05440.

[180] Max Pooling. Retrieved from https://paperswithcode.com/method/max-pooling, 28 April, 2024.

[181] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5, 115-133.

[182] Medsker, L., & Jain, L. C. (Eds.). (1999). Recurrent neural networks: design and applications. CRC press.

[183] Meng, F., Liu, H., Liang, Y., Tu, J., & Liu, M. (2019). Sample fusion network: An end-to-end data augmentation network for skeleton-based human action recognition. IEEE Transactions on Image Processing, 28(11), 5281-5295.

[184] Mikołajczyk, A., & Grochowski, M. (2018, May). Data augmentation for improving deep learning in image classification problem. In 2018 international interdisciplinary PhD workshop (IIPhDW) (pp. 117-122). IEEE.

[185] Miranda, L., Vieira, T., Martinez, D., Lewiner, T., Vieira, A. W., & Campos, M. F. (2012, August). Real-time gesture recognition from depth data through key poses learning and decision forests. In 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images (pp. 268-275). IEEE.

[186] Mitra, S., & Acharya, T. (2007). Gesture recognition: A survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 37(3), 311-324.

[187] Molchanov, P., Gupta, S., Kim, K., & Kautz, J. (2015). Hand gesture recognition with 3D convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-7).

[188] Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., & Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4207-4215).

[189] Motion Capture, retrieved from https://en.wikipedia.org/wiki/Motion_capture, 24 November 2023

[190] Müller, M., Röder, T., Clausen, M., Eberhardt, B., Krüger, B., & Weber, A. (2007). Mocap database hdm05. Institut für Informatik II, Universität Bonn, 2(7).

[191] Neverova, N., Wolf, C., Taylor, G. W., & Nebout, F. (2015). Multi-scale deep learning for gesture detection and localization. In Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part I 13 (pp. 474-490). Springer International Publishing.

[192] Nguyen, T. H., & Grishman, R. (2015, July). Event detection and domain adaptation with convolutional neural networks. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers) (pp. 365-371).

[193] Oberweger, M., Wohlhart, P., & Lepetit, V. (2015). Hands deep in deep learning for hand pose estimation. arXiv preprint arXiv:1502.06807.

[194] Ofli, F., Chaudhry, R., Kurillo, G., Vidal, R., & Bajcsy, R. (2013, January). Berkeley mhad: A comprehensive multimodal human action database. In 2013 IEEE workshop on applications of computer vision (WACV) (pp. 53-60). IEEE.

[195] Oliphant, T. E. (2006). Guide to numpy (Vol. 1, p. 85). USA: Trelgol Publishing.

[196] O'Regan, G., & O'Regan, G. (2013). Marvin Minsky. Giants of Computing: A Compendium of Select, Pivotal Pioneers, 193-195.

[197] Ostwal P. (2023). Data Augmentation for Computer Vision. Retrieved from https://pranjal-ostwal.medium.com/data-augmentation-for-computer-vision-b88b818b6010.

[198] Oyedotun, O. K., & Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. Neural Computing and Applications, 28(12), 3941-3951.

[199] Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on knowledge and data engineering, 22(10), 1345-1359.

[200] Pandolfini, B. (1997). Kasparov and Deep Blue: The historic chess match between man and machine. Simon and Schuster.

[201] Paoletti, G., Cavazza, J., Beyan, C., & Del Bue, A. (2021, January). Subspace clustering for action recognition with covariance representations and temporal pruning. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 6035-6042). IEEE.

[202] Papadakis, A., Mathe, E., Vernikos, I., Maniatis, A., Spyrou, E., & Mylonas, P. (2019). Recognizing human actions using 3d skeletal information and CNNs. In Engineering Applications of Neural Networks: 20th International Conference, EANN 2019, Xersonisos, Crete, Greece, May 24-26, 2019, Proceedings 20 (pp. 511-521). Springer International Publishing.

[203] Papadakis, A., Mathe, E., Spyrou, E., & Mylonas, P. (2019). A geometric approach for cross-view human action recognition using deep learning. In 2019 11th International Symposium on Image and Signal Processing and Analysis (ISPA) (pp. 258-263). IEEE.

