



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ**

ΠΡΟΓΡΑΜΜΑ ΔΙΔΑΚΤΟΡΙΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αναγνώριση Ανθρώπινης Δραστηριότητας Υπό Παρεμπόδιση

Ιωάννης Μ. Βερνίκος

ΛΑΜΙΑ

ΔΕΚΕΜΒΡΙΟΣ 2023

Η υλοποίηση της διδακτορικής διατριβής συγχρηματοδοτήθηκε από την Ελλάδα και την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) μέσω του Επιχειρησιακού Προγράμματος «Ανάπτυξη Ανθρώπινου Δυναμικού, Εκπαίδευση και Διά Βίου Μάθηση», 2014-2020, στο πλαίσιο της Πράξης «Ενίσχυση του ανθρώπινου δυναμικού μέσω της υλοποίησης διδακτορικής έρευνας Υποδράση 2: Πρόγραμμα χορήγησης υποτροφιών ΙΚΥ σε υποψηφίους διδάκτορες των ΑΕΙ της Ελλάδας



**Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση**
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





UNIVERSITY OF THESSALY

**SCHOOL OF SCIENCES
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS**

DOCTORAL STUDIES PROGRAM

PhD THESIS

Human Activity Recognition Under Occlusion

Ioannis M. Vernikos

LAMIA

DECEMBER 2023

The implementation of the doctoral thesis was co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the Act “Enhancing Human Resources Research Potential by undertaking a Doctoral Research” Sub-action 2: IKY Scholarship Programme for PhD candidates in the Greek Universities.



**Operational Programme
Human Resources Development,
Education and Lifelong Learning**
Co-financed by Greece and the European Union



ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

Αναγνώριση Ανθρώπινης Δραστηριότητας Υπό Παρεμπόδιση

Ιωάννης Μ. Βερνίκος

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: **Ευάγγελος Σπύρου**, Αναπληρωτής Καθηγητής Πανεπιστήμιο Θεσσαλίας

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ:

Ευάγγελος Σπύρου, Αναπληρωτής Καθηγητής, Πανεπιστήμιο Θεσσαλίας

Σταύρος Περαντώνης, Ερευνητής Α, Ε.ΚΕ.Φ.Ε-Δημόκριτος

Φοίβος-Απόστολος Μυλωνάς, Αναπληρωτής Καθηγητής, Πανεπιστήμιο Δυτικής Αττικής

ΕΠΤΑΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ

**Ευάγγελος Σπύρου,
Αναπληρωτής Καθηγητής, Πανεπιστήμιο
Θεσσαλίας**

**Σταύρος Περαντώνης,
Ερευνητής Α, Ε.ΚΕ.Φ.Ε.-
Δημόκριτος**

**Φοίβος-Απόστολος Μυλωνάς,
Αναπληρωτής Καθηγητής, Πανεπιστήμιο
Δυτικής Αττικής**

**Γεώργιος Καρράς,
Αναπληρωτής Καθηγητής,
Πανεπιστήμιο Θεσσαλίας**

**Σταύρος Καρκάνης,
Καθηγητής, Πανεπιστήμιο Θεσσαλίας**

**Μιχαήλ Σαβελώνας,
Επίκουρος Καθηγητής,
Πανεπιστήμιο Θεσσαλίας**

**Σπυρίδων Γεωργακόπουλος ,
Επίκουρος Καθηγητής, Πανεπιστήμιο
Θεσσαλίας**

Ημερομηνία Εξέτασης: 14 ΔΕΚΕΜΒΡΙΟΥ 2023

PhD THESIS

Human Activity Recognition Under Occlusion

Ioannis M. Vernikos

SUPERVISOR: **Evangelos Spyrou**, Associate Professor University Of Thessaly

THREE-MEMBER ADVISORY COMMITTEE:

Evangelos Spyrou, Associate Professor, University Of Thessaly

Stravros Perantonis, Researcher A, NCSR-"Demokritos"

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

SEVEN-MEMBER EXAMINATION COMMITTEE

Evaggelos Spyrou,
Associate Professor, University Of Thessaly

Stravros Perantonis,
Researcher A, NCSR-
"Demokritos"

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

Georgios Karras,
Associate Professor, University Of Thessaly

Stavros Karkanis,
Professor, University Of Thessaly

Michail Savelonas,
Assistant Professor, University Of Thessaly

Spyridon Georgakopoulos,
Assistant Professor, University Of Thessaly

Examination Date: DECEMBER 14, 2023

ΠΕΡΙΛΗΨΗ

Η αναγνώριση ανθρώπινης δραστηριότητας ασχολείται με την αναγνώριση της ανθρώπινης συμπεριφοράς με βάση μετρήσεις και παρατηρήσεις. Πρόκειται για ένα ερευνητικό πεδίο που εντάσσεται στο ευρύτερο επιστημονικό πεδίο της υπολογιστικής όρασης και τυγχάνει σημαντικής προσοχής από την ερευνητική κοινότητα. Οι προσεγγίσεις αναγνώρισης ανθρώπινης συμπεριφοράς μπορούν να χρησιμοποιήσουν διάφορους τύπους αισθητήρων, για να καταγράψουν και να αναλύσουν τις σχετικές δραστηριότητες. Ωστόσο, οι φορητοί αισθητήρες δεν προτιμώνται από τους χρήστες, ενώ οι αισθητήρες που τοποθετούνται στο περιβάλλον μπορεί να είναι δαπανηροί ή να απαιτούν παρεμβάσεις στο χώρο τους. Ως εκ τούτου, χρησιμοποιούνται συνήθως λύσεις χαμηλού κόστους που βασίζονται στην καταγραφή της κίνησης με κάμερες σε τρισδιάστατο χώρο. Η αναγνώριση ανθρώπινης δραστηριότητας έχει εφαρμογές στη επιτήρηση με χρήση βίντεο, την αλληλεπίδραση ανθρώπου-υπολογιστή/ρομπότ, την επαυξημένη πραγματικότητα, την παρακολούθηση της υγείας, την έξυπνη οδήγηση, τα παιχνίδια, τα κινούμενα σχέδια και άλλα. Ως δραστηριότητες θεωρούνται πλήρεις ανθρώπινες κινήσεις ή ακολουθίες ενεργειών που σχηματίζουν μια ουσιαστική ανθρώπινη δράση, ενώ η αναγνώρισή τους από δεδομένα κίνησης θεωρείται ένα από τα πιο δύσκολα προβλήματα υπολογιστικής όρασης, που περιλαμβάνει την αναγνώριση της ανθρώπινης συμπεριφοράς σε εικόνες ή ακολουθίες βίντεο.

Οι προσεγγίσεις αναγνώρισης συμπεριφοράς που βασίζονται σε κάμερες είναι χαμηλού κόστους και αποδίδουν καλά σε εργαστηριακές συνθήκες. Ωστόσο, αντιμετωπίζουν προκλήσεις σε πραγματικές συνθήκες, συμπεριλαμβανομένων των αλλαγών οπτικής γωνίας και φωτισμού, καθώς και της παρεμπόδισης. Η μεταβολή του σημείου παρατήρησης συμβαίνει όταν το θέμα παρατηρείται από διαφορετικό σημείο από αυτό που χρησιμοποιήθηκε για την εκπαίδευση. Οι πολλαπλές κάμερες μπορούν να αντισταθμίσουν αυτό το πρόβλημα. Οι αλλαγές φωτισμού επηρεάζουν τις προσεγγίσεις που βασίζονται σε βίντεο, αλλά οι πληροφορίες βάθους μπορούν να βοηθήσουν να ξεπεραστεί αυτή η πρόκληση. Η παρεμπόδιση, που προκαλείται από έπιπλα ή την παρουσία άλλων ανθρώπων, είναι το πιο περιοριστικό πρόβλημα. Η πλήρης παρεμπόδιση καθιστά αδύνατη την αναγνώριση, αλλά η μερική παρεμπόδιση μπορεί να επιτρέψει την αναγνώριση ανάλογα με το ορατό υποσύνολο του σκελετού. Η επίδραση της μερικής παρεμπόδισης στην ακρίβεια της αναγνώρισης έχει μελετηθεί, δείχνοντας ότι μπορεί να επηρεάσει σημαντικά την απόδοση με την έκταση να ποικίλλει ανά δραστηριότητα.

Η παρούσα διατριβή αρχίζει με την παρουσίαση των συνόλων δεδομένων που θα χρησιμοποιηθούν για την πειραματική αξιολόγηση των προτεινόμενων προσεγγίσεων. Παρουσιάζονται οι οπτικοί αισθητήρες που χρησιμοποιούνται συνήθως στις σύγχρονες προσεγγίσεις υπολογιστικής όρασης για την αναγνώριση της δραστηριότητας, καθώς και διάφορες βιβλιοθήκες λογισμικού υπολογιστικής όρασης που μπορούν να συμπληρώσουν αυτούς τους αισθητήρες. Παρέχεται επίσης μια επισκόπηση των ερευνητικών προσπαθειών στην ευρύτερη ερευνητική περιοχή, με χρήση οπτικών δεδομένων. Στη συνέχεια, παρουσιάζ-

ζεται το θεωρητικό και πειραματικό υπόβαθρο για τις μεθοδολογίες που θα συζητηθούν στην παρούσα διατριβή. Συγκεκριμένα, περιγράφονται οι κυριότεροι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιούνται, συμπεριλαμβανομένων των νευρωνικών δικτύων συνελκτικής μάθησης, των αναδρομικών νευρωνικών δικτύων και των γεννητικών ανταγωνιστικών δικτύων. Επιπλέον, περιγράφονται οι γλώσσες προγραμματισμού και οι βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίηση, παρέχεται λεπτομερή εξήγηση των τρισδιάστατων σκελετικών δεδομένων που χρησιμοποιήθηκαν ως δεδομένα εισόδου για τις προτεινόμενες προσεγγίσεις και παρουσιάζονται τα πρωτόκολλα πειραματικής αξιολόγησης που ευθυγραμμίζονται με την υπάρχουσα βιβλιογραφία.

Στη συνέχεια, ακολουθεί η παρουσίαση της πραγματοποιηθείσας έρευνας, η οποία ξεκινά με μια ψευδο-χρωματική αναπαράσταση εικόνας που έχει ως στόχο να συλλάβει τα χαρακτηριστικά της τρισδιάστατης σκελετικής κίνησης κατά τη διάρκεια μιας δραστηριότητας μέσα σε μοναδική εικόνα. Έπειτα, προτείνεται μια προσέγγιση σύντηξης που συνδυάζει χειροποίητα και χαρακτηριστικά βαθιάς μάθησης. Ο στόχος αυτής της προσέγγισης είναι να αξιολογηθεί κατά πόσον τα χαρακτηριστικά που εξάγονται από ένα συνελκτικό νευρωνικό δίκτυο που εκπαιδεύεται με ψευδοχρωματικές αναπαραστάσεις της τρισδιάστατης σκελετικής κίνησης μπορούν να βελτιώσουν την απόδοση των παραδοσιακών προσεγγίσεων. Με άλλα λόγια, διερευνάται κατά πόσον τα χαρακτηριστικά αυτά μπορούν να παρέχουν πρόσθετες ενδείξεις για την αναγνώριση δραστηριοτήτων. Επιπλέον, προτείνεται μια προσέγγιση για την αντιμετώπιση του προβλήματος της απόκρυψης σε προβλήματα αναγνώρισης συμπεριφοράς. Η προσέγγιση αυτή περιλαμβάνει την επαύξηση του συνόλου δεδομένων εκπαίδευσης με δείγματα τόσο με όσο και χωρίς παρεμπόδιση. Στη συνέχεια γίνεται χρήση ενός βαθιού δικτύου το οποίο υλοποιεί παλινδρόμηση, με σκοπό να ανακατασκευάσει αρθρώσεις του σκελετού που δεν είναι ορατές λόγω της παρεμπόδισης. Τέλος διερευνάται η χρήση ενός γεννητικού ανταγωνιστικού δικτύου για την ανακατασκευή αρθρώσεων. Και στις δύο προσεγγίσεις ανακατασκευής, η ταξινόμηση πραγματοποιείται με τη χρήση ενός δικτύου που εκπαιδεύεται αποκλειστικά με πλήρη δείγματα. Η διατριβή κλείνει με την παρουσίαση εκτενών συμπερασμάτων σχετικά με όλες τις προσεγγίσεις που έχουν παρουσιαστεί και συζητώνται πιθανές ερευνητικές κατευθύνσεις προς τη συνέχιση των προτεινόμενων μεθολογιών.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αναγνώριση ανθρώπινης δραστηριότητας

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Μηχανική Μάθηση, Βαθιά Μάθηση, ΠΑΔ, Παλινδρόμηση

ABSTRACT

Human activity recognition (HAR) is the task of identifying human behavior based on measurements and observations. It is a challenging computer vision task that has gained significant attention in the research community. HAR approaches can utilize various types of sensors, to capture and analyze human activities. However, wearable sensors are not preferred by users, and sensors placed in the environment can be costly or require interventions. Therefore, low-cost solutions based on camera-based motion capture in 3D space are commonly used. HAR has applications in video surveillance, human-computer/robot interaction, augmented reality, health monitoring, intelligent driving, gaming, animation, and more. Actions and activities are considered complete human motions or sequences of actions forming a meaningful human action, although these terms are mostly used interchangeably, while their recognition from motion data is considered one of the most challenging computer vision problems, involving the recognition of human behavior in images or video sequences.

Camera-based HAR approaches are low-cost and perform well in laboratory conditions. However, they face challenges in real-life conditions, including viewpoint and illumination changes, as well as occlusion. Viewpoint variance occurs when the subject is observed from a different viewpoint than during training. Multiple cameras can compensate for this. Illumination changes affect video-based approaches, but depth information can help overcome this challenge. Occlusion, caused by furniture or the presence of other humans, is the most limiting problem. Full occlusion makes HAR impossible, but partial occlusion can still allow recognition depending on the visible skeleton subset. The effect of partial occlusion on HAR accuracy has been studied, showing that it can significantly impact recognition performance, with the extent varying for each activity.

This Thesis begins by introducing the datasets that will be used to experimentally evaluate the proposed approaches. We discuss the primary visual sensors commonly used in contemporary computer vision approaches for Human Activity Recognition (HAR), as well as various computer vision libraries that can complement these sensors. We also provide an overview of the research efforts in the broader field of HAR using visual data. Next, we present the theoretical and experimental background for the methodologies that will be discussed in this Thesis. Specifically, we outline the main machine learning algorithms employed, including Convolutional Neural Networks, Recurrent Neural Networks, and Generative Adversarial Networks. Additionally, we describe the programming languages and libraries utilized for implementation, we provide a detailed explanation of the 3D skeletal data used as input for our approaches, and we present the experimental evaluation protocols that align with the existing literature.

Subsequently, we commence the presentation of our research with a pseudo-colored image representation for HAR. This representation aims to capture the characteristics of 3D skeletal motion during an activity within a single image. We propose a fusion approach that

combines handcrafted and contextual features for HAR. The objective of this approach is to assess whether features extracted from a Convolutional Neural Network trained with pseudo-colored representations of 3D skeletal motion can enhance the performance of traditional approaches. In other words, we investigate whether these features can provide additional cues for activity recognition. Furthermore, we propose an approach to address the issue of occlusion in HAR problems. This approach involves augmenting the training dataset with both occluded and non-occluded samples. We then attempt to overcome the occlusion problem using a deep regression network, which aims to reconstruct missing skeletal joints. Additionally, we explore the use of a Generative Adversarial Network for joint reconstruction. In both reconstruction approaches, classification is performed using a network trained exclusively on non-occluded samples. Finally, we present extensive conclusions regarding all approaches that are presented and we discuss possible research directions towards the continuation of this work.

SUBJECT AREA: Human Activity Recognition

KEYWORDS: machine Learning, deep learning, gan, regression

This work is dedicated to my beloved ones who passed away while I was writing this thesis.

ACKNOWLEDGEMENTS

The implementation of the doctoral thesis was co-financed by Greece and the European Union (European Social Fund-ESF) through the Operational Programme «Human Resources Development, Education and Lifelong Learning» in the context of the Act “Enhancing Human Resources Research Potential by undertaking a Doctoral Research” Sub-action 2: IKY Scholarship Programme for PhD candidates in the Greek Universities.



Operational Programme
Human Resources Development,
Education and Lifelong Learning
Co-financed by Greece and the European Union



CONTENTS

1	HUMAN ACTIVITY RECOGNITION	33
1.1	Human Activity Recognition from Visual Data	33
1.2	Applications of Human Activity Recognition	34
1.3	Research Questions/Hypotheses	35
1.4	Thesis Structure	36
1.5	Datasets	37
1.5.1	The PKU-MMD dataset	37
1.5.2	The NTU-RGB+D dataset	39
1.5.3	The SYSU 3D Human-Object Interaction dataset	42
1.5.4	The UTKinect-Action3D dataset	43
1.5.5	Other HAR datasets	44
1.6	Video Sensors used in HAR	44
1.6.1	The Microsoft Kinect v1	44
1.6.2	The Microsoft Kinect v2	46
1.6.3	The Intel RealSense camera	47
1.6.4	The Kinect Azure camera	48
1.6.5	Other sensors	49
1.7	Computer Vision Libraries for Skeletal Motion Extraction	50
1.7.1	OpenPose	50
1.7.2	PoseNet	51
1.8	Contemporary Research Efforts on HAR	52
1.8.1	Data Input	53
1.8.2	Feature Extraction	55
1.8.3	Descriptor Formation	57
1.8.4	Classification	58
2	THEORETICAL AND EXPERIMENTAL BACKGROUND	61
2.1	Artificial Intelligence	61

2.2	Machine Learning	62
2.3	Deep Learning	63
2.4	Convolutional Neural Networks	63
2.4.1	Convolutional Layers	64
2.4.2	Pooling Layers	67
2.4.3	Normalization Layers	68
2.4.4	Fully-Connected (Dense) Layers	68
2.4.5	Classification with CNNs	68
2.5	Recurrent Neural Networks	69
2.6	Generative Adversarial Networks	71
2.7	Implementation	73
2.7.1	Programming Languages, Frameworks and Libraries	74
2.7.2	Implementation Details	75
2.8	Skeletal Data	76
2.9	Experimental Protocols	77
2.10	Performance Metrics	77
3	PSEUDO-COLORED IMAGE REPRESENTATION FOR HAR	81
3.1	Introduction	81
3.2	Related work	81
3.3	Methodology	83
3.4	Experimental Results	85
3.5	Conclusions	87
4	FUSION OF HANDCRAFTED AND CONTEXTUAL FEATURES FOR HAR	93
4.1	Introduction	93
4.2	Related work	93
4.3	Proposed Methodology	94
4.3.1	Handcrafted Features	94
4.3.2	Context-aware deep Feature Extraction using a CNN	94
4.3.3	HAR fusing Contextual and Handcrafted Features	96
4.4	Experimental Results	97

4.4.1	Gesture recognition dataset	97
4.4.2	Model Training	97
4.4.3	Results	98
4.5	Conclusions	98
5	A DATA AUGMENTATION APPROACH FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION	101
5.1	Introduction	101
5.2	Related work	101
5.3	Methodology	103
5.3.1	Image Representation of Skeletal Data	103
5.3.2	Occlusion of Skeletal Data	104
5.3.3	Classification	105
5.4	Experiments and Results	106
5.4.1	Evaluation Protocol	107
5.4.2	Results and Discussion	108
5.5	Conclusions	113
6	A DEEP REGRESSION APPROACH FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION	115
6.1	Introduction	115
6.2	Methodology	115
6.2.1	Skeletal Data	115
6.2.2	Occlusion	115
6.2.3	Regression of Skeletal Data	117
6.3	Network Training	120
6.4	Evaluation Protocol and Results	120
6.5	Conclusions	123
7	USING GANS FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION	129
7.1	Introduction	129
7.2	Related work	129
7.3	Methodology	131

7.3.1	Skeletal Data and Occlusion	131
7.3.2	Occlusion	131
7.3.3	Generative Adversarial Networks	131
7.3.3.1	Generator	132
7.3.3.2	Discriminator	132
7.3.4	Classification	132
7.3.5	The GAN objective	133
7.3.6	Network Training and Evaluation	134
7.3.7	Evaluation Protocol and Results	135
7.4	Conclusions	140
8	CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS	147
	ABBREVIATIONS - ACRONYMS	151
	REFERENCES	173
	APPENDICES	173
A	SHORT CV, DECEMBER 2023	175
B	LIST OF RELEVANT PUBLICATIONS	177
B.1	Publications in international scientific journals	177
B.2	Publications in international scientific conferences	177

LIST OF FIGURES

1.1	Examples of frames of the PKU-MMD dataset demonstrate the diversity of human subjects and camera angles. Figure taken from [125].	38
1.2	Example of activities of the PKU-MMD Dataset. From left to right, some of the activities are: first row: wear a hat, handshaking, fighting and combing my hair second row: eating , third row. Figure taken from [125].	38
1.3	Examples of frames of the NTU RGB+D dataset and the NTU RGB+D+120 demonstrate the diversity of human subjects and camera angles. Figure taken from [172].	40
1.4	Available data modalities in the NTU-RGB+D and the NTU-RGB+D+120 dataset. From left to right: RGB data, RGB data with imposed 3D skeletal data, depth data, depth data with imposed 3D skeletal data, infrared data. Figure taken from [125].	40
1.5	Examples of random activities and topics from the SYSU 3D HOI dataset in RGB and depth modes are displayed. Figure taken from [68].	42
1.6	Examples of random activities and topics from the UTKinect dataset in RGB and depth modes are displayed. Figure taken from [208].	43
1.7	A Microsoft Kinect v1 device.	45
1.8	A Microsoft Kinect v2 device.	46
1.9	An Intel RealSense camera.	47
1.10	The Kinect Azure camera.	48
1.11	Example skeletons extracted by OpenPose. Figure taken from [25].	51
1.12	Example skeletons extracted by OpenPose. Figure taken from [88].	52
1.13	Example of multi-view scenery which shows a person waving his left hand, observed from four different viewpoints. Figure taken from [158].	54
1.14	Examples of multi-view (side and frontal views) setup. Figure taken from [121].	55
1.15	Examples of sports action with human-to-object interactions. Figure taken from [67].	55
2.1	A Venn diagram illustrating the relation of machine learning (ML) and deep learning (DL) to artificial intelligence (AI). Specifically, ML is a subset of AI, while DL in turn is a subset of ML.	61
2.2	An Artificial neural network architecture. Figure taken from [19]	63