[204] Papadakis, A., Vernikos, I., Mathe, E., & Spyrou, E. (2020, June). Skeleton geometric transformation for human action recognition using convolutional neural networks. In Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments (pp. 1-2).

[205] Paraskevopoulos, G., Spyrou, E., & Sgouropoulos, D. (2016, May). A real-time approach for gesture recognition using the Kinect sensor. In Proceedings of the 9th Hellenic Conference on Artificial Intelligence (p. 31). ACM.

[206] Paraskevopoulos, G., Spyrou, E., Sgouropoulos, D., Giannakopoulos, T., & Mylonas, P. (2019). Real-time arm gesture recognition using 3D skeleton joint data. Algorithms, 12(5), 108.

[207] Park, J., Kim, C., & Kim, S. C. (2023). Enhancing Robustness of Viewpoint Changes in 3D Skeleton-Based Human Action Recognition. Mathematics, 11(15), 3280.

[208] Patel, V. M., Gopalan, R., Li, R., & Chellappa, R. (2015). Visual domain adaptation: A survey of recent advances. IEEE signal processing magazine, 32(3), 53-69.

[209] Pazhoumand-Dar, H., Lam, C. P., & Masek, M. (2015). Joint movement similarities for robust 3D action recognition using skeletal data. Journal of Visual Communication and Image Representation, 30, 10-21.

[210] Pei, Z., Cao, Z., Long, M., & Wang, J. (2018, April). Multi-adversarial domain adaptation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[211] Peng, K., Roitberg, A., Yang, K., Zhang, J., & Stiefelhagen, R. (2023). Delving deep into one-shot skeleton-based action recognition with diverse occlusions. IEEE Transactions on Multimedia.

[212] Peng, X., & Schmid, C. (2016). Multi-region two-stream R-CNN for action detection. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14 (pp. 744-759). Springer International Publishing.

[213] Pereira, D. R., Piteri, M. A., Souza, A. N., Papa, J. P., & Adeli, H. (2020). FEMa: A finite element machine for fast learning. Neural Computing and Applications, 32, 6393-6404.

[214] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. arXiv preprint arXiv:1712.04621.

[215] Pham, H. H., Salmane, H., Khoudour, L., Crouzil, A., Zegers, P., & Velastin, S. A. (2019). Spatio–temporal image representation of 3D skeletal movements for view-invariant action recognition with deep convolutional neural networks. Sensors, 19(8), 1932.

[216] Pigou, L., Van Den Oord, A., Dieleman, S., Van Herreweghe, M., & Dambre, J. (2018). Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. International Journal of Computer Vision, 126, 430-439.

[217] Pikramenos, G., Mathe, E., Vali, E., Vernikos, I., Papadakis, A., Spyrou, E., & Mylonas, P. (2020). An adversarial semi-supervised approach for action recognition from pose information. Neural Computing and Applications, 1-15.

[218] Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. IEEE transactions on neural networks and learning systems, 28(12), 3074-3083.

[219] Rahmani, H., & Mian, A. (2016). 3d action recognition from novel viewpoints. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1506-1515).

[220] Rahmani, H., & Bennamoun, M. (2017). Learning action recognition model from depth and skeleton videos. In Proceedings of the IEEE international conference on computer vision (pp. 5832-5841).

[221] Rao, H., Xu, S., Hu, X., Cheng, J., & Hu, B. (2021). Augmented skeleton based contrastive action learning with momentum lstm for unsupervised action recognition. Information Sciences, 569, 90-109.

[222] Ratner, B. (2017). Statistical and machine-learning data mining: Techniques for better predictive modeling and analysis of big data. Chapman and Hall/CRC.

[223] Redko, I., Morvant, E., Habrard, A., Sebban, M., & Bennani, Y. (2019). Advances in Domain Adaptation Theory. Elsevier.

[224] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[225] Reyes, M., Dominguez, G., & Escalera, S. (2011, November). Featureweighting in dynamic timewarping for gesture recognition in depth data. In 2011 IEEE international conference on computer vision workshops (ICCV Workshops) (pp. 1182-1188). IEEE.

[226] Reynolds A. H. (2019) Convolutional Neural Networks (CNNs), Retrieved from `https://anhreynolds.com/blogs/cnn.html`, 28 April, 2024.

[227] Ribó, A., Warcho, D., & Oszust, M. (2016). An approach to gesture recognition with skeletal data using dynamic time warping and nearest neighbour classifier. International Journal of Intelligent Systems and Applications, 8(6), 1.

[228] Sadanand, S., & Corso, J. J. (2012, June). Action bank: A high-level representation of activity in video. In 2012 IEEE Conference on computer vision and pattern recognition (pp. 1234-1241). IEEE.

[229] Saha, S., Singh, G., & Cuzzolin, F. (2017). Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4414-4423).

[230] Sanchez-Riera, J., Hsiao, Y. S., Lim, T., Hua, K. L., & Cheng, W. H. (2014, July). A robust tracking algorithm for 3d hand gesture with rapid hand motion through deep learning. In 2014 IEEE International Conference on Multimedia and Expo Workshops (ICMEW) (pp. 1-6). IEEE.

[231] Salakhutdinov, R., Tenenbaum, J. B.,& Torralba, A. (2012). Learning with hierarchical-deep models. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1958-1971.

[232] Sarker, I. H., Abushark, Y. B., Khan, A. I., Alam, M. M., & Nowrozy, R. (2021). Mobile deep learning: Exploring deep neural network for predicting context-aware smartphone usage. SN Computer Science, 2(3), 146.

[233] Sarker, K., Masoud, M., Belkasim, S., & Ji, S. (2018, December). Towards robust human activity recognition from rgb video stream with limited labeled data. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 145-151). IEEE.

[234] Schuldt, C., Laptev, I., & Caputo, B. (2004, August). Recognizing human actions: a local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. (Vol. 3, pp. 32-36). IEEE.

[235] Shahroudy, A., Liu, J., Ng, T. T., & Wang, G. (2016). Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1010-1019).

[236] Shahroudy, A., Ng, T. T., Gong, Y., & Wang, G. (2017). Deep multimodal feature analysis for action recognition in rgb+ d videos. IEEE transactions on pattern analysis and machine intelligence, 40(5), 1045-1058.

[237] Sharma, S., Kiros, R., & Salakhutdinov, R. (2015). Action recognition using visual attention. arXiv preprint arXiv:1511.04119.

[238] Sharma, P., (2023). Basic Introduction to Convolutional Neural Network in Deep Learning. Retrieved from https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-convolutional-neural-network-in-deep-learning/, 28 April, 2024.

[239] Shen, J., Dudley, J., & Kristensson, P. O. (2021, December). The imaginative generative adversarial network: Automatic data augmentation for dynamic skeleton-based hand gesture and human action recognition. In 2021 16th IEEE international conference on automatic face and gesture recognition (FG 2021) (pp. 1-8). IEEE.

[240] Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. Advances in neural information processing systems, 28.

[241] Shi, Y., Tian, Y., Wang, Y., Zeng, W., & Huang, T. (2017). Learning long-term dependencies for action recognition with a biologically-inspired deep network. In Proceedings of the IEEE International Conference on Computer Vision (pp. 716-725).

[242] Shi, Z., & Kim, T. K. (2017). Learning and refining of privileged information-based RNNs for action recognition from depth sequences. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3461-3470).

[243] Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 60.

[244] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., ... & Blake, A. (2011, June). Real-time human pose recognition in parts from single depth images. In CVPR 2011 (pp. 1297-1304). Ieee.

[245] Shou, Z., Wang, D., & Chang, S. F. (2016). Temporal action localization in untrimmed videos via multi-stage cnns. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1049-1058).

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

[247] Simonyan, K., & Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. Advances in neural information processing systems, 27.

[248] Singh, B., Marks, T. K., Jones, M., Tuzel, O., & Shao, M. (2016). A multi-stream bi-directional recurrent neural network for fine-grained action detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1961-1970).