2.3	Top image: Consider reading a book written in English: to read a specific page, one begins from the top-left corner, moving to the right and proceeding to the next row upon reaching the end of a line, repeating this process until the bottom-right corner is reached. At this point, all words of the page have been read. Bottom image: Similarly to the aforementioned process, a kernel in a convolutional layer scans a given image from top to bottom and from left to right. Figure taken from [96].	65
2.4	Left: a 3×3 kernel; Right: a 3×3 pixel window. Figure taken from [96]. . .	65
2.5	A visual example showing how the activation values in a feature map are calculated in a convolution layer. Figure taken from [96].	67
2.6	A max-pooling layer is demonstrated here, with a 4×4 activation map. It works similarly to a convolutional layer, moving from left to right and top to bottom over the matrix of values. A 2×2 filter is used, which pertains only the largest of the 4 input values (e.g., the “orange 5” in the top-left corner of the 2×2 hatch-marked area). With a 2×2 stride, the output of this max-pooling layer has one-fourth of the volume of its input, resulting in a 2×2 activation map. Figure taken from [96].	68
2.7	Batch normalization modifies the distribution of the activations produced by a layer of neurons to be closer to a standard normal distribution. Figure taken from [96]	69
2.8	A visual example of how a CNN works. Firstly, A given input image of a traffic sign is filtered by 45×5 convolutional kernels. This way 4 feature maps are created and are then subsampled by the process of max pooling. The next layer applies 105×5 convolutional kernels to these subsampled images and the features maps are again subsampled by max pooling. The final layer is a fully connected layer where all generated features are combined and used for classification. Figure taken from [143].	70
2.9	Another visual example of how a CNN works. The network’s 8 layers alternate between convolutional (C) and max-pooling (MP). The first layer (C) filters the input 64×64 px image with a bank of 10×10 filters producing 10 maps of size 63×63 , that are down-sampled to 21×21 px by the second (MP) layer. 2. The procedure continues this way until the input image is transformed to just 3 features that are used for classification. Figure taken from [94].	70
2.10	Visual explanation of the way a discriminator of a GAN works. Figure taken from [96].	73
2.11	Visual explanation of the way a generator of a GAN works. Figure taken from [96].	74

2.12	A human body pose with the 20 and 25 skeletal joints that are extracted using the Microsoft Kinect v1 (left) and v2 (right) cameras. Joints have been divided into subsets, each corresponding to one of the five main body parts, i.e., torso (blue), left hand (green), right hand (red), left leg (orange) and right leg (magenta). For illustrative purposes and also to facilitate comparisons between the two different versions, body parts have been colored using the same colors. Numbering follows the Kinect SDK in both cases, therefore there exist several differences between the two versions.	76
2.13	A sequence of an actor performing the activity <i>wearing a hat</i> . Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the NTU dataset [172] and have been trimmed for illustration purposes.	76
2.14	A sequence of an actor performing the activity <i>salute</i> . Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the PKU-MMD dataset [125] and have been trimmed for illustration purposes.	77
3.1	Representative pseudo-colored images of the eleven classes of the PKU-MMD Dataset.	84
3.2	The architecture of the CNN that has been used for classification of pseudo-colored images.	85
3.3	Normalized confusion matrices for classification of the medical classes of the NTU-RGB+D dataset. M, L and R denote middle, left and right camera angles, respectively.	89
3.4	Normalized confusion matrices for classification of the ADL classes of the PKU-MMD dataset. M, L and R denote middle, left and right camera angles, respectively.	90
3.5	Normalized confusion matrices for the classification of the (a) UTKinect-Action3D and the (b) SYSU-3D 3D Human-Object Interaction Dataset. . . .	91
4.1	Representative pseudo-colored images from actions of the PKU-MMD dataset [125] and by using all 25 skeletal joints. Top: <i>eat_meal_snack</i> , bottom: <i>falling</i>	96
4.2	A visual overview of the approach for classification of activities based on the fusion of contextual and handcrafted features.	97
5.1	Pseudocoloured images for the PKU-MMD [125] and the NTU [172] datasets. LA, RA, LL and RL denote the cases of occlusion of left arm, right arm, left leg and right leg.	104
5.2	Example skeleton sequences of the activity <i>salute</i> from the PKU-MMD dataset [125]. First row: skeletons include all 25 joints; Second row: joints corresponding to left arm have been discarded.	106

5.4	The convolutional neural network that has been used for the experimental evaluation of the methodology presented in this Chapter.	106
5.3	Example skeleton sequences of the activity <i>wearing a hat</i> from the NTU dataset [172]. First row: skeletons include all 25 joints; Second row: joints corresponding to left arm have been discarded.	107
6.1	Example skeleton sequences of the activities (a) <i>handshaking</i> and (b) <i>hugging other person</i> 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 (a) left arm; (b) both arms (see Fig. 2.12) have been discarded (i.e., the skeleton is partially occluded); Third row: skeletons have been reconstructed using the proposed deep regression approach. The example of (a) is successfully reconstructed and correctly classified, while the example of (b) is unsuccessfully reconstructed and incorrectly classified.	116
6.2	The architecture of the CRNN, for the three-camera case.	118
6.3	The architecture of the LSTM network, for the three-camera case.	119
6.4	A visual overview of the proposed approach.	120
6.5	Normalized confusion matrices for classification for all datasets, without removing any body part.	123
6.6	Normalized confusion matrices for classification for the PKU-MMD dataset, upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	125
6.7	Normalized confusion matrices for classification for the NTU-RGB+D dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	126
6.8	Average confusion matrices from five trials, for classification for the SYSU-3D-UOI dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	127
6.9	Average confusion matrices from five trials, for classification for the UTKinect-Action-3D dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	128
7.1	A visual overview of the proposed approach.	134
7.2	The architecture of the Classifier of the proposed GAN architecture., for the one-camera case.	135

7.3	The architecture of the Classifier of the proposed GAN architecture., for the three-camera case.	136
7.4	The architecture of the Discriminator of the proposed GAN architecture. . .	137
7.5	The architecture of the Generator of the proposed GAN architecture.	138
7.6	Normalized confusion matrices for classification for the NTU-RGB+D dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	139
7.7	Normalized confusion matrices for classification for the UTKinect-Action3D dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	140
7.8	Normalized confusion matrices for classification for the PKU-MMD dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	141
7.9	Normalized confusion matrices for classification for the SYSU-3D-HOI dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.	141
7.10	Example skeleton sequences of the activities (a) <i>handshaking</i> and (b) <i>hugging other person</i> 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 (a) left arm; (b) both arms (see Fig. 2.12) have been discarded (i.e., the skeleton is partially occluded); Third row: skeletons have been reconstructed using the proposed deep regression approach. The example of (a) is successfully reconstructed and correctly classified, while the example of (b) is unsuccessfully reconstructed and incorrectly classified.	142

LIST OF TABLES

1.1	The NTU RGB+D dataset includes 16 different heights and distances of setup positions. Additionally, the NTU RGB+D+120 dataset is an expansion of this dataset, which includes additional 16 setup positions.	41
2.1	Definition of a confusion matrix for a binary classification problem.	78
3.1	Results of the evaluation of the proposed system, when tested on 11 classes of the PKU-MMD dataset, using the middle, left and right camera angles (M, L and R, respectively). Best result denoted in bold.	86
3.2	Results of the evaluation of the proposed system, when tested on the PKU-MMD dataset, using the middle, left and right camera angles (M, L and R). Best result denoted in bold.	87
3.3	Results of the evaluation of the proposed system, when tested on eleven classes of the NTU-Medical dataset, using the middle, left and right camera angles (M, L and R).	88
3.4	Results of the evaluation of the proposed system, when tested on the NTU dataset, using the middle, left and right camera angles (M, L and R).	88
3.5	Results of the evaluation of the proposed system, when tested on the SYSU-3D Dataset and the UTKinect-Action3D dataset, using the middle camera angle M.	88
4.1	Handcrafted features, extracted from the skeletal joints. Note that each feature is calculated for a given joint J and may involve a different subset of frames. Features marked with *, are calculated using only <i>HandLeft</i> and/or <i>HandRight</i> . Also, $B(\bullet)$ denotes a 3D bounding box of a set of vectors, while $a_{B(\bullet)}, b_{B(\bullet)}$ the lengths of the sides of $B(\bullet)$. For a given joint J , J_c, J_p denote its child and parent node, respectively.	95
4.2	Classification accuracy for all feature extraction methods.	98

5.1 Results of the evaluation using the 11 activities of daily living of the PKU-MMD dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases. 109

5.2 Results of the evaluation using the PKU-MMD dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases. 110

5.3 Results of the evaluation using the 12 medical conditions of the NTU-RGB+d dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases. 111

- 5.4 Results of the evaluation using the NTU-RGB+D dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases. 112
- 6.1 Results on PKU-MMD dataset. “Bas.” / “Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”. 123
- 6.2 Results on NTU-RGB+D dataset. “Bas.” / “Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”. . 124
- 6.3 Results on SYSU-3D-HOI dataset. “Bas.”/“Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”. 124
- 6.4 Results on UTKinect-Action3D dataset. “Bas.”/“Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”. . 126

7.1	Results on PKU-MMD dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.	143
7.2	Results on NTU-RGB+D dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.	144
7.3	Results on UTKinect-Action3D dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.	145
7.4	Results on SYSU-3D-HOI dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.	146

1. HUMAN ACTIVITY RECOGNITION

1.1 Human Activity Recognition from Visual Data

Human activity recognition (HAR) is the task of recognizing the behaviour of humans, based on measurements and/or observations and is considered as a very challenging computer vision task [188], continuously attracting increasing attention within the research community [195]. HAR approaches may rely on several types of sensors that are either worn by the subject or placed within her/his environment while observing her/him or capturing interactions with objects. For example, a human activity such as *walking* may be recognized by capturing changes of her/his appearance using a camera,[135] by analyzing inertial measurements from a smartwatch or band she/he wears,[80, 103, 230] or by analyzing her/his interactions with pressure sensors placed on the floor.[31] Yet, wearable/worn sensors demonstrate below average usability and are not preferred by users [133, 91]. Also sensors placed within the user's environment either require interventions or are expensive. Thus, many low-cost solutions are based on the use of cameras which capture users' motion within the 3D space. Typical HAR applications include video surveillance, human-computer/robot interaction, augmented reality (AR), ambient assisted environments, health monitoring, intelligent driving, gaming and immersion, animation, etc. [195, 159, 12]. We should herein note that several works such as the one of Turaga et al. [181] consider the term "action" as a simple complete human motion (e.g., walking, waving), while "activity" as a sequence of actions forming a meaningful human action (e.g., picking up an apple). In this context, actions are performed by one actor, while activities either by more than one actors or an actor and an object. Another definition by Aggarwal and Ryoo [8] considers actions as combinations of small movements of body limbs, and activities as series of actions. However, in this Thesis and following the paradigm of Abu-Bakar [6], we will use "action" and "activity" interchangeably.

Inarguably, HAR from motion data is considered as one of the most challenging computer vision-related problems and may be defined as "the recognition of some human behaviour within an image or a video sequence using visual data deriving from the human's motion into the 3D space." This behaviour consists of a series of consecutive activities ("actions"). An activity may be defined as a distinct type of motion, that is performed by a human, with a relatively short temporal duration. [195] Activities are not instantaneous and typically involve multiple body parts. Special types of activities include *interactions*, which may involve either a human and an object or two humans and *group activities*, which involve more than one humans. In the context of this work, we will refer to all the aforementioned types simply as "activities." Also, note that in this work (a) we will not deal with gestures, which are instantaneous and involve at most a couple of body parts; (b) we will work with segmented sequences, i.e., video parts wherein each will contain exactly one action to be recognized.

HAR approaches that are based on cameras are typically low-cost and demonstrate more than satisfactory performance in laboratory conditions. However, in real-life conditions

they suffer from three main problems, which compromise their performance, i.e., viewpoint and illumination changes and occlusion. Specifically, viewpoint variance may occur e.g., when the subject is observed by a different viewpoint than the one(s) involved in training. However, it has been shown [148] that the decrease of accuracy due to viewpoint changes may be compensated when capturing activities by using more than one cameras. Illumination changes causing low-light conditions mainly affect video-based approaches. Of course, recent advances in technology have allowed for camera sensors that also capture depth information (which is invariant to illumination changes) and perform significantly better in low-light conditions. In that case, fusion of video and depth data may result in robust extraction of human figures, e.g., as a set of 3D points [220].

Therefore, from the three aforementioned problems, occlusion is the one that imposes most limitations. In real-life applications, partial or full occlusion may occur e.g., due to furniture or the presence of other humans. Of course, full occlusion renders HAR impossible. However, partial occlusion may significantly affect the accuracy of recognition yet depending on the subset of the visible skeleton, recognition is still possible. The effect of partial occlusion in a HAR task has been assessed by Giannakos et al. [53], wherein a classification model that was trained using non-occluded data was then evaluated using only occluded data. Specifically, occlusion was simulated by removing structured sets of captured moving human skeletons that corresponded to one/two body parts (i.e., arms and legs). The authors showed that partial occlusion of the subject, in certain cases significantly affected the accuracy of recognition, although the performance drop was tightly depending on each activity.

1.2 Applications of Human Activity Recognition

Human activity recognition has a wide range of applications [71] across various and heterogeneous fields. At the following we attempt to present a non-exhaustive list of several of its key applications:

- **Video Surveillance and Security [183]:** HAR is used in video surveillance systems to automatically detect and analyze suspicious activities, intrusions, and security breaches in real-time. It can help in crime prevention and rapid response to security threats.
- **Healthcare and Elderly Care:** in healthcare, HAR may be used to monitor the motion and activities of patients or elderly individuals who live without a carer. It is crucial for e.g., fall detection systems, medication adherence monitoring, detection of activities of daily living (ADLs) [99] and remote patient care.
- **Human-Computer Interaction (HCI):** gesture recognition, which consists a subset of HAR, is essential in HCI applications enabling users to interact with computers and devices [174, 202] using hand gestures or body movements. This technology is used in gaming, virtual reality, smart devices etc.

- Sports Analysis and Training [204]: HAR may be employed in sports' analysis to track and analyze athletes' movements. Coaches and sports scientists may use it e.g., for performance assessment, injury prevention, and strategy optimization.
- Human-Robot Interaction and Automation [161]: robots that collaborate use HAR to understand and respond to human actions. This is particularly important in collaborative robotics and service robots that work alongside humans in fields such as manufacturing and healthcare.
- Behavior Analysis [30]: researchers in psychology and social sciences use HAR for behavior analysis. This may help in studying human behavior in various contexts, including social interactions, group dynamics, and stress detection.
- Entertainment and Gaming [85]: in the entertainment industry, HAR may be used to create interactive and immersive experiences in video games, theme park attractions, and augmented reality applications.
- Assistive Technologies [71]: individuals with disabilities may benefit from HAR in the context of assistive technologies, e.g., incorporation of gesture/activity recognition to enable hands-free control of devices and home automation. This may help with tasks such as communication, control of devices, and improving overall accessibility.
- Retail and Marketing [71]: retailers may use HAR for customer behavior analysis as it may provide insights into shopping patterns, customer engagement, and store layout optimization.
- Law Enforcement and Forensics [137]: in law enforcement and forensics, HAR may be used to analyze surveillance footage, reconstruct events, and identify suspects. It may contribute in investigations and evidence collection.
- Cultural Heritage and Art [100]: museums and cultural institutions may use HAR to create interactive exhibits and installations, so as to enhance the visitor experience and make cultural heritage more accessible.
- Education and e-Learning [129]: in educational settings, HAR can provide feedback on students' engagement and participation, improving the quality of online courses and interactive learning platforms.

1.3 Research Questions/Hypotheses

As we have already mentioned, one of the main challenges in HAR tasks is occlusion. This thesis aims to propose approaches that are able to deal with partial occlusion. To this, it will try to answer the following research questions:

RQ1: To what extent is the performance of a recognition model that has been trained without occluded samples affected when presented with occluded samples?

- RQ2:** Can the performance of a recognition model, like the one mentioned above, be enhanced when presented with occluded samples if occluded samples are artificially introduced during its training phase?
- RQ3:** Is it possible to automatically complete the occluded parts of the skeleton before they are presented to the model in order to improve its performance?

The following research hypotheses will be experimentally assessed within the following chapters:

- HC1:** Since the model has been trained on samples without occlusion, its performance is expected to significantly decrease when presented with samples in which “critical” parts of the body for the specific action have been obstructed.
- HC2:** If the training process of a model is enhanced with samples that have been affected by occlusion, then the model is expected to show improved performance in evaluation.
- HC3:** If the obstructed parts of the skeleton are automatically completed before being fed into the model, it will show improved performance in evaluation.

1.4 Thesis Structure

The rest of this Thesis is organized as follows: in the remaining of Chapter 1 we present the datasets that we will use for experimental evaluation of the proposed approaches, we discuss the main visual sensors that are used in contemporary computer vision approaches for HAR and several computer vision libraries that may complement sensors. We also present an overview of research efforts in the broad field of HAR using visual data. Then, in Chapter 2 we provide the theoretical and experimental background for the methodologies that will be presented in the following Chapters. Specifically, we present the main machine learning algorithms that we have used, i.e., the Convolutional Neural Networks, the Recurrent Neural Networks and the Generative Adversarial Networks. We also provide implementation details regarding the programming languages and libraries we have used, we describe in detail the 3D skeletal data that we have used as input to our approaches and we provide the experimental evaluation protocols that we have adopted and are inline with the bibliography. Then, in Chapter 3 we begin the presentation of our research with a pseudo-colored image representation for HAR. This representation aims to capture the properties of 3D skeletal motion during an activity in a single image. In Chapter 4 we present an approach for fusion of handcrafted and contextual features for HAR. This approach aims to evaluate whether the features extracted from a Convolutional Neural Network that has been trained with pseudo-colored representations of 3D skeletal motion is able to boost performance of traditional approaches, i.e., whether these features are able to provide some extra cues for the recognition of activities. In Chapter 5 we present an approach for dealing with occlusion in HAR problems. This approach is

based on the augmentation of the training dataset with occluded samples, along with non-occluded ones. Then, in Chapter 6 we attempt to overcome the problem of occlusion using a deep regression network, whose goal is to reconstruct missing skeletal joints, while in Chapter 7 we take a further step by using a Generative Adversarial Network, whose goal is also to reconstruct missing skeletal joints. In both reconstruction approaches, classification takes place with a network that has been trained exclusively using non-occluded samples. This Thesis concludes with Chapter 8, wherein we draw our conclusions and we discuss possible future extensions of this work.

1.5 Datasets

At the following, we will present the 4 main datasets that have been used for the experimental evaluation of our work. Note that to the best of our knowledge, there is no publicly available data set comprising 3D occluded actions. To overcome this, we manually discarded subsets of joints that correspond to body parts (i.e., arms and legs), as we will discuss at the following chapters.

1.5.1 The PKU-MMD dataset

PKU-MMD [125] is a human motion activity dataset that focuses on long, continuous sequences of activities from several visual modalities. It has been captured using the Kinect v2 sensor, which as we will discuss in section 1.6.2 is able to capture color and depth videos, infrared sequences, and sequences of 3D human skeleton joints, simultaneously. The dataset consists of more than 1000 long activity sequences recorded at 30 fps, each lasting 3 to 4 min. and containing approx. 20 activity instances. The total size of the dataset is 5,312,580 frames with a total duration of more than 3K min. and more than 20K temporally localized activities. A graphical summary of the main properties of PKU-MMD is illustrated in Fig. 1.1.

In total, there exist 51 action classes, which are divided into two categories: 41 daily activities (such as drinking, waving, putting on glasses, etc.) and 10 interactions (hugging, shaking hands, etc.). Activities have been performed by 66 different human subjects. Specifically, within the data collection process each individual was asked to take part in four daily activity videos and two interaction. The ages of the participants ranged between 18–40. Several examples of activity examples of the PKU-MMD dataset are illustrated in Fig. 1.2.

In order to facilitate the sequential flow of extended action sequences, the daily activities that take place sequentially have been selected in a way that a weak “connection” is created. For instance, they create a sequence of taking off a shirt, taking off a cat, drinking water and sitting down to illustrate a typical scene that may occur upon the subject returning to her/his home. We should note that all videos only include one type of action, either daily or interaction, while 54 activity sequences have been designed. Moreover, the

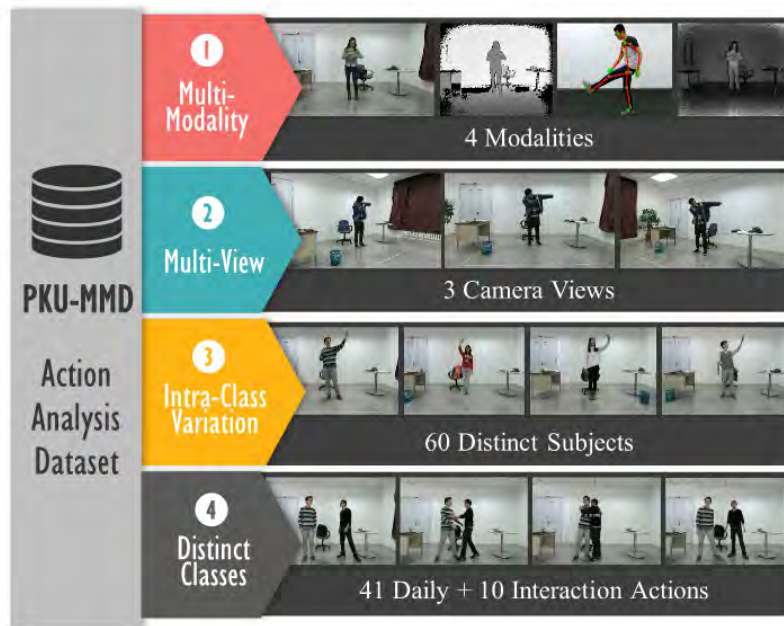


Figure 1.1: Examples of frames of the PKU-MMD dataset demonstrate the diversity of human subjects and camera angles. Figure taken from [125].

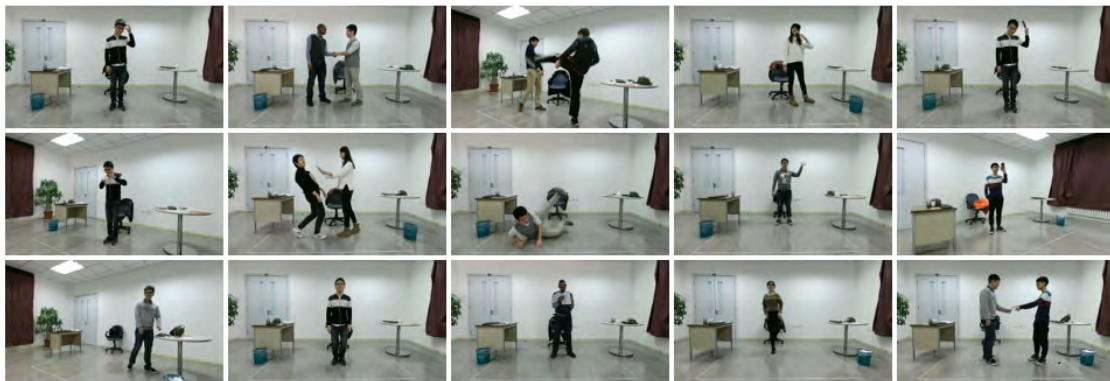


Figure 1.2: Example of activities of the PKU-MMD Dataset. From left to right, some of the activities are: first row: wear a hat, handshaking, fighting and combing my hair second row: eating , third row. Figure taken from [125].

participants have been divided into 9 groups; within each group 6 sequences have been randomly selected so as to be performed.

As we have already mentioned, PKU-MMD facilitates multimodal research, as it provides 5 categories of resources: RGB images and videos, the corresponding depth maps, infrared sequences and 3D skeleton joints of the human subjects. RGB videos are recorded in the maximum available resolution, i.e., 1920×1080 px. Depth maps are sequences of 2D depth values in mm., while the resolution of each depth frame is 512×424 px. Infrared sequences are also collected and stored frame by frame in a resolution of 512×424 px. Skeletal joint information consists of 3D locations of 25 major body joints for up to 6 detected and tracked

human subjects within the scene. The confidence of each joint point is also provided. To maintain all the extracted information, lossless compression for each individual frame has been applied.