[249] Singh, G., Saha, S., Sapienza, M., Torr, P. H., & Cuzzolin, F. (2017). Online real-time multiple spatiotemporal action localisation and prediction. In Proceedings of the IEEE international conference on computer vision (pp. 3637-3646).

[250] Singhal, P., Walambe, R., Ramanna, S., & Kotecha, K. (2023). Domain adaptation: challenges, methods, datasets, and applications. IEEE access, 11, 6973-7020.

[251] Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. Statistics and computing, 14, 199-222.

[252] Song, S., Lan, C., Xing, J., Zeng, W., & Liu, J. (2017, February). An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[253] Soo Kim, T., & Reiter, A. (2017). Interpretable 3d human action analysis with temporal convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 20-28).

[254] Soomro, K., Zamir, A. R., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402.

[255] Soomro, K., & Zamir, A. R. (2015). Action recognition in realistic sports videos. In Computer vision in sports (pp. 181-208). Cham: Springer International Publishing.

[256] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15(1), 1929-1958.

[257] Srivastava, N., Mansimov, E., & Salakhudinov, R. (2015, June). Unsupervised learning of video representations using lstms. In International conference on machine learning (pp. 843-852). PMLR.

[258] Steven Eyobu, O., & Han, D. S. (2018). Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network. Sensors, 18(9), 2892.

[259] Stylianou-Nikolaidou, S., Vernikos, I., Mathe, E., Spyrou, E., & Mylonas, P. (2021, June). A novel CNN-LSTM hybrid architecture for the recognition of human activities. In International Conference on Engineering Applications of Neural Networks (pp. 121-132). Cham: Springer International Publishing.

[260] Sun, B., Feng, J., & Saenko, K. (2017). Correlation alignment for unsupervised domain adaptation. Domain adaptation in computer vision applications, 153-171.

[261] Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE international conference on computer vision (pp. 843-852).

[262] Sun, J., Zhou, B., Black, M. J., & Chandrasekaran, A. (2022). LocATe: End-to-end Localization of Actions in 3D with Transformers. arXiv preprint arXiv:2203.10719.

[263] Sun, L., Jia, K., Yeung, D. Y., & Shi, B. E. (2015). Human action recognition using factorized spatio-temporal convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4597-4605).

[264] Sun, L., Jia, K., Chen, K., Yeung, D. Y., Shi, B. E., & Savarese, S. (2017). Lattice long short-term memory for human action recognition. In Proceedings of the IEEE international conference on computer vision (pp. 2147-2156).

[265] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.

[266] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 31, No. 1).

[267] Tang, A., Lu, K., Wang, Y., Huang, J., & Li, H. (2015). A real-time hand posture recognition system using deep neural networks. ACM Transactions on Intelligent Systems and Technology (TIST), 6(2), 1-23.

[268] Tang, Y., Tian, Y., Lu, J., Li, P., & Zhou, J. (2018). Deep progressive reinforcement learning for skeleton-based action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5323-5332).

[269] Tas, Y., & Koniusz, P. (2018). Cnn-based action recognition and supervised domain adaptation on 3d body skeletons via kernel feature maps. arXiv preprint arXiv:1806.09078.

[270] Taylor, G. W., Fergus, R., LeCun, Y., & Bregler, C. (2010). Convolutional learning of spatio-temporal features. In Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI 11 (pp. 140-153). Springer Berlin Heidelberg.

[271] Theis, F. (2006). Towards a general independent subspace analysis. Advances in Neural Information Processing Systems, 19.

[272] Theoharis, T., Papaioannou, G., Platis, N., & Patrikalakis, N. M. (2008). Graphics and visualization: principles & algorithms. CrC Press.

[273] Theodoridis, S. (2015). Machine learning: a Bayesian and optimization perspective. Academic press.

[274] Tran, C., & Trivedi, M. M. (2011). 3-D posture and gesture recognition for interactivity in smart spaces. IEEE Transactions on Industrial Informatics, 8(1), 178-187.

[275] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[276] Tsai, C.J., Tsai, Y.W., Hsu, S.L. and Wu, Y.C., 2017, May. Synthetic Training of Deep CNN for 3D Hand Gesture Identification. In Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO), 2017 International Conference on (pp. 165-170). IEEE.

[277] Tu, J., Liu, H., Meng, F., Liu, M., & Ding, R. (2018, October). Spatial-temporal data augmentation based on LSTM autoencoder network for skeleton-based human action recognition. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 3478-3482). IEEE.

[278] Tzeng, E., Hoffman, J., Saenko, K., & Darrell, T. (2017). Adversarial discriminative domain adaptation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7167-7176).

[279] Uddin, M. A., & Lee, Y. K. (2019). Feature fusion of deep spatial features and handcrafted spatiotemporal features for human action recognition. Sensors, 19(7), 1599.

[280] Van Dyk, D. A., & Meng, X. L. (2001). The art of data augmentation. Journal of Computational and Graphical Statistics, 10(1), 1-50.

[281] Varol, G., Laptev, I., & Schmid, C. (2017). Long-term temporal convolutions for action recognition. IEEE transactions on pattern analysis and machine intelligence, 40(6), 1510-1517.

[282] Vapnik, V. (2013). The nature of statistical learning theory. Springer science & business media.

[283] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[284] Veeriah, V., Zhuang, N.,& Qi, G. J. (2015). Differential recurrent neural networks for action recognition. In Proceedings of the IEEE international conference on computer vision (pp. 4041-4049).

[285] Vernikos, I., Mathe, E., Papadakis, A., Spyrou, E., & Mylonas, Ph. (2019). An Image Representation of Skeletal Data for Action Recognition using Convolutional Neural Networks. In Proc. of PErvasive Technologies Related to Assistive Environments (PETRA) Conference.

[286] Vernikos, I., Mathe, E., Spyrou, E., Mitsou, A., Giannakopoulos, T., & Mylonas, P. (2019, June). Fusing handcrafted and contextual features for human activity recognition. In 2019 14th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP) (pp. 1-6). IEEE.

[287] Vernikos, I., Spyropoulos, T., Spyrou, E., & Mylonas, P. (2023). Human Activity Recognition in the Presence of Occlusion. Sensors, 23(10), 4899.

[288] Vernikos, I., Spyrou, E., Kostis, I. A., Mathe, E., & Mylonas, P. (2023). A deep regression approach for human activity recognition under partial occlusion. International Journal of Neural Systems, 33(09), 2350047.

[289] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P. A., & Bottou, L. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(12).

[290] Wan, J., Zhao, Y., Zhou, S., Guyon, I., Escalera, S., & Li, S. Z. (2016). Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 56-64).

[291] Wang, J., Liu, Z., Wu, Y., & Yuan, J. (2012, June). Mining actionlet ensemble for action recognition with depth cameras. In 2012 IEEE conference on computer vision and pattern recognition (pp. 1290-1297). IEEE.

[292] Wang, J., Nie, X., Xia, Y., Wu, Y., & Zhu, S. C. (2014). Cross-view action modeling, learning and recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2649-2656).

[293] Wang, L., Qiao, Y., & Tang, X. (2015). Action recognition with trajectory-pooled deep-convolutional descriptors. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4305-4314).

[294] Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L. (2016). Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision (pp. 20-36). Springer, Cham.

[295] Wang, L., Xiong, Y., Lin, D., & Van Gool, L. (2017). Untrimmednets for weakly supervised action recognition and detection. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (pp. 4325-4334).

[296] Wang, M., & Deng, W. (2018). Deep visual domain adaptation: A survey. Neurocomputing, 312, 135-153.

[297] Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., & Ogunbona, P. O. (2015). Action recognition from depth maps using deep convolutional neural networks. IEEE Transactions on Human-Machine Systems, 46(4), 498-509.

[298] Wang, P., Li, W., Gao, Z., Tang, C., Zhang, J., & Ogunbona, P. (2015, October). Convnets-based action recognition from depth maps through virtual cameras and pseudocoloring. In Proceedings of the 23rd ACM international conference on Multimedia (pp. 1119-1122).