An indoor environment from everyday life has been selected, in order to record video samples, taking into account any potential variables that could affect the results. Since temperature changes could cause infrared sequences to be inaccurate, specific care was taken for a constant room temperature. To ensure consistent lighting, the windows were covered and artificial lights have been used for all recordings. Three cameras were set up at fixed angles and heights to capture three horizontal views. The action area was 180 cm long and 120 cm wide. Each subject was asked to perform each action instance in a long sequence towards a random camera, and was allowed to do two consecutive actions towards different cameras. The horizontal angles of each camera were -45° , 0° , and $+45^\circ$, with a height of 120 cm.

All recorded videos were annotated on a frame-by-frame basis. To ensure the accuracy of the annotations, only experienced volunteers were employed to review each video and provide temporal boundaries for the actions present. However, due to the potential for discrepancies in the temporal labels of the same action from different people, the actions were divided into several groups and only one annotator was responsible for labeling the actions in each group. After the completion of this process, a set of verified untrimmed videos was obtained, each of which being associated with several action intervals and labeled accordingly.

Unlike typical annotation tasks which only require one label for each example, the probability of introducing errors while annotating temporal boundaries is much higher. During the labeling process, it was observed that an approximate 10-frame expansion of the action interval may be accepted in some cases. To further enhance the reliability of the dataset, a system of labeling correction confidence evaluation was proposed to verify and improve the manual labels. To begin with, a basic evaluation protocol is designed for each video, such as whether there is any overlap of actions or if the length of an action is reasonable. Thanks to the multi-view capturing, a cross-view method was used to evaluate and validate the data labels. This protocol had been designed so as to guarantee the consistency of video annotations for each view.

1.5.2 The NTU-RGB+D dataset

The NTU-RGB+D dataset [172] is a large-scale collection of 3D human activity data, captured from three different camera angles. It has been captured using the Microsoft Kinect v2 sensor. As in the case of the PKU-MMD dataset, it provided four major data modalities provided by this sensor: RGB frames/sequences, depth maps, infrared sequences, 3D human skeleton joints. The dataset consists of more than 56.880 long activity sequences, with 4.000.000 frames, recorded at 30 fps, with each action having a different duration.

In total, there are a total of 60 action classes, which are divided into three main categories: 40 everyday activities (such as drinking, eating, reading, etc.), 9 health-related activities

(like sneezing, staggering, falling, etc.), and 11 interactive activities (punching, kicking, hugging, etc.). For the data collection, 40 different individuals were invited to participate. The ages of the participants ranged between 10–35 years old. In Fig. 1.3 we illustrate several activity examples of the NTU-RGB+D dataset.



Figure 1.3: Examples of frames of the NTU RGB+D dataset and the NTU RGB+D+120 demonstrate the diversity of human subjects and camera angles. Figure taken from [172].

As in the case of PKU-MMD, NTU-RGB+D also facilitated multimodal research, as it provides RGB videos that have been captured in a resolution of 1920×1080 px., depth maps with a resolution of 512×424 px., infrared sequences of frames with a resolution of 512×424 and skeletal joint information, made up of 3D coordinates of 25 major body joints for humans subjects that have been detected and tracked in the scene. To ensure that all the data is retained, they used lossless compression for each frame. In Fig. 1.4 we illustrate the available data modalities.

Contrary to the PKU-MMD dataset, wherein the setup remained fixed, in this case, the



Figure 1.4: Available data modalities in the NTU-RGB+D and the NTU-RGB+D+120 dataset. From left to right: RGB data, RGB data with imposed 3D skeletal data, depth data, depth data with imposed 3D skeletal data, infrared data. Figure taken from [125].

authors constructed their dataset using 16 data acquisition setups. In each setup, they captured 3 different horizontal views of the same action sample simultaneously using 3 cameras. The cameras had been set up at the same height but from 3 different horizontal angles: -45° , 0° , and $+45^\circ$. Each subject performed each action twice, once towards the left camera and once towards the right camera. In this way, they captured 2 front views, 1 left side view, 1 right side view, 1 left side (45° view), and 1 right side (-45° view) in each collection setup. The cameras were assigned consistent camera numbers in their dataset, where camera #1 always observed the 45° degrees views, while cameras #2 and #3 observed the front and side views. To further increase the number of camera views, they also varied the vertical heights of the cameras and their distances from the subjects in different collection setups, as reported in 1.1.

We should herein note that Jun et al. [127] provided a new version of the dataset called NTU-RGB+D+120. The differences from the initial dataset were as follows: instead of 40 distinct subject, 60 action classes, 56,880 video samples with 5M frames and 16 camera angles, the newer version consists of 106 distinct subject, 120 classes, 114,000 video samples with 8M frames and 16 more camera angles. In 1.3 we depict some of the new angles 1.1, and new actions, as a brief comparison to the original.

Table 1.1: The NTU RGB+D dataset includes 16 different heights and distances of setup positions. Additionally, the NTU RGB+D+120 dataset is an expansion of this dataset, which includes additional 16 setup positions.

Setup No.	Height (m)	Distance (m)	Setup No.	Height (m)	Distance (m)
01	1.7	3.5	17	2.5	3.0
02	1.7	2.5	18	1.8	3.3
03	1.4	2.5	19	1.6	3.5
04	1.2	3.0	20	1.4	4.0
05	1.2	3.0	21	1.7	3.2
06	0.8	3.5	22	1.9	3.4
07	0.5	4.5	23	2.0	3.2
08	1.4	3.5	24	2.4	3.3
09	0.8	2.0	25	2.5	3.3
10	1.8	3.0	26	1.5	2.7
11	1.9	3.0	27	1.3	3.5
12	2.0	3.0	28	1.1	2.9
13	2.1	3.0	29	2.5	2.8
14	2.2	3.0	30	2.4	2.7
15	2.3	3.5	31	1.6	3.0
16	2.7	3.5	32	2.3	3.0

1.5.3 The SYSU 3D Human-Object Interaction dataset

The SYSU 3D Human-Object Interaction (HOI) dataset [68] is an RGB-D dataset focusing on the dynamics of human-object interactions. In the assembly of this dataset, 40 individuals have been engaged to demonstrate a repertoire of 12 distinct activities, specifically *drinking, pouring, calling phone, playing phone, wearing backpacks, packing backpacks, sitting chair, moving chair, taking out wallet, taking from wallet, mopping, and sweeping*. These activities incorporated the utilization of one of the six designated objects: a phone, a chair, a bag, a wallet, a mop, and a besom. Consequently, a total of 480 video recordings have been collected. A glimpse of some of the aforementioned activities can be seen in Fig. 1.5. For every human subject recorded while performing one of the aforementioned activities, 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, having been recorded, using a Kinect v1 camera.

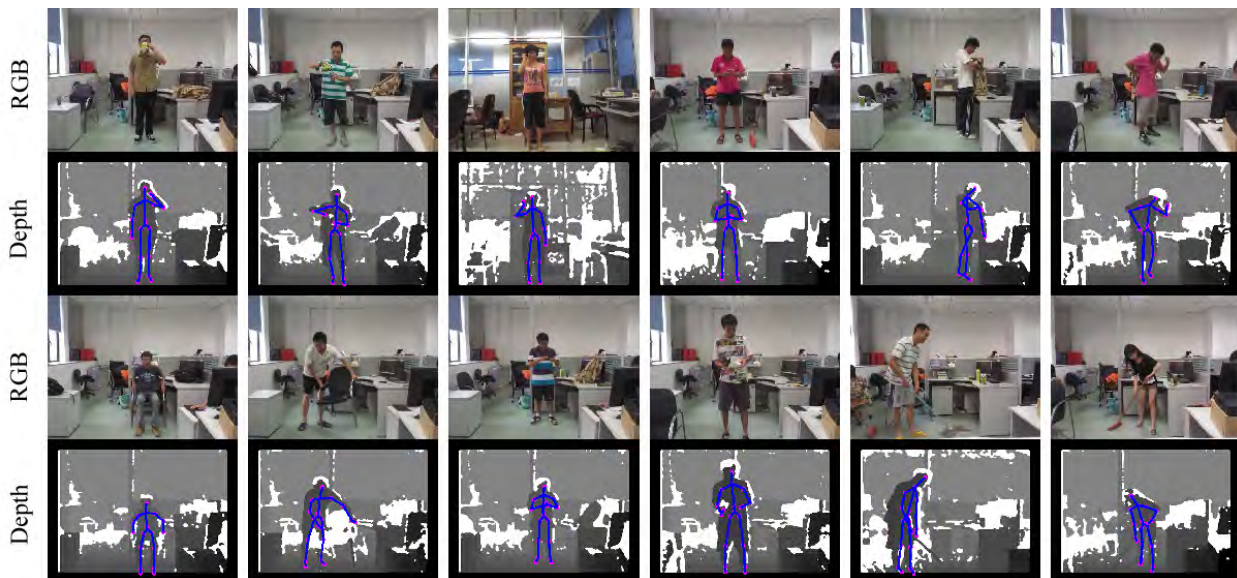


Figure 1.5: Examples of random activities and topics from the SYSU 3D HOI dataset in RGB and depth modes are displayed. Figure taken from [68].

We should herein note that according to its authors, SYSU 3D-HOI brings forth several research challenges, including:

- high similarity in movements and object appearance, i.e., certain activities showcase a high degree of resemblance in both the motions demonstrated and the appearance of the objects manipulated;
- substantial participant count at the time of the dataset presentation, wherein the number of subjects quadrupled when compared with contemporary 3D activity datasets

1.5.4 The UTKinect-Action3D dataset

The UTKinect-Action3D dataset [208] encompasses 10 distinct human activities performed in indoor environments. These activities were recorded utilizing a stationary Kinect sensor positioned at a fixed location. The data collection process involved capturing RGB images and depth maps at a rate of 30 fps, yielding resolutions of 640×480 px. for RGB images and 320×240 px. for depth maps. The assortment of actions encapsulated in the dataset were *walking, sitting down, standing up, picking up, carrying, throwing, pushing, pulling, waving hands, and clapping hands*. In Fig. 1.6 we illustrate several activity examples of the UTKinect dataset.



Figure 1.6: Examples of random activities and topics from the UTKinect dataset in RGB and depth modes are displayed. Figure taken from [208].

To ensure a rich and diverse dataset, each of the actions was performed twice by a group of 10 individuals, comprising 9 males and 1 female, with one of the individuals being left-handed. The dataset contains a total of 6220 frames derived from 200 instances of action performances. These instances vary in length, ranging from a brief 5 frames to extended sequences of up to 120 frames. A visual representation of sample frames extracted from the dataset is illustrated in Fig.1.6.

The UTKinect dataset is characterized by activity sequences captured from diverse angles, while posing three principal challenges:

- considerable variations in activity execution, i.e., regarding how different individuals

perform the same action. A notable illustration of this is the variation in the method employed to pick up objects wherein some subjects used a single hand, while others utilized both;

- disparate activity durations of different activity sequences, making the dataset more complex. A testament to this is the standard deviation in the length of “carry” sequences, which is equal to 27 frames, and the noteworthy fact that the average “carry” sequence is 48 frames longer compared to the average “push” sequences;
- occlusions and limited field of view (FOV), since in some activity examples some body parts move outside the FOV, adding an additional layer of difficulty in data interpretation.

1.5.5 Other HAR datasets

In the realm of human action recognition for AI and computer vision research, a variety of datasets beyond PKU-MMD and NTU-RGB+D prove instrumental in training and evaluating models. UCF101 [168], featuring 101 action categories, stands as a widely-utilized benchmark, while HMDB [97], with 51 action classes, provides a diverse range for algorithm assessment. The expansive Kinetics dataset [84], encompassing 600 action classes, offers a rich resource for large-scale video-based investigations. For those focusing on nuanced interactions, the Something-Something dataset [56] captures intricate human-object engagements, emphasizing hand-object manipulations. In addition, the AVA (Atomic Visual Actions) dataset [59] uniquely annotates actions in terms of their elemental visual components, contributing to a nuanced exploration of individual elements within broader activities. These datasets are not exhaustive, but they are certainly significant in the area of the HAR.

1.6 Video Sensors used in HAR

The development of innovative input devices has played a pivotal role in the fields of computer vision and human-computer interaction. In this section we present the main sensors that are used in order to capture the visual data that are used for the experimental evaluation of the research that will be presented within the next chapters.

1.6.1 The Microsoft Kinect v1

One such pioneering device is the Microsoft Kinect v1[180], often referred to as the “Kinect for Xbox 360.” Released by Microsoft in November 2010, Kinect v1, illustrated in Fig. 1.7 was initially designed for gaming. However, its impact extended far beyond the gaming world, finding applications in various domains due to its advanced depth sensing and motion tracking capabilities. This section delves into the technical features, applications,

limitations, and legacy of Kinect v1, highlighting its historical significance in the evolution of interactive technology.



Figure 1.7: A Microsoft Kinect v1 device.

In the beginning of the 2000s, there was a growing need for more intuitive ways of interacting with computers and gaming consoles. Traditional input devices, such as keyboards and controllers, often posed barriers to entry for users seeking a more natural and immersive experience. Microsoft recognized this need and embarked on the development of Kinect v1 to address these challenges. The primary motivation was to create a device that could enable users to interact with computers and gaming consoles through gestures and voice commands, ultimately enhancing the user experience.

One of Kinect v1's defining features was its advanced depth sensing technology. It employed a structured light approach, projecting a pattern of infrared dots onto the environment. An infrared camera then captured the distortion of this pattern, enabling the creation of a precise depth map of the surroundings. In addition to depth sensing, Kinect v1 was equipped with a standard RGB camera. This camera was capable of capturing color images and videos, enhancing the device's ability to interpret and respond to users' actions. The inclusion of a microphone array allowed Kinect v1 to recognize voice commands and capture audio input. This feature facilitated voice recognition and enabled chat and communication in games and applications.

The release of the Kinect for Windows Software Development Kit (SDK) [92] opened up a world of possibilities for developers. This SDK allowed them to create applications for Windows PCs, expanding the device's utility in non-gaming scenarios. Developers could harness Kinect v1's capabilities for a wide range of innovative applications. This way Kinect v1 extended its applications beyond gaming. Specifically, it found use in fields such as healthcare, where it enabled motion tracking for rehabilitation exercises, education, where it enhanced interactive learning experiences, and in the arts, it fueled creative installations and performances. Additionally, Kinect v1 played a significant role in robotics, enabling robots to perceive and interact with their environments.

Despite its innovations, Kinect v1 had certain limitations. It was sensitive to ambient lighting conditions, which could affect its accuracy. The device also had a limited range for depth sensing, and occlusions could pose challenges. These limitations influenced its adoption and use in different contexts.

Kinect v1 was succeeded in 2013 by Kinect v2, designed for the Xbox One, which offered improved performance and capabilities. However, the legacy of Kinect v1 lives on. It remains a significant milestone in the development of depth sensing and motion tracking technology. Its affordability and availability sparked innovation in various fields, marking



Figure 1.8: A Microsoft Kinect v2 device.

a crucial step in the evolution of human-computer interaction.

1.6.2 The Microsoft Kinect v2

The Kinect v2 [180], developed by Microsoft, is the second iteration of the Kinect series of motion sensing input devices. Originally designed for the Xbox One gaming console, it quickly found applications well beyond gaming, notably in the fields of computer vision, robotics, and healthcare. It consists a substantial upgrade over its predecessor, featuring an enhanced field of view, improved depth sensing, and higher resolution color and infrared cameras. The device, illustrated in Fig. 1.8 incorporates a 1080p color camera, an infrared emitter, and a time-of-flight depth sensor to capture both 2D and 3D data of its environment. This results in a depth sensing range from approximately 0.5 meters to 4.5 meters.

The most significant enhancement in the Kinect v2 is its depth sensing technology. Utilizing time-of-flight technology, the device sends out infrared pulses and measures the time taken for them to bounce back. This allows for more accurate and reliable depth maps compared to the structured light technique used in the first Kinect. The technology is robust enough to capture complex scenes and can even distinguish between different people in the same frame. One of the standout features of the Kinect v2 is its ability to perform skeletal tracking. The device can recognize and track the movement of up to six individuals simultaneously, providing 25 joints per person, including thumb and hand-tip joints. This feature has found applications in gesture-based control, human-robot interaction, and physical therapy.

Microsoft provides an SDK also for the Kinect v2, making it easier to develop custom applications for various platforms. The SDK offers pre-built functionalities like facial recognition, voice recognition, and gesture control, thereby speeding up the development process. This way, Kinect Version 2's rich data streams can be integrated seamlessly into computer vision algorithms and artificial intelligence models. Its capabilities are often used in tandem with machine learning techniques for object detection, activity recognition, and even emotion analysis. Furthermore, its depth sensing and skeletal tracking features make it particularly useful for tasks like Simultaneous Localization and Mapping (SLAM).

While the Kinect v2 is a powerful tool, it also showed several limitations. The device requires a considerable amount of processing power and bandwidth, which might necessitate specialized hardware for real-time applications. Also, environmental factors like lighting and obstructions can affect its performance. In summary, the Kinect v2 has proved

to be a versatile and robust tool for a variety of applications beyond its original gaming intent. Its advanced depth sensing and skeletal tracking capabilities, coupled with strong software support, make it a valuable asset in the fields of computer vision, robotics, and beyond.

1.6.3 The Intel RealSense camera

The Intel RealSense camera [86], illustrated in Fig. 1.9 is part of a series of depth cameras that have been designed to provide machines and devices the ability to perceive the world in 3D. These cameras are frequently used in various applications ranging from robotics to human-computer interaction, and they have become a cornerstone in the field of computer vision. RealSense cameras typically come with a variety of sensors including color, infrared (IR), and depth sensors. The depth sensor often employs a technique known as “structured light” or “time-of-flight” to capture the depth information of a scene. This enables the camera to create a 3D point cloud of the environment. The cameras also offer high-resolution color imaging, often up to 1080p, and some models come with an integrated Inertial Measurement Unit (IMU) for motion tracking.



Figure 1.9: An Intel RealSense camera.

One of the most compelling features of RealSense cameras is their depth sensing capability. This is crucial for computer vision applications that require object detection, segmentation, and tracking. RealSense cameras can perceive depth at a range of distances, often from a few cm to several meters, which makes them versatile for both indoor and outdoor applications. The depth data and color imagery collected by RealSense cameras can be easily integrated into AI and machine learning algorithms. For instance, the depth data can augment convolutional neural networks (CNNs) by providing additional spatial context, which is beneficial in tasks like object recognition and localization. This depth information is also invaluable for SLAM (Simultaneous Localization and Mapping) algorithms, commonly used in robotics.

Intel provides extensive software support for RealSense cameras, including SDKs and APIs that are compatible with a variety of programming languages and platforms. This makes it easier to incorporate these cameras into custom computer vision applications, from simple projects to complex research experiments. While Intel RealSense cameras offer a wide array of features, it is important to note their limitations. The accuracy of depth sensing can be affected by environmental conditions such as lighting and the presence of reflective or transparent surfaces. Additionally, the computational overhead for processing 3D point clouds in real-time may require specialized hardware for optimal performance.

In conclusion, Intel RealSense cameras have proven to be an invaluable tool in the field of computer vision, offering a blend of depth sensing, high-resolution color imaging, and extensive software support. Their versatility and ease of integration make them a popular choice for researchers and developers alike.

1.6.4 The Kinect Azure camera

Kinect for Azure [180] is an evolution of Microsoft's Kinect technology and stands as a testament to Microsoft's continuous efforts to bridge the physical and digital worlds. Leveraging the cutting-edge developments in artificial intelligence and computer vision, Kinect Azure provides developers with a sophisticated set of tools to create a wide array of spatially aware applications. Specifically, the Kinect Azure sensor kit, illustrated in Fig. 1.10, comes with an array of impressive features and specifications which include a) a 12 MP RGB camera coupled with a 1 MP depth camera, facilitating high-resolution imaging and depth sensing; b) a 360-degree microphone array offering unmatched sound capture capabilities; c) accelerometer and gyroscope to understand spatial orientations and movements; and d) a time-of-flight depth sensor that operates by measuring the time it takes for light to bounce back from objects, providing detailed spatial and depth information.



Figure 1.10: The Kinect Azure camera.

Kinect Azure has a broad range of applications, especially in the domains of AI and computer vision. Some of the critical applications including a) 3D scanning, for the creation of detailed 3D models of environments and objects; b) gesture recognition, for enabling natural user interfaces through gesture and body movement recognition; c) spatial analytics, for understanding and analyzing physical spaces to enhance retail experiences, smart homes, and industrial setups; d) healthcare, for facilitating rehabilitation through motion analysis and enabling remote healthcare solutions; and e) robotics, for assisting in navigation and spatial awareness for robots.

The true power of Kinect Azure comes from its seamless integration with Azure AI services, offering developers a robust platform to build intelligent applications. Developers can access a rich set of APIs and SDKs, allowing them to leverage machine learning and cognitive services to interpret and analyze the data captured by the Kinect Azure sensors.

Thus, Kinect for Azure is not just a sensor kit; it is a gateway to a realm of possibilities, enabling immersive experiences and smarter applications. By harnessing the potentials of artificial intelligence and computer vision, it is paving the way for a future where technology is more intuitive, understanding, and integrated into our physical reality. With a rich history of transforming gaming experiences, the Kinect technology, through Azure, is set to redefine how we perceive and interact with the digital world. It is an indispensable tool for researchers, developers, and innovators, helping them to bring their vision to life in unprecedented ways.

1.6.5 Other sensors

Apart from Kinect, Intel RealSense, and Kinect Azure [180, 86], several other video sensors are commonly employed in Human Activity Recognition (HAR) systems. These sensors offer diverse capabilities and cater to various application scenarios. Here's a comprehensive overview of some notable video sensors utilized in HAR:

- **Depth Cameras:** Depth cameras capture depth information alongside RGB color data, enabling the recognition of 3D body poses and movements with greater precision. Popular depth cameras for HAR include the Kea ToF Camera and PrimeSense Carmine [44, 3].
- **Thermal Cameras:** Thermal cameras capture heat signatures, making them suitable for HAR applications in low-light or challenging lighting conditions. Examples of thermal cameras used in HAR include FLIR Lepton 3 and FLIR Tau 2 [58].
- **RGB-D Cameras:** RGB-D cameras combine RGB color data with depth information, providing a rich representation of the scene. These cameras are particularly useful for HAR applications that require both visual and spatial context. A Notable RGB-D camera for HAR is the Asus Xtion [55].
- **Event Cameras:** Event cameras capture changes in pixel intensity, rather than entire frames, making them energy-efficient and capable of capturing high-speed movements. Event cameras are gaining traction in HAR applications due to their low latency and reduced bandwidth requirements. Examples of event cameras used in HAR include the Prophesee A1, Samsung DynaFace, and DVS 256 [50].
- **Time-of-Flight (ToF) Cameras:** ToF cameras measure the time it takes for light to travel from the camera to an object and back, providing precise depth information. ToF cameras offer advantages in terms of accuracy and range compared to traditional depth cameras. Notable ToF cameras for HAR include the PM3D CamCube 3.1, PMDTEC TOF Cam 2.0, and Swiss Ranger SR4000 [144, 49].

1.7 Computer Vision Libraries for Skeletal Motion Extraction

1.7.1 OpenPose

OpenPose [25] is an open-source library that focuses on the real-time detection of the human body, face, and hand keypoints. Developed initially by the Carnegie Mellon Perceptual Computing Lab, the library has gained significant attention in both academic and industrial applications. OpenPose has a wide range of uses, including human-computer interaction, augmented reality, and, more broadly, computer vision research.