[299] Wang, P., Li, W., Liu, S., Zhang, Y., Gao, Z., & Ogunbona, P. (2016, December). Large-scale continuous gesture recognition using convolutional neural networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 13-18). IEEE.

[300] Wang, P., Li, Z., Hou, Y., & Li, W. (2016, October). Action recognition based on joint trajectory maps using convolutional neural networks. In Proceedings of the 24th ACM international conference on Multimedia (pp. 102-106).

[301] Wang, P., Li, W., Gao, Z., Zhang, Y., Tang, C., & Ogunbona, P. (2017). Scene flow to action map: A new representation for rgb-d based action recognition with convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 595-604).

[302] Wang, P., Li, W., Ogunbona, P., Wan, J., & Escalera, S. (2018). RGB-D-based human motion recognition with deep learning: A survey. Computer Vision and Image Understanding, 171, 118-139.

[303] Wang, P., Li, W., Li, C., & Hou, Y. (2018). Action recognition based on joint trajectory maps with convolutional neural networks. Knowledge-Based Systems, 158, 43-53.

[304] Wang, P., Li, W., Gao, Z., Tang, C., & Ogunbona, P. O. (2018). Depth pooling based large-scale 3-d action recognition with convolutional neural networks. IEEE Transactions on Multimedia, 20(5), 1051-1061.

[305] Wang, P., Li, W., Wan, J., Ogunbona, P., & Liu, X. (2018, April). Cooperative training of deep aggregation networks for RGB-D action recognition. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[306] Wang, P., Wen, J., Si, C., Qian, Y., & Wang, L. (2022). Contrast-reconstruction representation learning for self-supervised skeleton-based action recognition. IEEE Transactions on Image Processing, 31, 6224-6238.

[307] Wang, Y., Song, J., Wang, L., Van Gool, L., & Hilliges, O. (2016, September). Two-Stream SR-CNNs for Action Recognition in Videos. In Bmvc.

[308] Wei, S., Zou, S., & Liao, F. (2020). A comparison on data augmentation methods based on deep learning for audio classification. In Journal of physics: Conference series (Vol. 1453, No. 1, p. 012085). IOP Publishing.

[309] Weinzaepfel, P., Harchaoui, Z.,& Schmid, C. (2015). Learning to track for spatio-temporal action localization. In Proceedings of the IEEE international conference on computer vision (pp. 3164-3172).

[310] Wu, D., Pigou, L., Kindermans, P. J., Le, N. D. H., Shao, L., Dambre, J., & Odobez, J. M. (2016). Deep dynamic neural networks for multimodal gesture segmentation and recognition. IEEE transactions on pattern analysis and machine intelligence, 38(8), 1583-1597.

[311] Xia, L., Chen, C. C., & Aggarwal, J. K. (2012, June). View invariant human action recognition using histograms of 3d joints. In 2012 IEEE computer society conference on computer vision and pattern recognition workshops (pp. 20-27). IEEE.

[312] Xin, C., Kim, S., Cho, Y., & Park, K. S. (2024). Enhancing Human Action Recognition with 3D Skeleton Data: A Comprehensive Study of Deep Learning and Data Augmentation. Electronics, 13(4), 747.

[313] Xu, T., Zhu, F., Wong, E. K., & Fang, Y. (2016). Dual many-to-one-encoder-based transfer learning for cross-dataset human action recognition. Image and Vision Computing, 55, 127-137.

[314] Yan, X., Chang, H., Shan, S., & Chen, X. (2014). Modeling video dynamics with deep dynencoder. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13 (pp. 215-230). Springer International Publishing.

[315] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., & Brémond, F. (2021, December). Self-supervised video pose representation learning for occlusion-robust action recognition. In 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021) (pp. 1-5). IEEE.

[316] Yang, X., & Tian, Y. (2014). Effective 3d action recognition using eigenjoints. Journal of Visual Communication and Image Representation, 25(1), 2-11.