OpenPose employs a multi-stage Convolutional Neural Network (CNN) to detect various keypoints on the body, face, and hands. The library can identify up to 135 keypoints (body, face, and hands combined) and is capable of detecting multiple persons simultaneously. It can operate on a standard GPU, making it accessible for various research and commercial applications. One of the most salient features of OpenPose is its ability to estimate human poses in real-time. The algorithm can identify the positions of 25 body keypoints, 70 facial keypoints, and 20 keypoints per hand. This level of granularity allows for detailed activity recognition and has opened new avenues in human behavior analysis. Note that OpenPose can detect and estimate poses for multiple people in a single frame, making it highly useful in crowded or social scenarios. This is particularly beneficial for applications like crowd monitoring, team sports analysis, and collaborative robotics. Examples of skeletons extracted by OpenPose are illustrated in Fig. 1.11.

OpenPose seamlessly integrates with other computer vision and AI technologies. The keypoint data it generates can be used as an input for various machine learning algorithms, including those involved in gesture recognition, action classification, and human-robot interaction. The library can also be combined with other technologies like depth sensing cameras to create more robust and versatile systems. Being open-source, OpenPose has a large and active community that contributes to its ongoing development. It provides bindings for multiple languages like C++, Python, and even offers a Unity plugin. The library is compatible with popular platforms like Windows, macOS, and Linux, making it highly versatile.

While OpenPose offers impressive capabilities, it does come with some limitations. The computational requirements can be quite high, particularly for real-time multi-person detection, which may necessitate the use of powerful GPUs. Additionally, the performance can be impacted by factors like lighting conditions, occlusions, and the quality of the input images or video streams. In summary, OpenPose has emerged as a leading tool in the field of pose estimation and human activity recognition. Its real-time capabilities, accuracy, and versatility have made it a go-to solution for researchers and developers interested in understanding and leveraging human motion and behavior.



Figure 1.11: Example skeletons extracted by OpenPose. Figure taken from[25].

1.7.2 PoseNet

PoseNet [88] is a vision model developed by Google's TensorFlow team, designed to estimate human poses in real-time. Unlike many other pose estimation models that require substantial computational resources or specialized hardware, PoseNet can run directly in a web browser using TensorFlow.js. This offers a highly accessible entry point for developers and researchers interested in incorporating pose estimation into their projects. PoseNet operates on single RGB images and returns pose keypoints indicating the positions of different body parts. The model uses a Convolutional Neural Network (CNN) architecture and can work with any browser that supports WebGL. PoseNet can estimate either a single pose fast or multiple poses with higher accuracy, making it versatile for various applications. Examples of skeletons extracted by PoseNet are illustrated in Fig. 1.12.

Specifically, PoseNet is capable of identifying 17 body part keypoints, including eyes, ears, elbows, shoulders, and so forth. The model outputs a confidence score for each keypoint, offering an indication of the estimation's reliability. While it may not be as granular as some other models like OpenPose, it provides a good trade-off between speed and accuracy. One of the standout features of PoseNet is its ability to run entirely in a browser. This is particularly advantageous for client-side applications where sending data to a server for

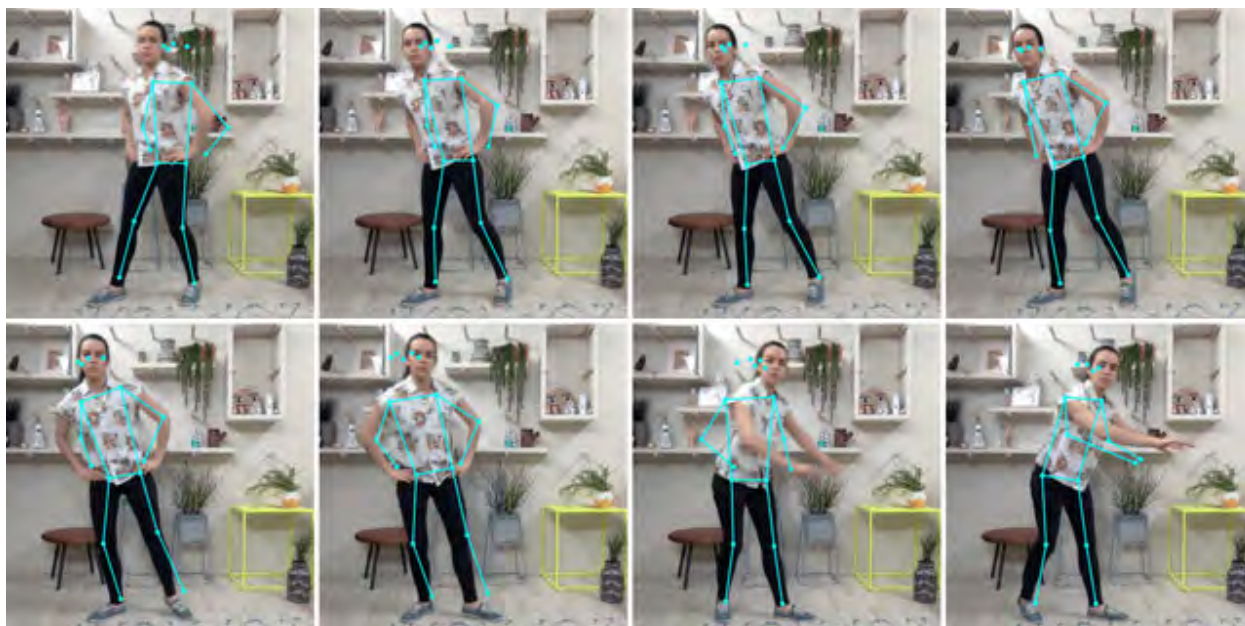


Figure 1.12: Example skeletons extracted by OpenPose. Figure taken from [88].

processing would be impractical or where privacy is a concern. The model is lightweight enough to run on a variety of devices, including smartphones and tablets.

PoseNet is often integrated into web applications for real-time interaction, augmented reality, and other client-side functionalities. Being a TensorFlow.js model, it's easy to integrate with other web technologies and libraries, providing a seamless user experience. Due to its real-time capabilities and easy-to-use API, PoseNet is well-suited for a wide range of applications. These include interactive installations, fitness coaching apps, dance tutorials, and even ergonomic monitoring systems. Its ability to run in a browser also makes it ideal for educational purposes, lowering the barrier of entry for those interested in computer vision and AI.

While PoseNet offers the advantage of lightweight, browser-based pose estimation, it does have limitations. The model's accuracy can be affected by factors such as image quality, lighting conditions, and occlusions. Moreover, being a 2D model, it does not provide depth information, which might be crucial for certain applications. In summary, PoseNet provides a unique blend of accessibility, real-time performance, and ease of use. Its browser-based architecture makes it an excellent choice for developers looking to integrate pose estimation into web applications or for researchers exploring the potential of real-time human interaction in virtual spaces.

1.8 Contemporary Research Efforts on HAR

During the last decades, many researchers in the fields of computer vision and pattern recognition have turned their efforts to the problem of HAR, using visual data, leading to

several challenges in the learning and classification approaches. There does not exist a unified end-to-end classification strategy which is able to describe all available techniques. In this section we will attempt to provide a non-exhaustive survey of the broad research area of HAR using visual data, following the taxonomy that has been used by Abu-Bakar [6], who considered a basic HAR pipeline comprising input data, feature extraction, descriptor formation, and classification. Thus, in this Section we will use these categories to present relevant research works.

1.8.1 Data Input

Originally, HAR approaches focused on visual data of human-centric motion captured by optical cameras, which may provide RGB and/or depth data or skeleton joint sequences. Moreover, multi-modal approaches may combine visual and non-visual data.

- *RGB/color-based*: this is the simplest case of input data and still remains the predominant choice since (a) such cameras are extensively used in surveillance and embedded devices, while they are available as webcams in a plethora of personal computers and laptops; and (b) they provide a cheap way for capturing visual data. Many contemporary approaches [218, 154, 141, 93, 223, 22, 117] use as input color-based images and directly work using the intensity values from the R, G, and B channels. An example of a simple RGB video frame of human motion data is illustrated in Fig. 1.4.
- *RGB + depth-based*: as we have already discussed in section 1.5, the plethora of the RGB-D cameras that have been made commercially available, such as the two versions of the Microsoft Kinect, or the Intel Realsense, have allowed researchers to incorporate depth into their approaches, so as to enrich the color-based information. We herein remind that RGB-D cameras do not only capture intensity values of the scene but they also provide depth images, i.e., depth information per pixel/region. The inclusion of depth has been shown to enhance pose estimation, improve robustness to illumination changes, and simplify foreground (e.g., human) segmentation from the background. Noteworthy HAR approaches utilizing depth information may be found in [191, 146, 2, 138, 42]. An example of a depth video frame of human motion data is illustrated in Fig. 1.4.
- *Skeleton-based*: The availability of portable and affordable depth sensor devices has sparked increased interest in recognizing human activity through human skeleton images. By using depth images, specialized algorithms are able to create skeleton representations, as illustrated in Fig. 1.4. Within such images, skeletons represent humans as “stick figures,” depicting human limbs’ joint positions with dots while they are connected by lines following the body structure. The number of joints varies, e.g., Kinect v2 offers up to 25 effective joint positions. The use of skeleton images/sequences in HAR addresses several challenges which are often faced by color-based approaches, such as illumination changes, background clutter, pose

variability, and appearance dependency on texture. Notable skeleton-based approaches are found in [1, 206, 26, 205].

- *Multimodal-based*: several works combine data acquired from both optical and non-optical devices. An obvious choice in this case are wearable inertial sensors, such as those mentioned in [28, 57, 27], recording acceleration and velocity signals from the attached body parts to provide additional cues for activity recognition. Moreover, fusion with audio signals has also been explored by various researchers [162, 192, 197]. In real-world scenarios, the performance of any HAR system should be robust across camera viewpoints and capable of considering interactions between individuals and objects.
- *Multi-view-based*: multi-/cross view approaches utilize more than one cameras at distinct, fixed locations and angles, to simultaneously record actions performed by an actor and resulting in diverse feature descriptors [158, 121, 116, 34, 123]. In Fig. 1.13 a sample from a multi-view approach is illustrated. Multi-view approaches may be also based on depth [28, 27, 29, 182]. Such approaches may provide robustness in the presence of occlusion, since in some views the subject should be fully captured.
- *Multi-actor-based*: most HAR algorithms focus in actions that are performed by a single subject. In more complex and realistic scenarios, interactions between two or more actors are considered. Contemporary approaches face multi-actor problems using color [93], or by adding depth [121, 51]. In Fig. 1.14 examples of multi-actor interactions are illustrated.
- *Object/scene dependent*: in real-life scenarios, it is common and also expected to observe human actions interacting with surrounding objects. The latter may be used as contextual information, which could be integrated in the HAR process, so as to enhance performance. This is also termed person-to-object or human-to-object interaction. For example, recognizing that an actor is holding a cricket bat with a ball nearby causes the action “swinging the bat to hit a ball” to be inferred. Numerous works in the literature explore human-to-object interactions, as seen in [222, 67, 223, 163, 119]. In Fig. 1.15 we provide examples of human-to-object interactions.



Figure 1.13: Example of multi-view scenery which shows a person waving his left hand, observed from four different viewpoints. Figure taken from [158].



Figure 1.14: Examples of multi-view (side and frontal views) setup. Figure taken from [121].



Figure 1.15: Examples of sports action with human-to-object interactions. Figure taken from [67].

1.8.2 Feature Extraction

Several studies have employed diverse features to represent human actions and activities, depending on the input data. Broadly, these features can be categorized as either global- or local-based.

- *Global/holistic*: in this case the complete human form is leveraged to capture dynamics, which are derived from silhouette, skeleton, or depth information. Silhouettes are typically obtained upon a background subtraction step, while scale and rotation

invariant color-based features are extracted to ensure consistent recognition performance. Typical approaches include grey-level co-occurrence matrix (GLCM) for silhouette extraction [186] and features such as contour compactness and Zernike moments [209]. When silhouettes incorporate depth information, they are more robust to texture and illumination changes. In that case descriptors extracted may be Motion History Images (MHI) and Depth Motion Maps (DMM) [74]. Moreover, the addition of depth has been shown to facilitate the use of silhouette images in deep learning frameworks [93]. Instead of silhouettes, another popular approach is to use frame differences to calculate optical flow [118]. Optical flow may be also used for descriptor learning and may be fused with color and texture. Wu et al. [199] utilized the original human shape, represented by a bounding box, to compute Histogram of Oriented Gradients (HOG), decomposing actions into sequences of attributes. As it has been already mentioned, the availability of inexpensive depth cameras has shifted attention to using skeletons or complete human joint locations as global features. Techniques include analyzing human shapes through joint locations, [42, 1, 90]. Another approach involves grouping joint locations into related body parts and tracking them temporally [106, 116]. In this context, discriminative pairs and combinations of skeleton joints may be used [206, 205], while several approaches use all available joints [51]. Markov Random Fields (MRF) have been used to encode each limb as part of predefined movements called states to identify noisy or irregular joint locations caused by sensor error [26]. Depth information is often used as an additional cue, either combined with color-based images or for extracting robust skeleton joint locations. Common approaches include weighted hierarchical Depth Motion Maps (WHDMM) using depth information for motion tracking [191], computation of eigenvalues using 3D point clouds and resulting to a histogram of oriented principal components [29]. Simple DMMs may be constructed by frame differencing on depth planes [115], or discrete wavelet transform (DWT) coefficients [138]. Multiple views may be obtained using projection schemes [101]. DMMs may also be stacked to compute volume motion templates (VMTs) [39], from which 3D oriented gradient may be extracted as features. Global features may be combined to enhance performance. Notable examples are the combination of shape, optical flow and interest points [146], MHI with depth map images [74]. Other approaches include the use of temporal difference of consecutive depth map frames, skeleton joints, and acceleration/angular velocity from a wearable inertia sensor and computing statistics before fusing them with DMMs [32], combination of skeletal joints with object and interaction features [119] or combination long-term common shapes and motion trajectories from a layered elastic motion tracking method based on human edges [228].

- *Local*: While global features capture complete information about human motion, they are less robust against noise, occlusion, and changes in illumination. Therefore, methods based on local features, such as Scale-Invariant Feature Transform (SIFT) [213], Speeded-Up Robust Features (SURF) [15], Space-Time Interest Points (STIPs) [38], dense sampling [200], and texture-based features like Local Binary Patterns (LBP) [153] and Grey-Level Co-occurrence Matrix (GLCM) [231], have gained

popularity due to their success in addressing these challenges. SIFT and SURF are commonly used for reliable feature extraction from still images. STIP and SIFT are fused in [225], gradient features are used in [93, 218] and [154] from 3D regions (cuboids) that are formed around interest points. For representation, histograms are used. Additional features such as HOGs or optical flow from STIP are often used [197, 225, 182, 157, 155]. Dense sampling methods have also been employed for local feature extraction [141, 117, 69, 78, 130, 224], such as HOG, HOF, and motion boundary features, or optical flow [34]. Dense trajectory features have been fused with audio cues in [162], while SURF and optical flow have been used to remove camera motion, prior to feature extraction [189, 177]. Texture-based features are less popular but are used to represent either global or local features, often combined with local or global features, respectively. Most common texture features are Local Binary Patterns [7], which have been used in [4, 167] and spatial distribution of gradients [186]. A unique approach is in [229], introducing a new local feature based on a block-matching approach, offering denser and more accurate motion trajectories compared to traditional optical flow techniques.

1.8.3 Descriptor Formation

Once relevant features are extracted, the subsequent step involves grouping these features into a form that is representable or descriptive for easy classification of actions. This step is crucial for traditional machine learning approaches, but not necessary for deep learning approaches, wherein features are automatically learnt. Commonly, extracted features are combined to form a vector-like descriptor. Since such descriptors often have high dimensions, feature reduction techniques such as Principal Component Analysis (PCA) or Linear Discriminant Analysis (LDA) are often applied. In many HAR approaches, multiple feature vectors are involved and are concatenated (fused) into a single descriptor.

- *Global-based*: Intensity-based approaches for global features employ various strategies. Traditionally, feature vectors are concatenated for classification [186, 201, 209, 175]. Silhouette images may undergo dimension reduction before classification [17, 221, 74]. Bag-of-Words represents action classes on silhouette sets [169]. Optical flow may be fused with color [118], while shape features may be concatenated with trajectory ones [228]. Skeleton-based features capture pose and motion descriptors [1, 116, 206, 26, 35, 62, 2, 63, 107, 205, 108, 42, 90, 106]. Linear locally embedding reduces joint coordinate feature size [16]. Multi-person scenarios use joint distance and motion distance measures clustered by k-means [70]. Depth-based features use HOG on DMMs [101, 29]. To handle action speed variations, methods concatenate temporal energy or fuse with multi-scale temporal pyramid energy [158, 39, 138, 29, 81, 115].
- *Local-based*: most popular local-based approach involves the use of bag-of-words (BoW) or bag-of-features algorithms to obtain descriptors. Typically, the latter are a result of clustering features from HOGs, Histograms of Optical Flow (HOF), motion

boundary histogram (MBH), or combinations of them through concatenation. The k-means clustering technique is then applied to form the visual words [116, 24, 218, 120, 154, 141, 197, 182, 227, 45, 78]. An alternative to BoWs is the formation of a descriptor using the Fisher vector approach, as adopted by [157]. Some attempts have been made to implement the k-means clustering method directly on HOG or HOF [226, 102]. Others simply use HOF on selected and tracked points from different views and concatenate them to form a descriptor [203]. A more direct approach is to concatenate the low-level features, such as HOG, HOF, MBH, and trajectories, to form feature vectors [224].

- *Hybrid*: some works combine local and global features to leverage the strengths of both methods. For instance, in [146] local descriptors have been employed using BoW obtained from STIP and local optical flow. In this approach the authors opted for a late fusion approach. Similarly in [32] both local and global descriptors, have been used with descriptors obtained from an inertia sensor. Finally in [78] local features were fused with learnt features from a CNN.

1.8.4 Classification

Classification is the last step in the HAR process, involving the categorization of actions based on obtained descriptions. Classifiers in HAR typically fall into three types: supervised, unsupervised, and semi-supervised. However we consider deep learning approaches as a separate category.

- *Supervised*: SVM consist the dominant “shallow” approach. Various SVM variations include single-class SVM, latent SVM, multi-class SVM, and multi-class Kernel SVM and have been used in HAR [197, 123]. Sparse representation-based classifiers such as the collaborative representative classifier (CRC) have been also used [138, 115, 32, 102]. Probabilistic approaches include topic modeling [bian2011cross], transfer learning with topic modeling and fuzzy logic [24], and scene topic model boosting [226]. Random Forest was employed in [62], while a region-based mixture model based on shape and motion distribution probabilities was employed in [228]. Dynamic programming-based approaches like the inference model [108] and DTW method [107] have been applied. Multi-task learning (MTL) was used in [206] to reveal relationships between features and action classes and in [119] in joint HAR. Multi-instance learning was used in [26, 218]. MTL variants were used in [205, 116]. Finally, transfer learning was adopted in [117], employing video–video transfer from simple to complex actions.
- *Unsupervised*: several unsupervised approaches like k-NN have been employed in HAR. Works utilizing k-NN include [42, 17, 121, 4]. In [186] a hybrid classifier cascading SVM with a 1-NN classifier was proposed. Decision trees have demonstrated outstanding performance [4]. Collaborative representation classifier based on sparse representation, has also been applied in HAR [28, 27, 29].

- *Semi-supervised*: semi-supervised methods adapt knowledge derived from images to classify target video actions. However, they have not been popular in the problem of HAR.
- *Deep-learning*: CNNs have gained significant attention in HAR research efforts. Deep learning transforms the classification problem framework, automating feature extraction and descriptor formation within the deep learning framework, reducing human intervention once the network is trained. For instance, in [191] and [90] CNNs have been employed in a late fusion approach. In [222] deep learning was applied to combine semantic cues involving human action, object, and the scene to determine the event. Additionally, the extreme learning machine (ELM) approach under the neural network framework has been considered, as proposed in iosifidis2015graph. Another variant is the recurrent neural network (RNN), tracking joint information of the subject [90]. More approaches from this category will be presented in the following Chapters.

2. THEORETICAL AND EXPERIMENTAL BACKGROUND

2.1 Artificial Intelligence

Since the 1950s, Artificial Intelligence (AI) has been a field of exploration, with the aim of automating intellectual tasks that are usually performed by humans. AI is a broad field that includes machine learning and deep learning, as well as other approaches that may not involve any learning. Before the 1980s, AI textbooks rarely mentioned learning. Early chess programs, for example, were created using hard-coded rules written by programmers, and did not involve machine learning. For a long time, many experts thought that human-level AI could be achieved by having programmers craft a large set of explicit rules to manipulate knowledge stored in explicit databases. This approach is known as “symbolic AI” and was the dominant paradigm in AI from the 1950s to the late 1980s, reaching its peak popularity during the expert systems boom of the 1980s. However, symbolic AI was found to be inadequate for solving complex, fuzzy problems such as image classification, speech recognition, or natural language translation. In Fig. 2.1 we may observe the basic components of AI. Each of these subsections is further divided into sub-categories, such as supervised learning, unsupervised learning, and reinforcement learning for machine learning.

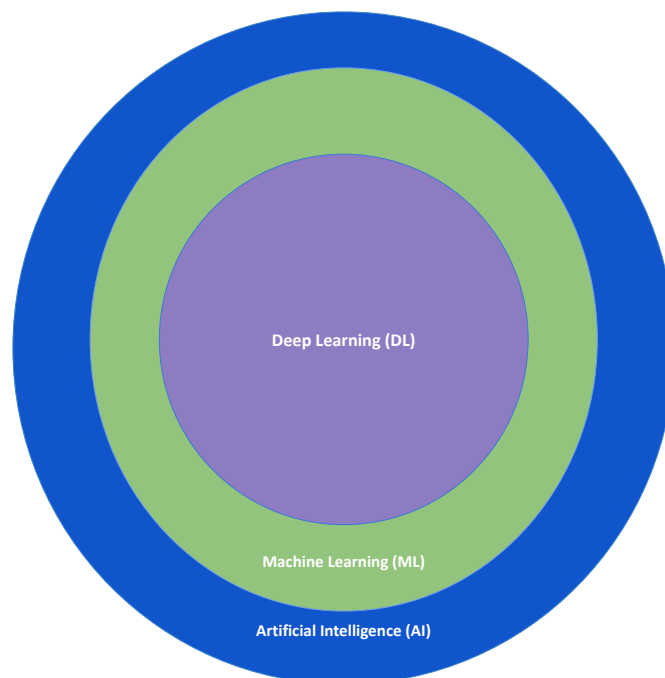


Figure 2.1: A Venn diagram illustrating the relation of machine learning (ML) and deep learning (DL) to artificial intelligence (AI). Specifically, ML is a subset of AI, while DL in turn is a subset of ML.

2.2 Machine Learning

Machine learning (ML) is a sub-field of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit programming. The essence of machine learning lies in the ability of a system to learn and improve from experience, making it particularly well-suited for tasks that involve pattern recognition, decision-making, and prediction. Typical ML approaches may be categorized into the following categories.

- *supervised learning*: the model is trained on a labeled dataset, where the input data is paired with corresponding output labels. Thus, it learns to map inputs to outputs and can make predictions on new, unseen data;
- *unsupervised learning*: the model is given data without explicit instructions on what to do with it, therefore, it attempts to discover patterns and relationships in the data, often through clustering or dimensionality reduction;
- *reinforcement learning*: the model learns by interacting with an environment, by receiving feedback in the form of rewards or penalties, enabling it to learn optimal strategies to achieve a goal.

A variety of ML algorithms exist, with some of the more popular ones being:

- *linear regression*: it is typically used for predicting a continuous outcome, based on one or more input features;
- *decision trees*: they are tree-like models used for classification and regression tasks, by recursively partitioning the data;
- *support vector machines (SVM)*: they are effective for classification and regression tasks by finding the hyperplane that best separates data into different classes.

Apart from the aforementioned algorithms, *Artificial Neural Networks* (ANNs) probably consist the most popular one. Specifically, ANNs are a subset of ML models that are based on the principle of connectionism that is encountered in biological neural networks found in animals' brains. We should note that these algorithms are the foundation of deep Neural Networks, which we will discuss later. Specifically, an ANN is composed of interconnected units, or nodes, which are analogous to the neurons found in biological brains. These connections, known as synapses, are able to transmit a signal from one neuron to another. Each connection has a number associated with it, which represents the "strength" of the signal. The output of a neuron is determined by a non-linear function applied to the sum of its inputs, and the weights of the neurons and edges can be adjusted during the learning process. This adjustment can either increase or decrease the signal's strength. Additionally, neurons may have a threshold, which allows a signal to be sent only when the cumulative signal surpasses that threshold. In most cases, neurons are organized into layers, where each layer may perform distinct transformations on their inputs.

Signals traverse from the initial layer (the input layer) to the concluding layer (the output layer), potentially traversing these layers multiple times. In Fig 2.2 there is an example of an ANN.

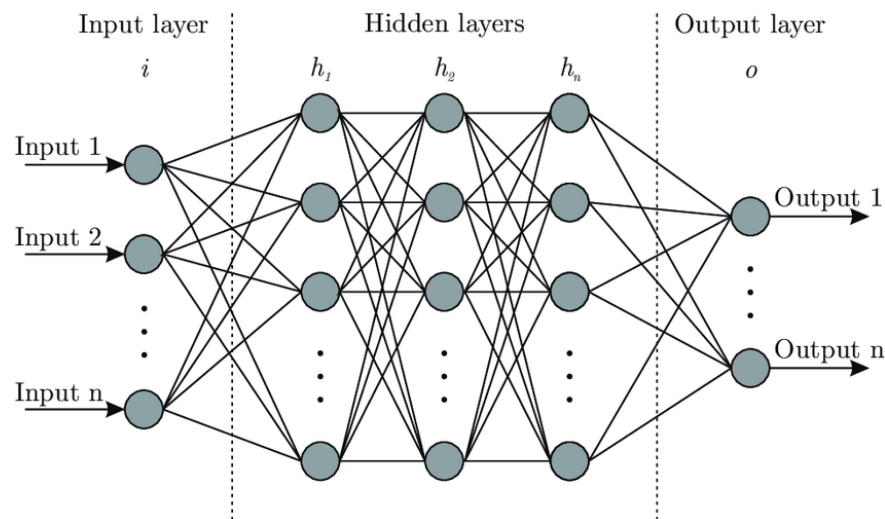


Figure 2.2: An Artificial neural network architecture. Figure taken from [19]

2.3 Deep Learning

Deep learning [105] is a sub-field of machine learning that focuses on the development and application of artificial neural networks that are able to model and solve complex problems. What sets deep learning apart is its use in *deep neural networks*, which are composed of multiple layers of interconnected nodes (neurons). These networks are capable of learning intricate hierarchical representations from data, allowing them to automatically discover patterns, features, and representations.

Deep learning structures, including deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, convolutional neural networks, and transformers, have been utilized in various domains. These include computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs. In these fields, they have demonstrated results that are on par with, and sometimes even exceed, the performance of human experts.

2.4 Convolutional Neural Networks

In the realm of computer vision, the most prominent deep learning architecture is undoubtedly the Convolutional Neural Network (CNN) [104]. CNNs can be seen as a major modification of traditional neural networks (NNs). A traditional NN consists of layers of

interconnected, non-linear neurons; the output of each one is calculated by applying a non-linear function to the sum of its inputs. The connections between neurons usually have a weight. The aim is to learn the set of weights of the entire network through a learning process. At the following we will present Convolutional Neural Networks (CNNs), which comprise the main deep architecture that will be used in the rest of this Thesis for the tasks that we have already mentioned in Section 1.3.

Typically, the input of a CNN is a raw image, composed of its pixel values. The goal is to learn a set of convolutional filters that can be applied to the input, with the aim of optimizing the classification error at the output of the network. CNNs can have an arbitrary number of layers, but the neurons in each layer are only affected by a sub-region of the previous layer, not the entire image. The filters are used to “learn” how to extract features in multiple levels of abstraction. CNNs are powerful models that provide better generalization while keeping the number of free parameters relatively low compared to traditional approaches. They do not require any prior knowledge in the domain of applications, as they omit the feature extraction step. The training process of a CNN is similar to that of a typical NN; it involves a forward propagation of data, followed by a backward propagation of the error to adjust the set of weights. Common approaches used in the learning process are gradient-based variations. The key component of a CNN is the set of convolutional layers, but layers may also fall into three other categories: pooling, normalizing and fully-connected layers. At the following we will present these four main components of CNNs.

2.4.1 Convolutional Layers

As we have already mentioned, a CNN is an artificial neural network that includes one or more convolutional layers. As we will see, these layers are especially useful for processing spatial patterns, thus ideal for computer vision applications. Specifically, convolutional layers are made up of “collections” of kernels, also known as filters. Each kernel is a small window (i.e., a “patch”) that scans across the image (more accurately, the filter *convolves*) from the top left to the bottom right as it may be illustrated in Fig. 2.3. This process is based on the response of a neuron in the visual cortex to a stimulus, making it biologically inspired.

Kernels are actually 2D arrays, comprising weights that are learned during the training phase of the network. The size of a kernel may vary. A typical size for working with grayscale images is a 3×3 -pixel window, comprising 9 weights, plus a bias term b , making a total of 10 parameters. In comparison, when working with RGB images, the same kernel would have a size $3 \times 3 \times 3$, i.e., three times as many weights and a total of 28 parameters. As we have already seen in Fig. 2.3, the kernel is placed in discrete image positions during the convolution process. For a given 3×3 kernel, during forward propagation, $w \cdot x + b$ is calculated at each image position, where x denotes the image input, b the bias term, while w the kernel weight. Note that the aforementioned equation corresponds to the calculation of the weighted sum $w \cdot x$, in which products are calculated element-wise based on the alignment kernel and image, i.e., as if the kernel were superimposed over the pixel values. For the kernel and the image patch of Fig. 2.4, $w \cdot x$ is calculated as follows:

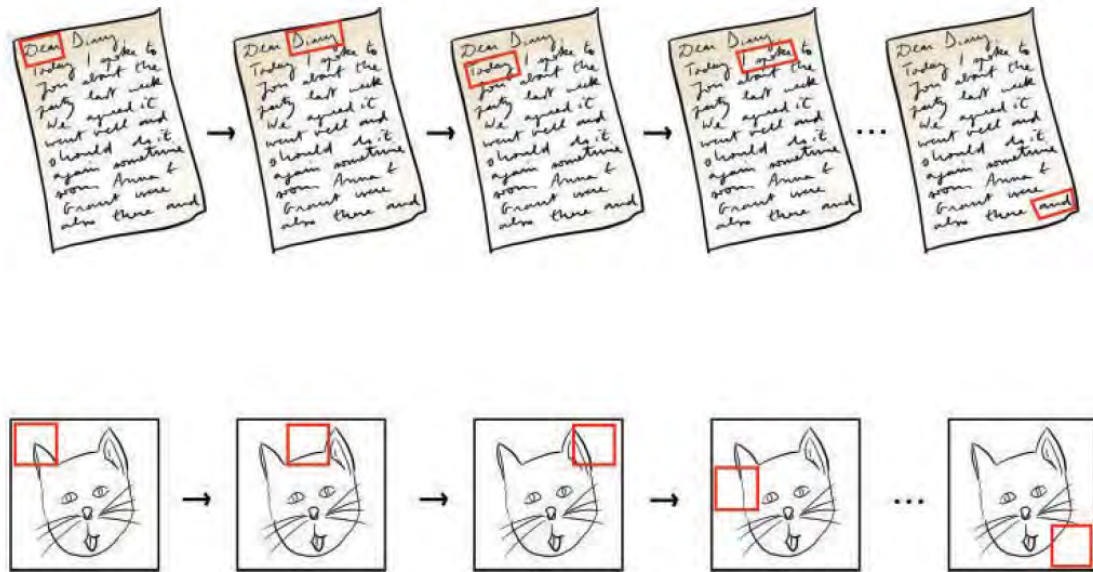


Figure 2.3: Top image: Consider reading a book written in English: to read a specific page, one begins from the top-left corner, moving to the right and proceeding to the next row upon reaching the end of a line, repeating this process until the bottom-right corner is reached. At this point, all words of the page have been read. Bottom image: Similarly to the aforementioned process, a kernel in a convolutional layer scans a given image from top to bottom and from left to right. Figure taken from [96].

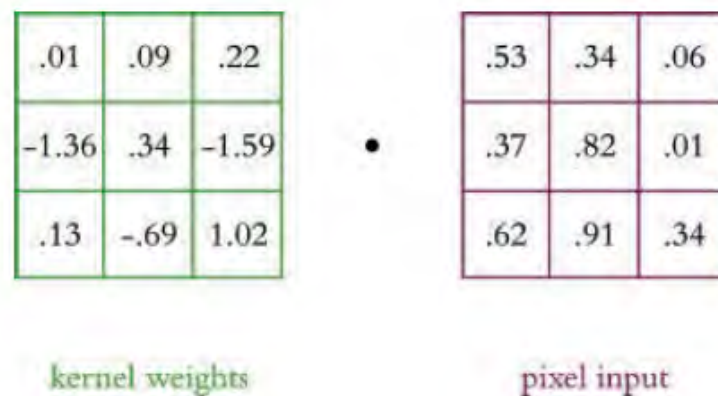


Figure 2.4: Left: a 3×3 kernel; Right: a 3×3 pixel window. Figure taken from [96].

$$w \cdot x = .01 \times .53 + .09 \times .34 + .22 \times .06 + (-1.36) \times .37 + .34 \times .82 + (-1.59) \times .01 \\ + .13 \times .62 + -.69 \times .91 + 1.02 \times .34 = -0.3917 \quad (2.1)$$

A bias term b (for example, -0.2) to obtain z is simply added as follows:

$$z = w \cdot x + b = -0.39 + 0.2 = -0.19 \quad (2.2)$$

At this point, the value of z may be used so as to calculate an activation value, a . This may be done by passing z through a non-linear activation function, such as the tanh or ReLU function. Convolutional kernels have weights, inputs, and a bias, and a weighted sum of these is produced. In a typical convolutional layer, multiple filters are used to scan the input image. Each filter is specialized to detect specific patterns, much like the simple cells in biological visual systems. For instance, one filter may be optimized to identify vertical lines, producing high activation values when it encounters them. Subsequent layers in the network receive these activation maps, and filters in these layers react to more complex combinations of simple features. This leads to a hierarchical representation of increasingly abstract spatial patterns, enabling the network to recognize complex objects.

We should herein note that the number of filters is a configurable hyperparameter. To find its optimal value, one should consider the following:

- more filters may help the network identify complex features, hence at the cost of computational efficiency;
- the ideal number of filters may differ from one layer to another, given that early layers focus on simple features and later layers on complex ones;
- to optimize performance, one should use the smallest number of filters that still yields a low validation cost.

Overall, the goal is to find a value allows the network to effectively learn from the data while minimizing computational requirements.

At the following, we will present a brief convolutional example. Let us consider an image patch from an RGB image. Let the dimension of the convolutional kernel be $3 \times 3 \times 3$ pixels, as shown in 2.5, where we also illustrate 3×3 matrices for each of the R, G and B channels. To make sure the feature map has the same size as the input data, the image has been padded with zeros on all four sides. Note that the kernel matrices w are different for each channel. Also, let the bias term b be 0.2 . The current position of the filter is indicated by an overlay on each array of pixel values, and the z value (calculated by taking the weighted sum from Eq. 2.1 across all three color channels, and then adding the bias as in Eq. 2.2) is given at the bottom right of the figure. In the end, all of these z values are added together to create the first entry in the feature map at the bottom right, as it may be seen in Fig. 2.5. The same procedure is repeated upon moving the filters by one pixel to the right and is concluded when it reaches the bottom right pixel of the image patch.

Upon the completion of the aforementioned procedure, its outcome are the z -values, also arranged in a 3×3 grid, and then transformed into a corresponding 3×3 activation map by applying an activation function, like the ReLU function, to each z -value. Given that a single

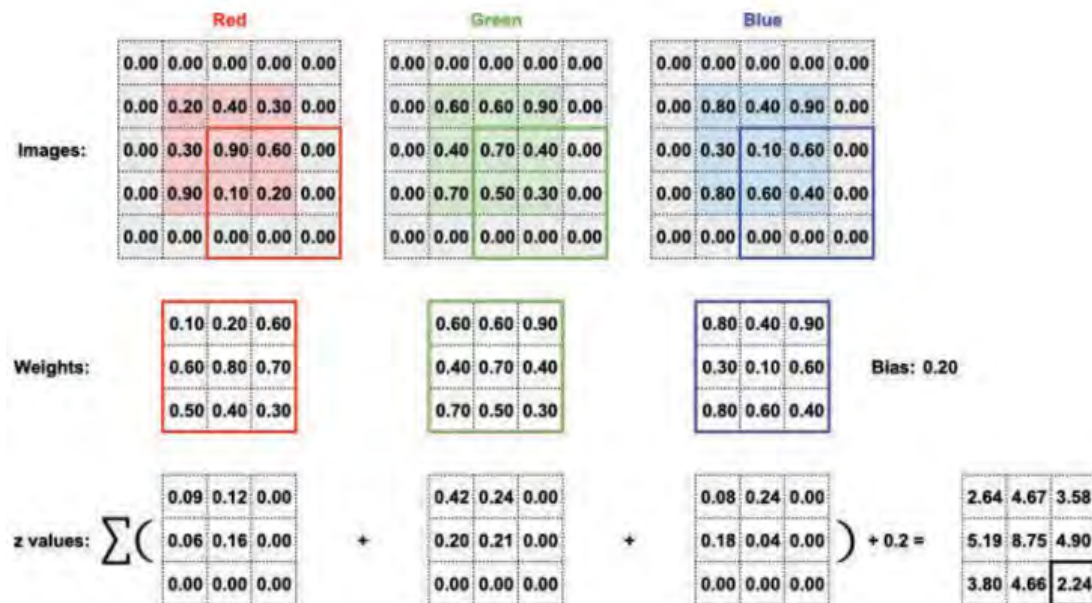


Figure 2.5: A visual example showing how the activation values in a feature map are calculated in a convolution layer. Figure taken from [96].

convolutional layer generally contains multiple filters, each one creates its individual 2D activation map, adding a depth dimension to the overall activation map. This is somewhat akin to how RGB images have depth derived from three channels. These “channels” in the kernel’s activation map symbolize the specific features the kernel is tailored to identify — for instance, a line with a definite orientation representing an edge. Figure 2.5 portrays the formation of a 3D activation map through the calculation of activation values from the input image. The convolutional layer depicted in Fig. 2.5 incorporates 16 kernels, each contributing to the activation map’s depth, culminating in a map endowed with 16 distinct “channels.”

2.4.2 Pooling Layers

Pooling layers are typically placed after a convolutional layer or a set of convolutional layers. They work by taking a subset of neurons from the previous convolutional layer and producing a single output from each subset, such as the maximum or average value. This way, the outputs of a group of neurons in the previous layer are combined into a single neuron in the next layer. This allows for the extraction of complex features from different resolutions. Additionally, it has been demonstrated that pooling layers are an effective way to prevent over-fitting, which is a major cause of poor generalization to unseen data. In Fig. 2.6 we illustrate a simple demonstration of the way pooling layers work.

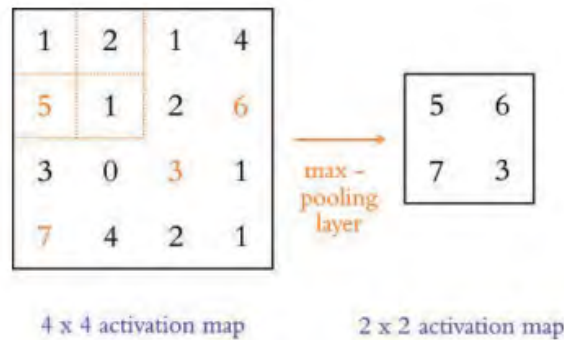


Figure 2.6: A max-pooling layer is demonstrated here, with a 4×4 activation map. It works similarly to a convolutional layer, moving from left to right and top to bottom over the matrix of values. A 2×2 filter is used, which pertains only the largest of the 4 input values (e.g., the “orange 5” in the top-left corner of the 2×2 hatch-marked area). With a 2×2 stride, the output of this max-pooling layer has one-fourth of the volume of its input, resulting in a 2×2 activation map. Figure taken from [96].

2.4.3 Normalization Layers

The purpose of normalization layers is to normalize data, leading to slight improvements in classification accuracy. These layers normalize data in local input regions, which allows them to better distinguish between neighboring regions. An example of a batch normalization can be shown in Figure 2.7

2.4.4 Fully-Connected (Dense) Layers

Fully-Connected Layers (Dense) are the highest level of every Convolutional Neural Network (CNN) architecture. They are responsible for combining features detected from image regions and transforming them into an N -dimensional vector, where N is the number of classes. Usually, more than one consecutive fully-connected layers are needed. It can be said that the fully-connected layers are the actual network, while the previous layers (convolutional, pooling, normalization) are feature extractors. These layers are not spatially located, while the last one is linked to a loss function, which is used to calculate the classification error and update the weights during the back propagation process.

2.4.5 Classification with CNNs

Having discussed the fundamentals of convolutional layers, we illustrate visual examples in Figs. 2.8 and 2.9, where the pipeline of classification using CNNs is demonstrated. Moreover, we may now review their essential features:

- deep convolutional neural networks are enabled to learn to recognize features re-

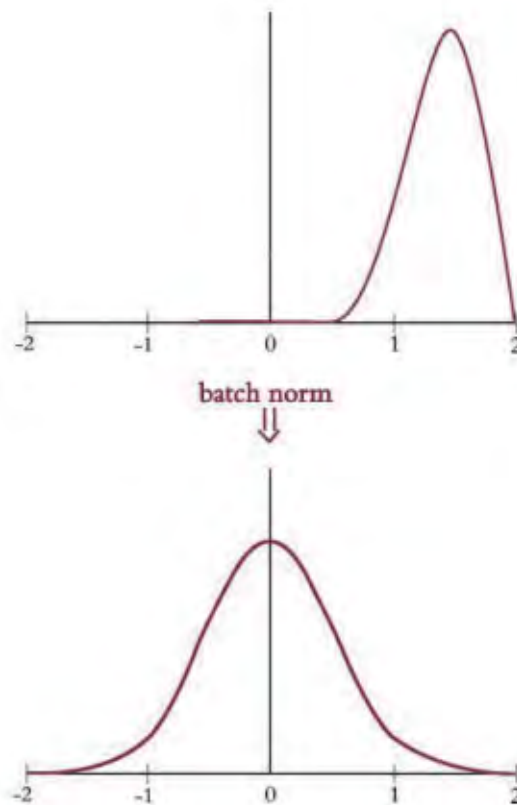


Figure 2.7: Batch normalization modifies the distribution of the activations produced by a layer of neurons to be closer to a standard normal distribution. Figure taken from [96]

regardless of their position, i.e., a single kernel can detect its related feature in any part of the input data;

- they stay devoted to the 2D arrangement of images, permitting characteristics to be distinguished inside their spatial setting;
- the amount of parameters needed to model image data is significantly decreased, resulting in a higher level of computational efficiency;
- they are able to carry out complex machine vision tasks (e.g., image classification) with greater precision compared to “shallow” approaches.

2.5 Recurrent Neural Networks

A recurrent neural network (RNN) [171] is a type of artificial neural network that differs from the uni-directional feed-forward neural network in that it allows the flow of information between its layers to occur in both directions. This bi-directional nature enables the output from certain nodes to influence subsequent input to those same nodes. RNNs are

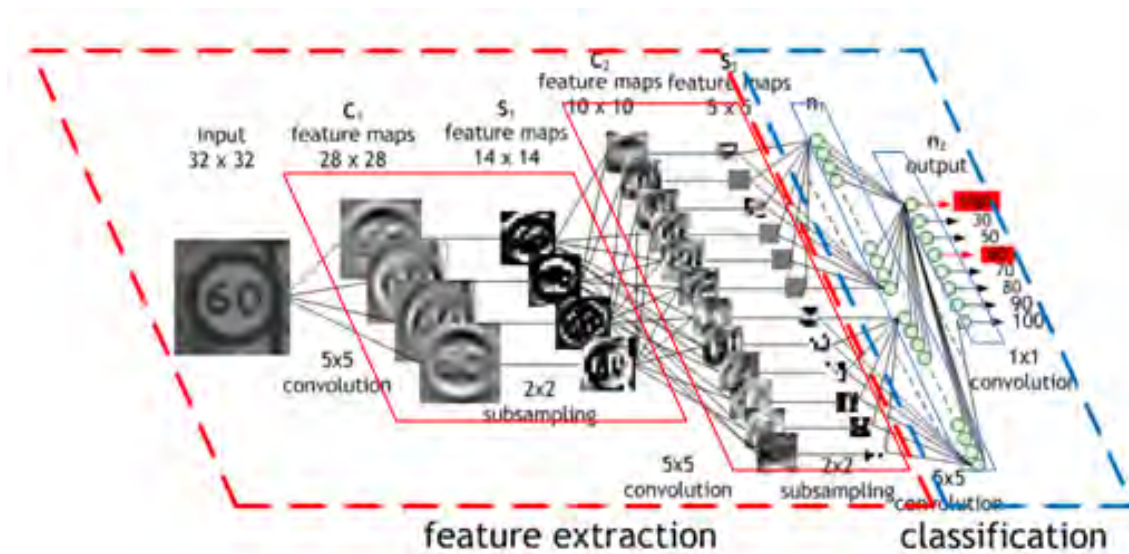


Figure 2.8: A visual example of how a CNN works. Firstly, A given input image of a traffic sign is filtered by 45×5 convolutional kernels. This way 4 feature maps are created and are then subsampled by the process of max pooling. The next layer applies 105×5 convolutional kernels to these subsampled images and the features maps are again subsampled by max pooling. The final layer is a fully connected layer where all generated features are combined and used for classification. Figure taken from [143].