[317] Yang, Z., Li, Y., Yang, J., & Luo, J. (2018). Action recognition with spatio–temporal visual attention on skeleton image sequences. IEEE Transactions on Circuits and Systems for Video Technology, 29(8), 2405-2415.

[318] Yeung, S., Russakovsky, O., Mori, G., & Fei-Fei, L. (2016). End-to-end learning of action detection from frame glimpses in videos. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2678-2687).

[319] Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., & Fei-Fei, L. (2018). Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision, 126, 375-389.

[320] Yin, Y., & Davis, R. (2014, July). Real-time continuous gesture recognition for natural human-computer interaction. In 2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (pp. 113-120). IEEE.

[321] Yue-Hei Ng, J., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga, R., & Toderici, G. (2015). Beyond short snippets: Deep networks for video classification. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4694-4702).

[322] Yun, K., Honorio, J., Chattopadhyay, D., Berg, T. L., & Samaras, D. (2012, June). Two-person interaction detection using body-pose features and multiple instance learning. In 2012 IEEE computer society conference on computer vision and pattern recognition workshops (pp. 28-35). IEEE.

[323] Zhang, B., Wang, L., Wang, Z., Qiao, Y., & Wang, H. (2016). Real-time action recognition with enhanced motion vector CNNs. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2718-2726).

[324] Zhang, J., Han, Y., Tang, J., Hu, Q., & Jiang, J. (2016). Semi-supervised image-to-video adaptation for video action recognition. IEEE transactions on cybernetics, 47(4), 960-973.

[325] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In Proceedings of the IEEE international conference on computer vision (pp. 2117-2126).

[326] Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In Proceedings of the IEEE International Conference on Computer Vision (pp. 2117-2126).

[327] Zhang, P., Xue, J., Lan, C., Zeng, W., Gao, Z., & Zheng, N. (2019). EleAtt-RNN: Adding attentiveness to neurons in recurrent neural networks. IEEE Transactions on Image Processing, 29, 1061-1073.

[328] Zhang, P., Lan, C., Zeng, W., Xing, J., Xue, J., & Zheng, N. (2020). Semantics-guided neural networks for efficient skeleton-based human action recognition. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1112-1121).

[329] Zhang, S., Liu, X., & Xiao, J. (2017, March). On geometric features for skeleton-based action recognition using multilayer lstm networks. In 2017 IEEE winter conference on applications of computer vision (WACV) (pp. 148-157). IEEE.

[330] Zhang, S., Yang, Y., Xiao, J., Liu, X., Yang, Y., Xie, D., & Zhuang, Y. (2018). Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks. IEEE Transactions on Multimedia, 20(9), 2330-2343.

[331] Zhang, X., Chen, X., Wang, W. H., Yang, J. H., Lantz, V., & Wang, K. Q. (2009, February). Hand gesture recognition and virtual game control based on 3D accelerometer and EMG sensors. In Proceedings of the 14th international conference on Intelligent user interfaces (pp. 401-406).

[332] Zhang, X., Yu, F. X., Chang, S. F., & Wang, S. (2015). Deep transfer network: Unsupervised domain adaptation. arXiv preprint arXiv:1503.00591.

[333] Zhang, Z. (2012). Microsoft kinect sensor and its effect. IEEE multimedia, 19(2), 4-10.

[334] Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., & Lin, D. (2017). Temporal action detection with structured segment networks. In Proceedings of the IEEE international conference on computer vision (pp. 2914-2923).

[335] Zhu, G., Zhang, L., Mei, L., Shao, J., Song, J., & Shen, P. (2016, December). Large-scale isolated gesture recognition using pyramidal 3d convolutional networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 19-24). IEEE.

[336] Zhu, G., Zhang, L., Shen, P., & Song, J. (2017). Multimodal gesture recognition using 3-D convolution and convolutional LSTM. Ieee Access, 5, 4517-4524.

[337] Zhu, H., Vial, R., & Lu, S. (2017). Tornado: A spatio-temporal convolutional regression network for video action proposal. In Proceedings of the IEEE International Conference on Computer Vision (pp. 5813-5821).

[338] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., & Xie, X. (2016, March). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 30, No. 1).