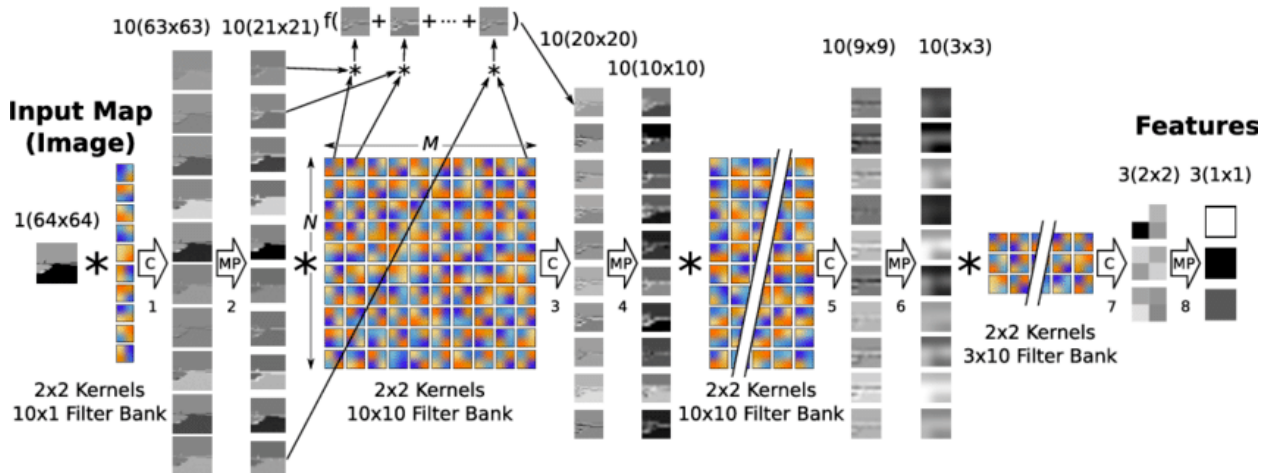


Figure 2.9: Another visual example of how a CNN works. The network's 8 layers alternate between convolutional (C) and max-pooling (MP). The first layer (C) filters the input 64×64 px image with a bank of 10×10 filters producing 10 maps of size 63×63 , that are down-sampled to 21×21 px by the second (MP) layer. 2. The procedure continues this way until the input image is transformed to just 3 features that are used for classification. Figure taken from [94].

particularly well-suited for tasks involving the processing of arbitrary sequences of inputs, thanks to their ability to utilize internal state (memory). Examples of such tasks include

unsegmented, connected handwriting recognition and speech recognition. The term “recurrent neural network” specifically refers to networks with an infinite impulse response, while “convolutional neural network” is used for networks with a finite impulse response. Both types of networks exhibit temporal dynamic behavior. It is worth noting that a finite impulse recurrent network can be transformed into a strictly feed-forward neural network by unrolling it and removing its cyclic structure, whereas an infinite impulse recurrent network cannot undergo this transformation.

Both infinite-impulse and finite-impulse networks have the capability to incorporate additional stored states and storage that is directly controlled by the network. This storage can be substituted with another network or graph that includes time delays or feedback loops. These controlled states are known as gated state or gated memory, and are integral components of long short-term memory networks (LSTMs) and gated recurrent units. Another term for this is Feed-forward Neural Network (FNN). Recurrent neural networks are theoretically capable of running arbitrary programs to process arbitrary sequences of inputs, making them Turing complete.

Long short-term memory (LSTM) is a deep learning system designed to overcome the problem of vanishing gradients. To enhance LSTM, recurrent gates known as “forget gates” are typically incorporated. This allows LSTM to prevent the vanishing or exploding of back-propagated errors. Instead, errors can flow backwards through virtual layers that are unfolded in space, enabling LSTM to learn tasks that involve remembering events from thousands or even millions of discrete time steps in the past. Problem-specific LSTM-like structures can be evolved as well. LSTM is effective even when there are long delays between significant events and can handle signals that contain both low and high frequency components. Many applications utilize stacks of LSTM recurrent neural networks (RNNs) and train them using Connectionist Temporal Classification (CTC) [61]. CTC aims to find an RNN weight matrix that maximizes the probability of label sequences in a training set, given the corresponding input sequences. CTC achieves both alignment and recognition. Unlike previous models based on hidden Markov models (HMM) and similar concepts, LSTM can learn to recognize context-sensitive languages.

2.6 Generative Adversarial Networks

Generative Adversarial Networks (GANs) have emerged as a revolutionary concept in the field of artificial intelligence, particularly in the domain of computer vision. These neural networks have gained widespread attention for their remarkable ability to generate synthetic data that closely resembles real-world data. GANs consist of two main components: a generator and a discriminator, which are trained in a game-theoretic manner, often referred to as a minimax game. This adversarial training process pushes the generator to create increasingly realistic data, while the discriminator becomes more proficient at distinguishing between real and fake data. As a result, GANs have shown exceptional promise in various computer vision applications.

1. **Image Synthesis:** One of the most prominent applications of GANs in computer vision is image synthesis. GANs can generate highly realistic images that range from art and design to photorealistic scenes [20]. The generator learns to produce images that are indistinguishable from those found in real datasets. This capability is valuable in various industries, including entertainment, fashion, and even video game development.
2. **Data Augmentation:** GANs can be used to augment datasets for training computer vision models [48]. By generating additional data points, GANs help address the issue of data scarcity, enhancing the performance and robustness of deep learning algorithms. This is particularly useful when labeled data is limited or expensive to acquire.
3. **Super-Resolution:** GANs have proven effective in super-resolution tasks, where low-resolution images are upscaled to higher resolutions [21]. These networks can restore fine details, making them essential in applications such as medical imaging, satellite imagery analysis, and enhancing the quality of consumer-grade images and videos.
4. **Image-to-Image Translation:** GANs excel in tasks involving image-to-image translation [216, 47]. They can convert images from one domain to another, such as turning day scenes into night or transforming sketches into realistic images. This is valuable in fields like architecture, urban planning, and art.
5. **Style Transfer:** GANs can be used for artistic style transfer, allowing users to apply the style of one image to another [13]. This technique has gained popularity in the creation of unique visual content and artistic expression.
6. **Anomaly Detection:** GANs can also be employed for anomaly detection in computer vision. By learning the characteristics of normal data, GANs can identify deviations or anomalies in real-time, making them useful for security and surveillance applications.

Although GANs have shown significant potential in computer vision, they also pose challenges, such as mode collapse, training instability, and ethical concerns related to deep-fake generation [136]. Future research in this area aims to mitigate these issues and explore novel applications, including 3D object generation, semantic segmentation, and more.

At its pinnacle, a GAN consists of two networks engaged in a competitive interplay. Illustrated in Fig. 2.10, one of the networks is designated as the generator, responsible for crafting fabricated images, while the other, the discriminator, works to differentiate the artificial creations from authentic images. The transition to a slightly more complex depiction is presented through schematic diagrams in Fig. 2.11, where the generator and discriminator processes are highlighted on the left and right, respectively, albeit in a highly simplified manner.

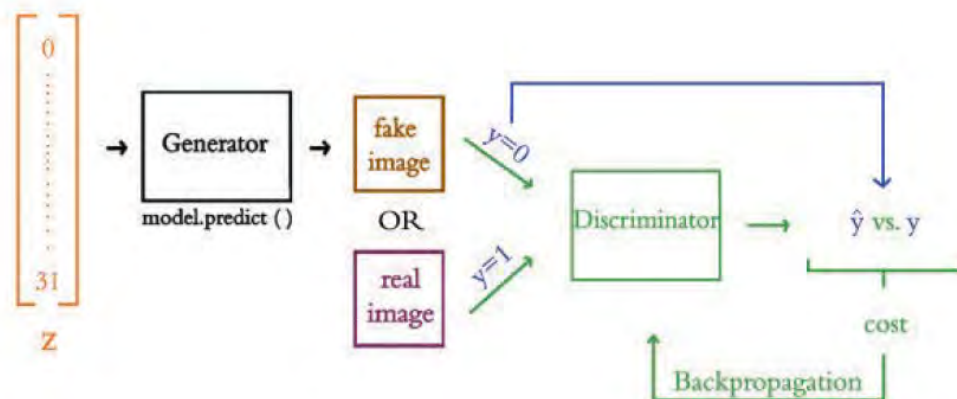


Figure 2.10: Visual explanation of the way a discriminator of a GAN works. Figure taken from [96].

During numerous iterations of training, both entities progressively hone their skills; the generator in creating increasingly realistic forgeries and the discriminator in enhancing its ability to identify the falsified images. As this training phase unfolds, a competitive dynamic emerges where both networks strive to surpass each other, fostering greater specialization in their respective roles. This iterative adversarial engagement potentially leads to a stage where the generator is capable of creating images so lifelike that they can deceive not only the discriminator but also human viewers.

Specifically, training a GAN consists of two opposing processes:

- **Discriminator Training:** As outlined in Fig. 2.10, during this phase, the generator creates synthetic images—engaging solely in inference—while the discriminator works to distinguish these fabricated images from genuine ones.
- **Generator Training:** Illustrated in Fig. 2.11, this stage involves the discriminator evaluating the false images created by the generator. In this scenario, only the discriminator is involved in inference while the generator utilizes the feedback to improve—specifically, to enhance its ability to deceive the discriminator into mislabeling the fake images as real ones.

2.7 Implementation

In this section we aim to present in brief the programming languages, frameworks and libraries that have been used for the implementation of the methodologies that will be presented in the following chapters and also provide implementation hardware and software details.

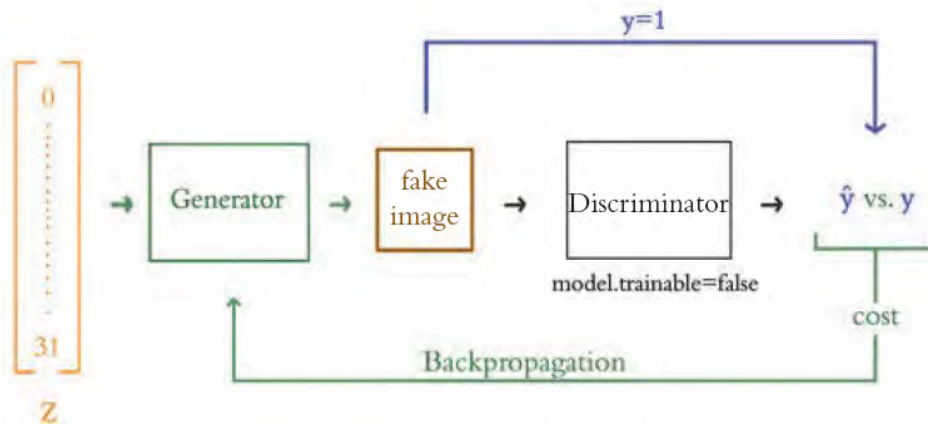


Figure 2.11: Visual explanation of the way a generator of a GAN works. Figure taken from [96].

2.7.1 Programming Languages, Frameworks and Libraries

Python [156] is a programming language that is easy to learn and has powerful capabilities. It has efficient data structures and a straightforward approach to object-oriented programming. Python's syntax is graceful and its dynamic typing, combined with its interpreted nature, make it a great choice for scripting and fast application development in a variety of areas on multiple platforms. In the context of this work, Python has been used for the implementation of all techniques that will be presented in Chapters 3 to 7.

TensorFlow [5] is a powerful machine-learning system that can be used in a variety of environments. It is based on dataflow graphs with mutable state, which can be mapped to different machines in a cluster and to CPUs, GPUs, and other devices. It is used for a range of applications, particularly for training and inference with deep neural networks. These applications include speech recognition, computer vision, robotics, information retrieval, and natural language processing.

Keras [36] is a Python-based deep learning API that runs on the TensorFlow machine learning platform. It was designed to facilitate rapid experimentation. Quickly going from concept to result is essential for successful research. Keras is simple so as to reduce the cognitive load on developers, allowing them to focus on the most important aspects of the problem, flexible, as it follows the principle of gradually increasing complexity and powerful, by providing robust performance and scalability. In the context of this work, Keras/Tensorflow have been used for the implementation of the deep neural network architectures.

Open Source Computer Vision Library (OpenCV) [18] is a free, open-source software library for computer vision and machine learning and contains over 2500 optimized algorithms, including a wide range of classic and state-of-the-art computer vision and machine learning algorithms, which can be used for a variety of tasks, such as face detection/recognition, object recognition, human action classification, tracking, image retrieval,

etc. In the context of this work, OpenCV has been used for the feature extraction of the hand-crafted features in [4].

Scikit-learn [149] is a Python library that provides a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. It is designed to make machine learning accessible to non-specialists, using a high-level language. In the context of this work, Scikit-learn has been used for splitting the datasets and training an svm algorithm in [4] .

Numpy [64] is a Python library that is essential for scientific computing. It offers a multi-dimensional array object, a variety of derived objects (like masked arrays and matrices), and a range of functions for quickly performing operations on arrays, such as mathematical, logical, shape manipulation, sorting, selection, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation, and much more. In the context of this work, Numpy has been used for manipulation of the dataset's, for creating pseudo-colour images[3] etc..

SciPy [187] is a library of numerical routines for the Python programming language that provides the essential components for constructing and solving scientific problems. It includes algorithms for optimization, integration, interpolation, eigenvalue problems, algebraic equations, differential equations, and many other types of problems; it also offers specialized data structures, such as sparse matrices and k-dimensional trees. SciPy is built on top of NumPy and consists the basis for higher-level scientific libraries, such as scikit-learn. In the context of this work, SciPy has been used for interpolating the input to our algorithms.

Matplotlib [73] is a Python library for creating graphs of high quality, written to take advantage of NumPy . It is able to produce plots, histograms, power spectra, bar charts, error charts, and scatterplots. In the context of this work, Matplotlib has been used for visualizing the skeleton of the dataset and plotting the confusion matrixes. **Pandas** [131] is a Python package that offers fast, flexible, and expressive data structures designed to make it easy and intuitive to work with “relational” or “labeled” data. In the context of this work, Pandas has been used for handling the datasets.

2.7.2 Implementation Details

Experiments were conducted on a personal workstation with an Intel™i7 4770 4-core processor running at 3.40 GHz and 32GB RAM, using a NVIDIA™Geforce RTX 3070 GPU with 8 GB VRAM and Ubuntu 20.04 (64 bit). The deep architecture was implemented in Python, using Keras 2.4.3 [36] with the Tensorflow 2.5 [5] backend. All data pre-processing and processing steps were implemented in Python 3.9 using NumPy and SciPy.

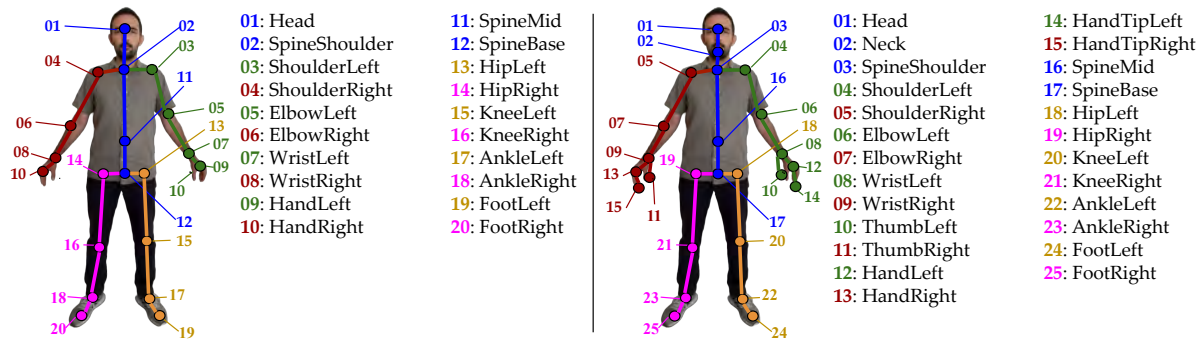


Figure 2.12: A human body pose with the 20 and 25 skeletal joints that are extracted using the Microsoft Kinect v1 (left) and v2 (right) cameras. Joints have been divided into subsets, each corresponding to one of the five main body parts, i.e., torso (blue), left hand (green), right hand (red), left leg (orange) and right leg (magenta). For illustrative purposes and also to facilitate comparisons between the two different versions, body parts have been colored using the same colors. Numbering follows the Kinect SDK in both cases, therefore there exist several differences between the two versions.

2.8 Skeletal Data

In the following sections, we will use 3D trajectories of human skeletons as our input. In 3D HAR problems, subjects perform actions in space and over time. We consider skeleton representations as sets of 3D joints. We use skeleton data that have been captured using the Microsoft Kinect RGB/depth camera.¹ A human skeleton comprises of 20 (Kinect v1) or 25 (Kinect v2) joints, organized as a graph; each node corresponds to a body part such as arms, legs, head, neck etc., while edges follow the body structure, appropriately connecting pairs of joints. In Fig. 2.12 we illustrate a skeleton extracted using Kinect v1 and v2. Note that joints are shown as being grouped; each group corresponds to a body part, i.e., an arm, a leg or the torso. In the context of this work, an activity is considered to be a temporal sequence of 3D skeleton representations. For the sake of explanation, in Fig. 2.13 and in Fig. 2.14, we illustrate an example of an action of the PKU-MMD DATASET and the NTU dataset.



Figure 2.13: A sequence of an actor performing the activity *wearing a hat*. Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the NTU dataset [172] and have been trimmed for illustration purposes.

¹<https://developer.microsoft.com/en-us/windows/kinect>



Figure 2.14: A sequence of an actor performing the activity *salute*. Extracted human skeleton 3D joints using the Kinect SDK have been overlaid. Frames have been taken from the PKU-MMD dataset [125] and have been trimmed for illustration purposes.

2.9 Experimental Protocols

In the experiments that will be presented in the following Chapters, we will follow typical experimental setups that are encountered in research works that are evaluated using multi-view datasets, such as [125, 172]. Specifically,

- a. *per camera position* (single view), where training and test sets derive from the same camera (viewpoint), e.g., both deriving from middle camera; this is a typical evaluation approach, that aims to evaluate the performance of the trained model in the *ideal* case where it is trained and tested under the same viewpoint and using the same actors;
- b. *cross view* experiments, where training and test sets derive from different cameras (viewpoints), e.g., training deriving from middle camera, testing deriving from left camera; this evaluation approach aims to evaluate whether the trained model is robust to viewpoint changes;
- c. *cross subject* experiments, wherein subjects are split into training and testing groups, i.e., each subject appears exactly into one of these groups; this evaluation approach aims to evaluate whether the trained model is able to generalize with unseen actors.

2.10 Performance Metrics

In this section our goal is to define the metrics that will be used in the following chapters, in order to evaluate the performance of the presented HAR approaches. Performance metrics play a vital role in assessing the effectiveness of machine learning models across various applications. These metrics provide valuable insights into the model's behavior, allowing practitioners to make informed decisions. To begin with, let us consider a binary classification problem, having two classes, namely “positive” and “negative.” We may firstly define the following:

- *true positives* (TP): the number of positive observations correctly predicted as positive;

Table 2.1: Definition of a confusion matrix for a binary classification problem.

Predicted Class	Actual Class	
	Positive	Negative
	Positive	Negative
	Positive	Negative
	Negative	Positive
	TP	FP
	FN	TN

- *false positives* (FP): the number of negative observations incorrectly predicted as positive;
- *true negatives* (TN): the number of negative observations correctly predicted as negative;
- *false negatives* (FN): the number of positive observations incorrectly predicted as negative.

We are now able to define the following metrics:

- *precision* (P) is the ratio of correctly predicted positive observations to the total predicted positives, i.e.,

$$P = \frac{TP}{TP + FP} \quad (2.3)$$

- *recall* (R), often referred to as “sensitivity” or “true positive rate,” is the ratio of correctly predicted positive observations to the total actual positives, i.e.,

$$R = \frac{TP}{TP + FN} \quad (2.4)$$

- *F1-score* ($F1$) is the harmonic mean of precision and recall, providing a balance between them, i.e.,

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (2.5)$$

- *Accuracy* ($Acc.$) is the ratio of correctly predicted observations to the total observations, i.e.,

$$Acc. = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.6)$$

A confusion matrix is a table that describes the performance of a classification model. It is often used to calculate various metrics, including True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN). Specifically, for a binary classification problem, a confusion matrix is defined as the one in Table 2.1.

Below are some common scenarios where performance metrics are employed and how they contribute to model evaluation.

Firstly, in classification problems, where the goal is to categorize instances into predefined classes, performance metrics are extensively used. The confusion matrix and associated metrics such as P , R , $F1$, $Acc.$ and others, not mentioned herein may place a crucial role in understanding the model's behavior with respect to different classes. For example, in medical diagnostics, P and R are crucial when evaluating a model's ability to correctly identify patients with a specific condition (true positives) while minimizing the risk of FP and FN.

Moreover, performance metrics become particularly important when dealing with imbalanced datasets, where one class significantly outnumbers the others. In such cases, $Acc.$ alone may be misleading. In that case, $F1$ score, which considers both P and R , provides a more balanced assessment of a model's performance in imbalanced scenarios. They may also serve as objective functions in hyper-parameter tuning. Specifically, during model training, often optimization of models is based on specific metrics to achieve the desired balance between e.g., P and R or other relevant criteria. Of course, when comparing multiple models or variations of a model, performance metrics provide a quantitative basis for comparison and may aid in selecting the model that best aligns with the specific goals of the task. Ultimately, performance metrics guide decision-making processes in deploying or refining machine learning models. They offer a comprehensive understanding of a model's strengths and weaknesses, providing help in making specific choices regarding the model's suitability for a given application.

3. PSEUDO-COLORED IMAGE REPRESENTATION FOR HAR

3.1 Introduction

The problem of understanding human actions based on visual information has been increasingly attracting the interest of the research community in the fields of computer vision and pattern recognition. Several approaches adopting pseudo-colored image representations of skeletal trajectories and working with convolutional neural networks (CNNs) have been proposed. In this chapter, we present the findings of a method for comprehending human activities in video sequences. Our results demonstrate the potential of our approach, since classification accuracy was satisfactory even in cross-view and cross-subject experiments.

3.2 Related work

In this section, we take a look at the most recent research works on Human Activity Recognition (HAR) that uses deep learning, focusing on those that are based on skeletal motion encoded through image-based representations, similar to the approach we follow in this work.

Huynh-The et al. [72] 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. [152] 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 Wang et al. [194], a representation called “joint trajectory maps” (JTM) was proposed, wherein skeleton data sequences are represented by three 2D images. The motion dynamics are captured as the image’s texture and color. Specifically, motion direction is reflected as hue in the colored image, different body parts are represented by multiple color maps and last but not least, the motion magnitude of joints is reflected by the image’s saturation and brightness. Another similar approach to representing skeletal information as color texture images is the method of “joint distance maps” (JDM), proposed by Li et al. [110]. 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. [109] presented a deep learning model that preserves spatial and temporal features, taking advantage of both LSTM and CNN architectures. Inspired by [152] and [194], 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. [126] 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.

An approach addressing the problem of view invariance was presented in the work of Liu et al. [124] where a transformation creating a 5D representation of joints has been proposed, 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. Similarly, Yang et al. [207] 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. Hou et al. [66] introduced an image-based representation called "joint skeleton spectra." The joint distribution maps are projected onto three Cartesian planes, reflecting the temporal variation of a skeleton sequence to hue values. Finally, Ke et al. [87] 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.

Chen et al. [33] encoded spatial-temporal information from skeleton sequences into color texture images created upon by projecting the 3D points to the three orthogonal, which are referred to as "Temporal Pyramid Skeleton Motion Maps" (TPSMMs). The TPSMMs had been designed so as to not only capture short temporal information but also embed long dynamic information over the period of action. This was achieved in three ways:

different types of visualization of joints, calculation of absolute differences between joints in consecutive frames and pseudo-color coding. They constructed a hierarchical structure composed of six TPSMMs and accordingly for classification they used six CNNs, one per each TPSMM and adopted a late fusion approach. Silva et al. [173] mapped the temporal and spatial joints dynamics into a color image-based representation, wherein, the position of the joints in the final image is clustered into groups. In order to verify whether the sequence of the joints in the final image representation can influence the performance of the model, they conducted two experiments: in the former, they changed the order of the grouped joints in the sequence, while in the latter, the joints were randomly ordered.

Tasnim et al. [179] proposed a spatio-temporal image formation (STIF) technique of 3D skeleton joints. To generate the spatio-temporal image, they mapped all the 20 joints in a frame with the same color, using the jet color map and then changed the colors as time was passing. They created the STIF, by connecting lines between joints in adjacent frames, subsequently. For feature extraction they conducted transfer learning, using well-known deep architectures. For classification they compared three fusion methods. Verma et al. [184] created skeleton intensity images, for 3 views (top, front and side) using a proposed algorithm from skeleton data. Note that they used a combination of a CNN and an LSTM; the former extracted spatial feature from an image, while the latter was used to temporally refine the feature extraction procedure by processing a sequence of inputs. Finally, Caetano et al. [23] introduced a novel skeleton image representation, named “SkeleMotion.” The proposed approach encodes the temporal dynamics by explicitly computing the magnitude and orientation values of the skeleton joints. Different temporal scales were employed to compute motion values to aggregate more temporal dynamics to the representation making it able to capture long-range joint interactions involved in actions as well as filtering noisy motion values. Note that contrary to typical contemporary approaches, they opted for a shallow neural network, instead of a deep one. This was made possibly by using compact video representation and allowed fast training times.

3.3 Methodology

In this chapter, we propose a novel visual representation of skeletal data, appropriate to be used as input to a CNN. This representation must capture and maintain the spatio-temporal properties of the skeleton trajectories, so that it can differentiate between different activities. Additionally, it must adhere to the graph structure of the skeleton representation. At the following we shall refer to this representation as “pseudo-colored” image.

The pseudo-colored image aims to capture the changing distances between joints throughout the activity. These distances are then used to create a simulated RGB image. This image is based on the 3D trajectories of the skeletal joints. Let $x(n)$, $y(n)$ and $z(n)$ denote the coordinates of each of the N available joints at the n -th frame F_n of the video sequence depicting the activity. This results in a set of $3 \times N$ signals for the given video sequence. To account for the temporal variability between actions and users, a linear interpolation step is imposed, setting the duration of all videos to $N = 150$ frames. The differences be-

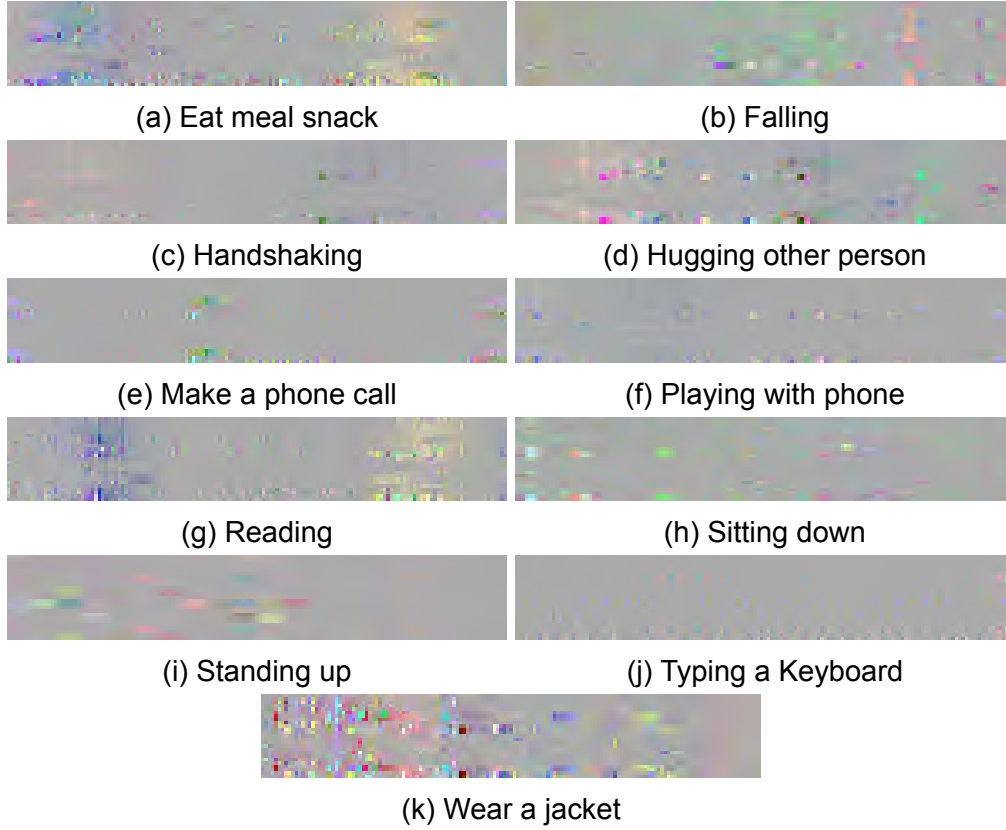


Figure 3.1: Representative pseudo-colored images of the eleven classes of the PKU-MMD Dataset.

tween consecutive frames are then calculated from each of the sequences. To create the pseudo-colored images, the x , y , z coordinates are assigned to the R, G, B color channels of the image, respectively.

The procedure for generating a pseudo-colored image $\mathbf{I}$ is as follows: Let $x_i(n)$, with i ranging from 1 to N , denote the x -position of the i -th joint in the n -th frame. The red, green and blue channels of $\mathbf{I}$ are denoted as r , g , and b , respectively. The pixel values of $r(i, n)$, $g(i, n)$, and $b(i, n)$ are calculated as:

$$\begin{aligned} r(i, n) &= x_i(n+1) - x_i(n) \\ g(i, n) &= y_i(n+1) - y_i(n) \\ b(i, n) &= z_i(n+1) - z_i(n) \end{aligned} \quad (3.1)$$

where $n = 1, \dots, T_a$ and $i = 1, \dots, N$. 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. 3.1 we illustrate several pseudo-colored images that correspond to activities.

For classification, we employ the CNN depicted in Fig.3.2. The first convolutional layer

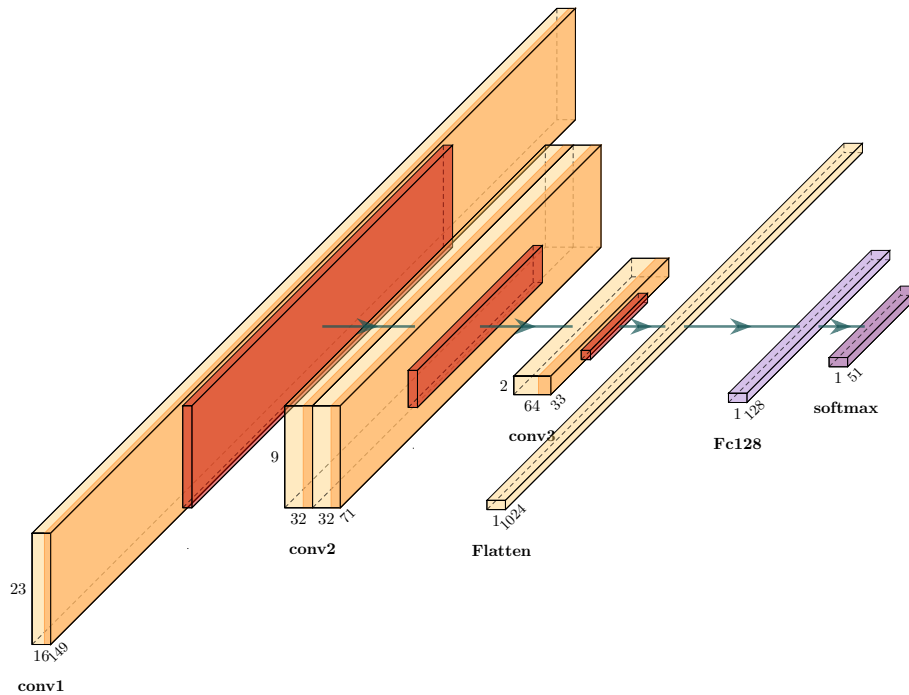


Figure 3.2: The architecture of the CNN that has been used for classification of pseudo-colored images.

filters the 25×149 input activity image with 16 kernels of size 3×3 . Subsequently, the first pooling layer applies “max-pooling” to perform 2×2 subsampling. The second convolutional layer filters the 11×73 resulting image with 32 kernels of size 3×3 . The third convolutional layer filters the 11×73 resulting image with 32 kernels of size 3×3 . The second pooling layer applies “max-pooling” to perform 2×2 subsampling. The fourth convolutional layer filters the 4×35 resulting image with 32 kernels of size 3×3 . The third pool layer applies “max-pooling” to perform 2×2 subsampling. The flatten layer then transforms the output image of size 1×16 of the last pooling to a vector, which is then used as input to a dense layer with dropout [176]. Finally, a second dense layer produces the output of the network.

3.4 Experimental Results

We evaluated the proposed approach using the four datasets that have been described in section 1.5. Specifically, in case of the PKU-MMD dataset we performed two types of experiments. Firstly we aimed to assess whether the proposed approach is suitable for use in an ambient assistive living environment, so we selected 11 classes of PKU-MMD that are most closely related to activities of daily living (ADLs) or events that should be monitored in such a use case. These classes are: *eating meals or snacks*, *falling*, *handshaking*, *hugging another person*, *making or answering a phone call*, *playing with a*

Table 3.1: Results of the evaluation of the proposed system, when tested on 11 classes of the PKU-MMD dataset, using the middle, left and right camera angles (M, L and R, respectively). Best result denoted in bold.

Experiment	Train	Test	Acc	F1	Act(Acc)
Single-view	M	M	0.88	0.87	0.82
	L	L	0.85	0.84	0.85
	R	R	0.84	0.83	0.75
Cross-view	M	L	0.71	0.71	0.56
	M	R	0.69	0.68	0.64
	L	M	0.72	0.71	0.61
	L	R	0.53	0.53	0.40
	R	M	0.70	0.70	0.56
	R	L	0.56	0.57	0.35
	M,L	R	0.76	0.76	-
	M,R	L	0.66	0.65	-
	L,R	M	0.81	0.81	-
	Cross-subject	M,L,R	0.86	0.84	-

phone or tablet, reading, sitting down, standing up, typing on a keyboard and putting on a jacket. We also conducted experiments with the full PKU-MMD dataset, which has 51 classes. We only used the available skeletal data, disregarding RGB, depth and infrared information. In case of the NTU-RGB+D dataset we firstly we aimed to assess whether the proposed approach is suitable for use in an medical environment, so we performed experiments with 12 classes of the dataset that have been described as “medical” by the dataset authors. These classes are: *sneeze/cough*, *staggering*, *falling down*, *headache*, *chest pain*, *back pain*, *neck pain*, *nausea/vomiting*, *fan self*, *yawn*, *stretch oneself* and *blow nose*. In cases of SYSU-3D and UTKinect-Action3D dataset we performed experiments with full datasets.

For PKU-MMD and NTU-RGB+D, since they are multi-view datasets we present results for the following cases that have been discussed in Section 2.9: a) *Single-view*: both training and testing sets derived from the same camera; b) *Cross-view*: different viewpoints were used for training and testing; and c) *Cross-subject*: actors were split in training and testing groups. In case of SYSU-3D and UTKinect-Action3D, since they are single-view datasets we only performed experiments for the single-view case. In all cases we measured classification accuracy and F1-score. Results are presented in Tables 3.1-3.5 and the corresponding confusion matrices are depicted in Figs 3.3 -3.5. It may be observed that compared to the work of Mathe et al. [135], it demonstrated superior results, both in single-view and in cross-view cases.

Table 3.2: Results of the evaluation of the proposed system, when tested on the PKU-MMD dataset, using the middle, left and right camera angles (M, L and R). Best result denoted in bold.

Experiment	Train	Test	Acc	F1	Acc[135]
Single-view	M	M	0.64	0.64	0.73
	L	L	0.68	0.67	0.55
	R	R	0.49	0.47	0.73
Cross-view	M	L	0.62	0.61	0.29
	M	R	0.61	0.61	0.28
	L	M	0.62	0.61	0.25
	L	R	0.58	0.59	0.11
	R	M	0.61	0.60	0.29
	R	L	0.48	0.47	0.12
	M,L	R	0.66	0.65	-
	M,R	L	0.65	0.65	-
	L,R	M	0.70	0.70	-
	Cross-subject	M,L,R	0.57	0.55	-

3.5 Conclusions

In this Chapter, our objective was to propose a novel method for recognizing human activities using human motion. We utilized 3D skeletal data as the input for our approach. However, we encountered a limitation concerning the input layer of convolutional neural networks, which require still images rendering the temporal sequences of the skeletal data unusable in this specific form. To address this issue, we introduced a novel 2D representation of skeletal motion. Our representation was designed to capture and preserve the spatio-temporal characteristics of the skeleton trajectories, while adhering to the graph-based skeleton representation. It was based on the inter-joint distances of skeleton joints and followed the order of skeleton joints based on their ids. We demonstrated that this representation can effectively be applied to the task of human activity recognition and can differentiate between different activities. Furthermore, we showed that this representation outperformed previous state-of-the-art approaches in both single-view and cross-view scenarios.

Table 3.3: Results of the evaluation of the proposed system, when tested on eleven classes of the NTU-Medical dataset, using the middle, left and right camera angles (M, L and R).

Experiment	Train	Test	Acc	F1
Single-view	M	M	0.59	0.58
	L	L	0.60	0.59
	R	R	0.61	0.59
Cross-view	M	L	0.64	0.63
	M	R	0.55	0.55
	L	M	0.49	0.48
	L	R	0.58	0.58
	R	M	0.52	0.53
	R	L	0.64	0.64
	M,L	R	0.57	0.57
	M,R	L	0.58	0.58
	L,R	M	0.60	0.60
Cross-subject	M,L,R	M,L,R	0.62	0.63

Table 3.4: Results of the evaluation of the proposed system, when tested on the NTU dataset, using the middle, left and right camera angles (M, L and R).

Experiment	Train	Test	Acc	F1
Single-view	M	M	0.58	0.57
	L	L	0.60	0.60
	R	R	0.59	0.58
Cross-view	M	L	0.46	0.46
	M	R	0.46	0.46
	L	M	0.37	0.38
	L	R	0.41	0.41
	R	M	0.50	0.50
	R	L	0.43	0.44
	M,L	R	0.45	0.46
	M,R	L	0.52	0.52
	L,R	M	0.43	0.42
Cross-subject	M,L,R	M,L,R	0.49	0.49

Table 3.5: Results of the evaluation of the proposed system, when tested on the SYSU-3D Dataset and the UTKinect-Action3D dataset, using the middle camera angle M.

Experiment	Train	Test	Acc	F1
UT	M	M	0.65	0.58
SYSU	M	M	0.44	0.42

Human Activity Recognition Under Occlusion

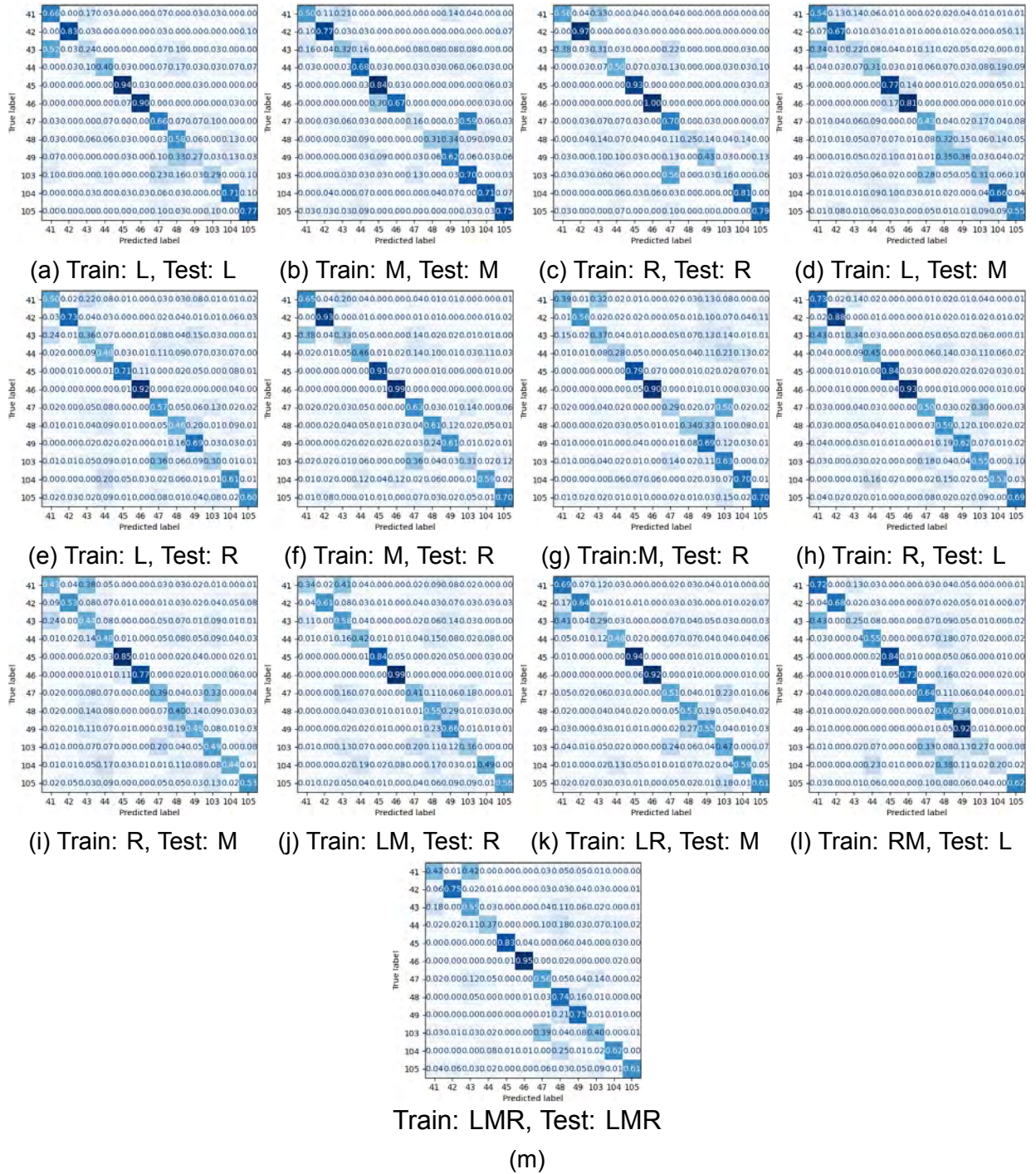


Figure 3.3: Normalized confusion matrices for classification of the medical classes of the NTU-RGB+D dataset. M, L and R denote middle, left and right camera angles, respectively.

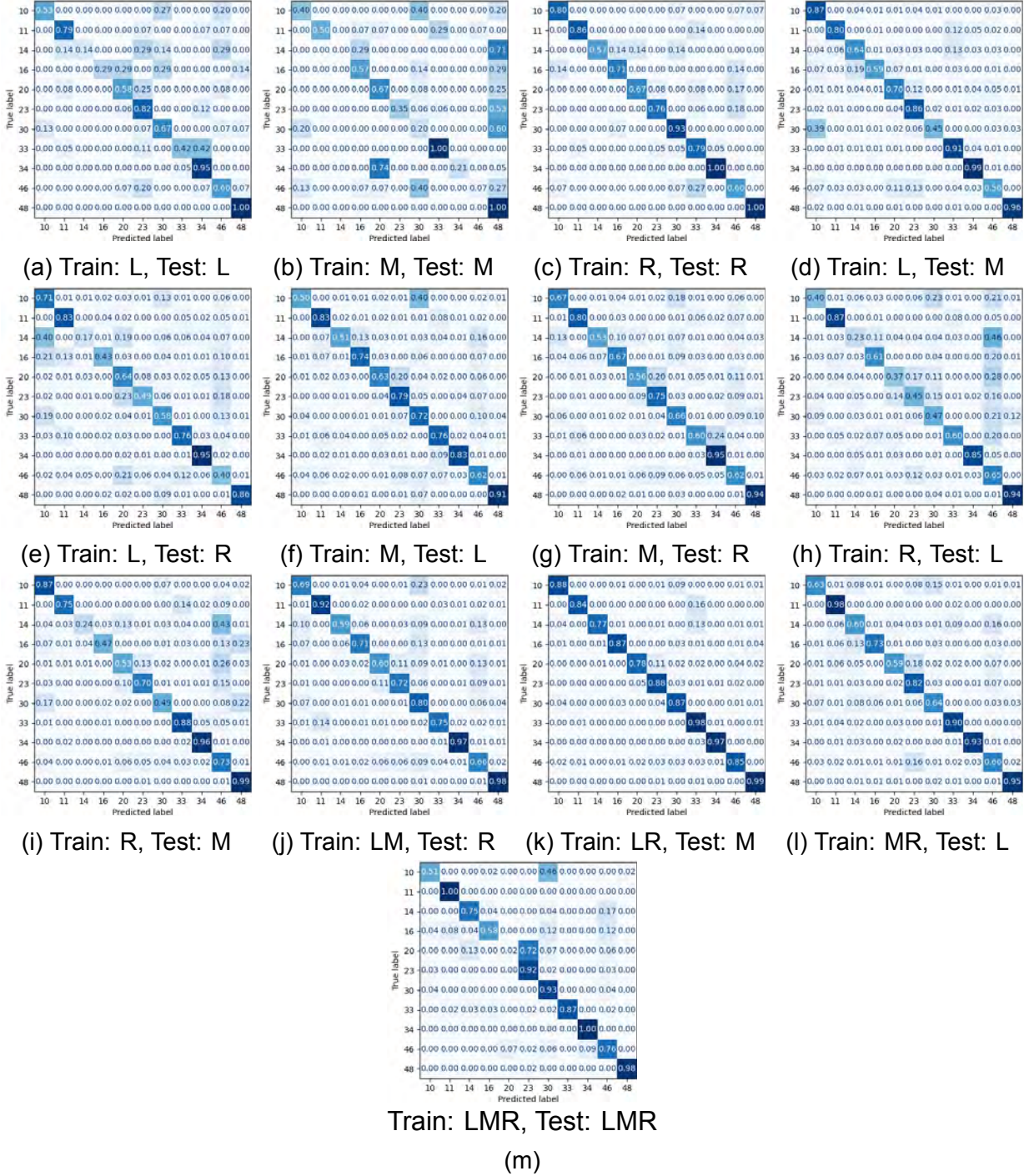


Figure 3.4: Normalized confusion matrices for classification of the ADL classes of the PKU-MMD dataset. M, L and R denote middle, left and right camera angles, respectively.

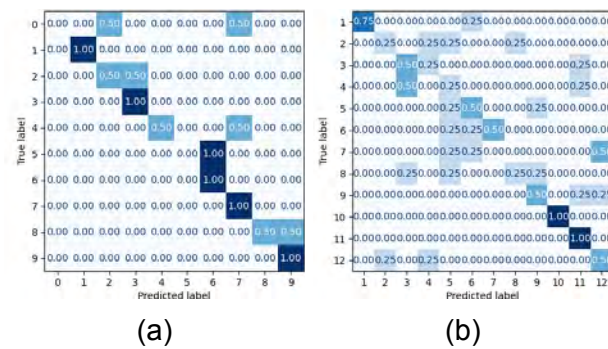


Figure 3.5: Normalized confusion matrices for the classification of the (a) UTKinect-Action3D and the (b) SYSU-3D 3D Human-Object Interaction Dataset.

4. FUSION OF HANDCRAFTED AND CONTEXTUAL FEATURES FOR HAR

4.1 Introduction

As we have already made clear in this Thesis, human activity recognition consists probably one of the most challenging problems in the field of computer vision. At this point we should note that before the rise of deep learning, early HAR approaches [166, 41] were based on the extraction of handcrafted features. The latter are typically algorithmic pipelines, manually designed to extract image properties using the information available within the image, in a deterministic way. Design takes place by keeping in mind that extracted features should be robust to variances such as illumination, viewpoint changes etc. Then, these features were used to train traditional machine learning algorithms such as support vector machines. Such approaches have several limitations including limiting robustness to viewpoint changes or significant drop of performance when trained for a large number of actions. Especially the first limitation is crucial, when dealing with real-life scenarios. Of course, “deep architectures” [89], contrary to traditional approaches, do not require a prior handcrafted feature extraction step, since features are “learnt” by the network, during the training process.

The aim of this Chapter is to demonstrate that deep architectures can be used to supplement handcrafted features and improve the performance of classification. To do this, we use a Convolutional Neural Network (CNN) that has been pre-trained to classify contextual classes in a human activity recognition problem as a visual feature extractor for the similar, yet distinct task of arm gesture recognition.

4.2 Related work

During the last few years, several research works have experimented with fusion of handcrafted features with learnt features that are extracted using deep neural network architectures. Nanni et al.[140] proposed the use of a trained deep CNN as a feature extractor and mixed its output with handcrafted features. More specifically three substructures were proposed. Firstly they remapped the output of a CNN so as to solve a different problem than the one it was trained. Classification was performed with an SVM. Secondly, they fed an SVM with the output of the last dense layer of a CNN as a feature vector. Lastly, they fused the output of several layers and also fed an SVM. They also used several methods for extracting non-handcrafted features and a wide range of state-of-the-art algorithms for the handcrafted ones. They applied the method at several image classification problems. Wu et al.[198] proposed a model for person re-identification called “Feature Fusion Net.” This model combines color histogram features and texture features with the last pooling layer of a CNN so as to adapt the weights of the CNN by taking into consideration the handcrafted features. Kashif et al. [83] proposed a technique that showed improved per-

formance when handcrafted features were combined with raw data in order to be used as an input in a CNN model. They used the aforementioned features to detect tumor cells in histology images. Another approach, proposed by Egede et al. [46], combined handcrafted features with deep-learned features in a regression problem for automatic pain estimation. Nguyen et al. [142] used deep image features extracted by a CNN with local binary pattern handcrafted features in a presentation attack detection in face recognition. Finally, Giannakopoulos et al. [52] applied a methodology similar to the one that will be presented in this Chapter, to solve a soundscape classification problem, combining statistical handcrafted short-term audio features with a CNN, trained on spectrogram representations of audio signals.

4.3 Proposed Methodology

4.3.1 Handcrafted Features

We begin by presenting the handcrafted features that will be used. They have been proposed by Paraskevopoulos et al. [145] and have been partially inspired by a set of features that have been proposed by Sheng [170], who extended the features of Rubine [160] from the 2D to the 3D space.

We use 3D skeletal data extracted by the Microsoft Kinect sensor and extract features based on the spatial relation of joints to their parent and child joints, over time. More specifically, for a given joint J we use its child and parent joints J_c and J_p , accordingly. Features are extracted from video sequences. Let F_i , $i = 1, 2, \dots, N$ denote a given video frame and $\mathbf{v}_i^{(J)} = (v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)})$ a vector corresponding to the 3D coordinates of J at frame F_i . Also, let $\mathcal{V}^{(J)}$ denote the set of all $\mathbf{v}_i^{(J)}$, $B(\mathcal{V}^{(J)})$ the 3D bounding box of $\mathcal{V}^{(J)}$, by $a_{B(\mathcal{V}^{(J)})}$ and $b_{B(\mathcal{V}^{(J)})}$ the two different lengths of its sides. Extracted features are depicted in Table 4.1.

4.3.2 Context-aware deep Feature Extraction using a CNN

We use the visual representation of skeletal information, which has been presented in Chapter 3, in order to create images so that they will be used as input to a CNN. Specifically, we create pseudo-colored images which aim at capturing inter-joint distances during an action. We remind that for a given joint J_i , $\mathbf{v}_i^{(J)} = (v_{x,i}^{(J)}, v_{y,i}^{(J)}, v_{z,i}^{(J)})$ is the aforementioned vector capturing its 3D location at frame i , i.e., its trajectory. The 3D trajectories of skeletal joints are used as input for the construction of the representation. Each joint corresponds to a row of the constructed images, while, its x , y , z coordinates correspond to R, G, B color channels, respectively.

The assumption that each video segment contains *exactly* one action to be recognized still holds. We remind that in order to address the problem of temporal variability between

Table 4.1: Handcrafted features, extracted from the skeletal joints. Note that each feature is calculated for a given joint J and may involve a different subset of frames. Features marked with *, are calculated using only *HandLeft* and/or *HandRight*. Also, $B(\bullet)$ denotes a 3D bounding box of a set of vectors, while $a_{B(\bullet)}, b_{B(\bullet)}$ the lengths of the sides of $B(\bullet)$. For a given joint J , J_c, J_p denote its child and parent node, respectively.

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	$F_1, \dots, F_N$	$\sum_{i=1}^N \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\ \mathbf{v}_i^{(J)}\ \ \mathbf{v}_{i-1}^{(J)}\ } \right)$
Squared total vector angle	$F_1, \dots, F_N$	$\sum_{i=1}^n \arccos \left(\frac{\mathbf{v}_i^{(J)} \cdot \mathbf{v}_{i-1}^{(J)}}{\ \mathbf{v}_i^{(J)}\ \ \mathbf{v}_{i-1}^{(J)}\ } \right)^2$
Total vector displacement	F_N, F_1	$\ \mathbf{v}_N^{(J)} - \mathbf{v}_1^{(J)}\ $
Total displacement	$F_1, \dots, F_N$	$\sum_{i=1}^n \ \mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\ $
Maximum displacement	$F_1, \dots, F_N$	$\max_{i=2, \dots, N} (\ \mathbf{v}_i^{(J)} - \mathbf{v}_{i-1}^{(J)}\)$
Bounding box diagonal length*	$F_1, \dots, F_N$	$\sqrt{a_{B(\mathcal{V}(\mathcal{J}))}^2 + b_{B(\mathcal{V}(\mathcal{J}))}^2}$
Bounding box angle*	$F_1, \dots, F_N$	$\arctan \frac{b_{B(\mathcal{V}(\mathcal{J}))}}{a_{B(\mathcal{V}(\mathcal{J}))}}$
Initial angle	F_1	$\angle \mathbf{v}_1^{(J)} \mathbf{Ov}_1^{(J_p)} \text{ or } \angle \mathbf{v}_1^{(J)} \mathbf{Ov}_1^{(J_c)}$
Final angle	F_N	$\angle \mathbf{v}_N^{(J)} \mathbf{Ov}_N^{(J_p)} \text{ or } \angle \mathbf{v}_N^{(J)} \mathbf{Ov}_N^{(J_c)}$
Mean angle	$F_1, \dots, F_N$	$\frac{1}{N} \sum_{i=1}^N \angle \mathbf{v}_i^{(J)} \mathbf{Ov}_i^{(J_p)} \text{ or } \frac{1}{N} \sum_{i=1}^N \angle \mathbf{v}_i^{(J)} \mathbf{Ov}_i^{(J_c)}$
Max angle	$F_1, \dots, F_N$	$\max_{i=1}^N \angle \mathbf{v}_i^{(J)} \mathbf{Ov}_i^{(J_p)} \text{ or } \max_{i=1}^N \angle \mathbf{v}_i^{(J)} \mathbf{Ov}_i^{(J_c)}$

actions and between users which results to video segments of different lengths, a linear interpolation step is imposed, setting the duration of all videos equal to N frames. By using skeletal information collected by the Microsoft Kinect v2, i.e., consisting of 25 joints, we ended up with pseudocolored images having dimension $25 \times N \times 3$. Each was created

as follows: we calculate coordinate differences between consecutive frames, thus if $v_{x,i}^{(j)}$ denotes the x -position of the j -th joint in the i -th frame. Let R denote the red channel of the color image. The value of $R(j, i)$ is calculated as: $R(j, i) = v_{x,i+1}^{(j)} - v_{x,i}^{(j)}$, $i = 1, \dots, N$. Similarly, blue and green channels are constructed. 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. Examples of pseudo-colored images are illustrated in Fig. 4.1.

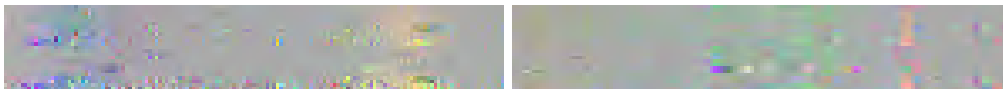


Figure 4.1: Representative pseudo-colored images from actions of the PKU-MMD dataset [125] and by using all 25 skeletal joints. Top: *eat_meal_snack*, bottom: *falling*.

We employ the same CNN architecture as depicted in Fig. 3.2 and discussed in the preceding section 3.3 for classification.

4.3.3 HAR fusing Contextual and Handcrafted Features

We will now describe the methodology for HAR that is based on the fusion of contextual and handcrafted features. A visual overview of the proposed approach is illustrated in Fig. 4.2. As it may be seen, we use two distinct feature extraction steps. The first consists of the handcrafted features that are extracted by 3D trajectories of skeletal joints as described in subsection 4.3.1. Each activity is represented by a feature vector. We should emphasize that this description aims to represent the activity in a space that is discriminative with regards to the involved classes of the problem at hand. The second consists of the context-aware deep features that are extracted using a CNN, trained by using visual representations of 3D skeletal joint trajectories, as described in subsection 4.3.2. The CNN has been trained in a different (yet semantically similar) problem and the output of its last fully connected layer is used as the extracted feature. Both feature representations are combined in an early fusion step and an SVM is used for classification upon a PCA step.

The motivation for our approach is that since, the CNN is trained using different context classes, it learns a complementary representation to the one provided by the handcrafted features. As it has been shown to a plethora of works, such as the one of Giannakopoulos et al. [52], the combination of such diverse representations leads to improved recognition accuracy. In our approach, the CNN is trained at a broader activity recognition problem, while the problem at hand is more specific, i.e., the one of arm gesture recognition.

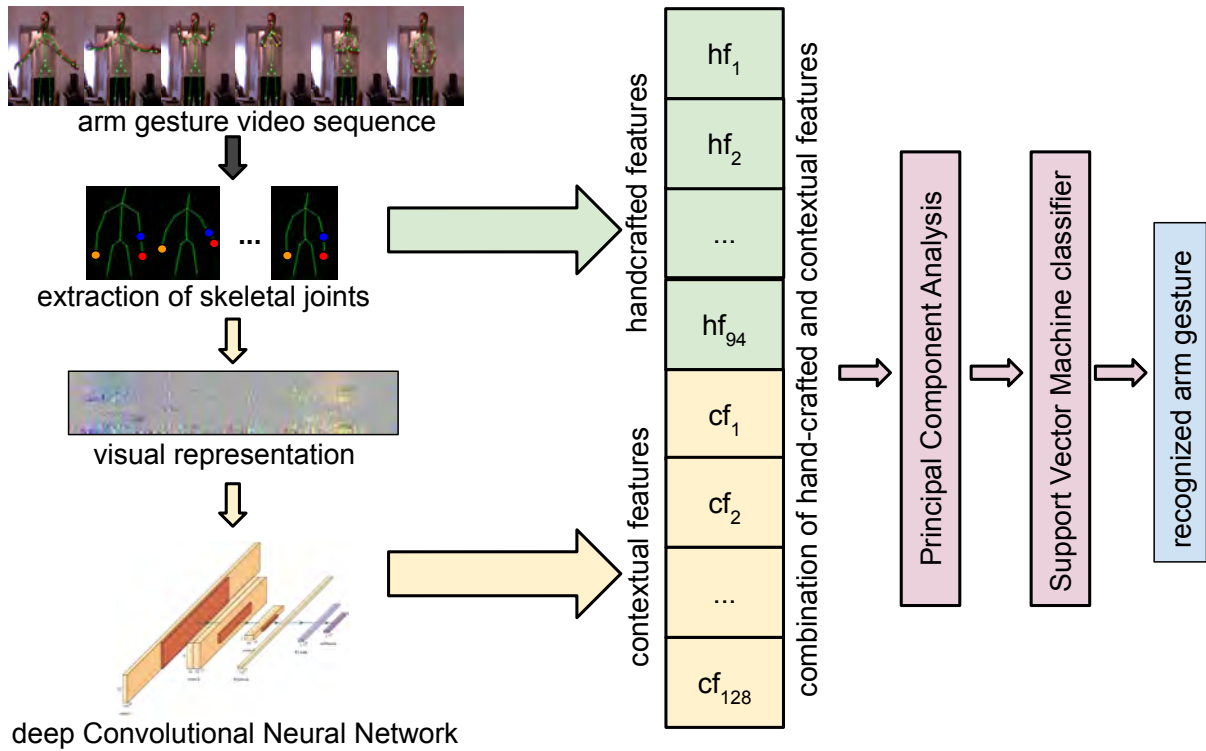


Figure 4.2: A visual overview of the approach for classification of activities based on the fusion of contextual and handcrafted features.

4.4 Experimental Results

4.4.1 Gesture recognition dataset

For the evaluation of the proposed fusion approach we used the segmented dataset of Mathe et al. [134]. Specifically this dataset consists of 200 arm gestures performed by 10 users and captured by a single camera. Gestures are divided into the following classes: *clapping*, *hands raise*, *hands circle*, *swipe up*, *swipe down*, *swipe upright*, *swipe downright*. In both cases, we have used the 3D skeletal data derived from the joints' motion.

4.4.2 Model Training

Since our goal was to solve an arm gesture classification problem, we extracted handcrafted features only from hands, arms and wrists, i.e., *ElbowLeft*, *ElbowRight*, *HandLeft*, *HandRight*, *WristLeft*, *WristRight*. From each of these 6 joints, we obtained the features listed in Table 4.1, resulting in a feature vector of size 94. The CNN was trained for 11 classes of the PKU-MMD dataset, as in [185], which were: *eat meal snack*, *falling*, *hand-shaking*, *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*. For clas-

Table 4.2: Classification accuracy for all feature extraction methods.

Method	Accuracy
HF	0.87
CF	0.43
HF+CF	0.93

sification, we used a linear SVM. The dimension of the last fully connected layer was 128, resulting in a combined representation of size 212. After PCA, we kept only the components that accounted for 95% of total variance, resulting in a feature vector of size 57, which was fed to the SVM. Note that data from the PKU-MMD dataset have been gathered using the Microsoft Kinect v2 camera, while data from our the arm gesture dataset have been collected with the Microsoft v1 camera, which only provides a subset of the available joints. To create the visual representation, we set to 0 all rows corresponding to the non-available joints and also applied a linear interpolation step to make the dimension equal to the input of the trained CNN.

4.4.3 Results

We conducted three series of experiments to evaluate the proposed approach. We used only handcrafted features (HF), only contextual features (CF) extracted by a CNN, and a combination of both (HF+CF). We pre-processed the features with PCA and trained a linear SVM. The results, shown in Table 4.2, demonstrate that the HF+CF feature extraction approach outperforms the other two approaches by approximately 7%. Notably, the CF-only approach exhibited the poorest performance.

4.5 Conclusions

In this Chapter, we have proposed an approach for human activity recognition using transfer learning. Our approach focuses on fusing handcrafted features and learned features from skeletal data, which capture human motion in the 3D space. The handcrafted features are extracted based on angles, displacements, and lengths of joints during an activity. On the other hand, the learned features are obtained from a convolutional neural network that has been trained on a large-scale action dataset. Our objective is to transfer the knowledge gained from a similar problem in the field of human activity recognition to the specific task of arm gesture recognition. To achieve this, we first extracted handcrafted features from a relatively small arm gesture dataset. Then, we utilized a pre-trained convolutional neural network as a feature extractor, specifically using the output of its last dense layer as a feature vector. This network provided “context-aware” deep features. We applied a late fusion approach, combining both the handcrafted and learned feature vectors, and trained a support vector machine for classification. Through experimentation, we demonstrated that the combination of these two types of features lead to improved per-

formance. The learned features contribute a diverse representation that is not captured by the handcrafted features alone. This successful transfer of knowledge allows us to leverage the more generalized action recognition task to improve the specific task of arm gesture recognition.

5. A DATA AUGMENTATION APPROACH FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION

5.1 Introduction

As we have already mentioned in Chapter [1], occlusion consists probably the most important challenge within the broader area of HAR. To understand the notion of occlusion, let us consider two objects in a 3D space. Given a certain viewpoint, occlusion occurs when one of the objects partially/fully blocks the view of the second. In the context of HAR, occlusion may occur e.g., in case when the subject performs an activity and is occluded by a physical object such as furniture, or by another person. Regarding its effect in HAR, let us consider the (extreme) case of a subject performing the activity “handwaving,” while the involved arm and hand are not visible. Clearly, in order to recognize this activity, crucial information regarding the subject’s motion is missing. This case might be excessive, but there are applications like e.g., surveillance systems in crowded places such as metro stations where an activity of a subject might not be recognized due to occlusion of some of its body parts from the crowd surrounding her/him. Occlusion may also be temporally partial, i.e., one or more body parts may be occluded but not during the whole activity. This may happen e.g., in the case of a moving subject.

In this chapter, our goal is to assess how partial occlusion of the subject affects the accuracy of HAR and to overcome its effect. To the best of our knowledge public motion-based datasets that may be used for HAR [125, 172] have been created under ideal laboratory conditions, which include prevention of any type of occlusion. Thus, since the creation of a large scale dataset is a time consuming task, we decided to follow an approach such as the one of Gu et al. [60]. More specifically, we simulate occlusion by manually discarding subsets of joints that correspond to one/two body parts of the subject, while she/he is performing a given activity. We assume that these parts remain occluded during the whole activity, that is, we do not consider the case of temporally partial occlusion. The contribution of this chapter is twofold: a) we extend the small-scale evaluation that has been performed by Giannakos et al. [53], where the authors studied how partial occlusion of the subject may affect the accuracy of HAR focusing on Activities of Daily Living (ADLs) [99]; herein we evaluate using two challenging large-scale human motion datasets [125, 172]. b) we try to overcome the effect of occlusion by incorporating occluded samples in our training procedure.

5.2 Related work

A plethora of research efforts have focused on the problem of HAR, while the majority of them is based on 2D representations of skeletal motion which are then used as input to CNNs [43, 194, 66, 110, 124, 87]. Interestingly, although it is widely accepted, as it is intuitive, that occlusion consists one of the most important factors that compromise the per-

formance of HAR approaches [76], rendering them to produce poor and in extreme cases unusable results, there do not exist many research works that focus either on studying its effects on the performance of HAR approaches or even attempt to propose methods so as overcome them. At the following, we shall attempt to present the approaches that have dealt with occlusion and their main findings.

Giannakos et al. [53], have extensively studied the effect of occlusion in the task of HAR. They relied on the HAR approach of Papadakis et al. [147] and simulated occlusion upon removing one or more body parts from 3D human skeletons and during the whole activity. Throughout their experiments they used a convolutional neural network that had been trained without using any occluded samples. Specifically, they considered the cases of removing one or both arms, one or both legs and also the cases of removing the arm and the leg of the same side. Within their experimental evaluation, they proved that the removal of one or both arms led to a drop of performance which in some cases was considered quite severe. However, they concluded that this was not a surprise, as most of the activities considered therein were expressed mainly by one or both arms' motion.

Angelini et al. [11] proposed a HAR approach that relied on OpenPose [25] for pose extraction. Specifically, they extracted both low- and high-level features from body poses, using an 1-D CNN and an LSTM for classification. They used both real CCTV data where part of the body was occluded, however they also created synthetic occluded data by removing body parts. They experimented by incorporating occluded samples in training and showed that in most cases this strategy is able to provide a boost of performance both in cases of occlusion and missing data. They experimented in multiple datasets, i.e., i3DPost [54], IXMAS [65], KTH [166] etc., while they also created their own dataset called ISLD, comprising classes such as bend, box, hand-clap, one-hand wave, run, walk, etc.

Iosifidis et al.[76], used a multi-camera setup, surrounding the subject from all sides. They trained a model using data from all available cameras and then, in order to simulate occlusion, they used a subset of those cameras. Their main idea was that due to occlusion, not all cameras are able to simultaneously capture the subject's motion, yet in all cases more than one cameras manage to capture the whole body of the subject. In addition, the recognition of an activity was performed using the combined results of cameras that had not been "affected" by any occlusion, while those that have been omitted "suffered" from occlusion. They conducted experiments using two publicly available datasets, the i3DPost [54] comprising 8 daily activities and the AIIA-MOBISERV [77] comprising 3 meal activities. The proposed method has been evaluated for both single-view and multi-view action classification problems, providing results with sufficient accuracy.

Moreover, Li et al. [114] used manually defined occlusion areas and proposed a new encoding technique to transform a skeleton into a feature matrix and integrated an attention model into a Generative Adversarial Network, focusing on completing the missing data of the feature matrix, rather than the exact skeleton. For classification, they used a Residual Network (ResNet). To evaluate their approach, they used a custom dataset of 8 actions of construction workers, such as driving truck, transporting cement, paving concrete, etc. Also, Li et al. [112] proposed an approach which is based on action graphs so as to capture the dynamics of actions and used a bag-of-3D points so as to describe the salient

postures that correspond to the nodes of the action graph. They simulated occlusion by first dividing the depth map into four quadrants and then by ignoring 3D points falling into the specified quadrants that are considered to be affected by occlusion. They did not include any occluded samples in training. They used a dataset comprising 20 actions such as high arm wave, horizontal arm wave, hammer, etc. Their results showed that unless the critical quadrants were omitted, performance drop was relatively small.

Similarly, Gu et al.[60], generated occlusion masks which were used both in training and evaluation, so as to simulate occlusion in more than one 2-D skeletal joints. Moreover, they attempted to reconstruct the missing skeleton parts, using a regression network. They evaluated their work using their own dataset, namely MMHuman, comprising 6 classes (e.g., Kung Fu, Shake Hand, etc.). They also demonstrated that their approach is able to outperform previous pose estimation methods on the Human3.6M dataset [75]. Finally, Yang et al.[211] proposed to pre-train a single pose embedding network, namely OR-VPE whose role was to learn occlusion robust representations for pose sequences. They simulated occlusions using an augmentation mechanism whose goal was to randomly prune the spatio-temporal skeleton structure. They used an encoder in order to create the pose embeddings into a latent space and a Contrastive Module to render this space occlusion-invariant. They evaluate their approach using the NTU-RGB+D dataset [172] and also the Toyota Smarthome [37], N-UCLA[190] and Penn Action [219] datasets.

5.3 Methodology

5.3.1 Image Representation of Skeletal Data

Since our goal is to use a CNN for the classification of activities, a mandatory step is to represent the 3-D skeletal motion into a 2-D image representation, so that the latter would capture and preserve the spatio-temporal properties of this motion. Of course, such representations should be able to discriminate among different activities, while also comply to the graph structure of the skeleton representation that is used. Therefore, we adopted the pseudo-colored representation of the skeletal motion that aims to capture the varying inter-joint distances during the performance of an activity and has been presented in chapter 3.

We remind in brief that by $x(n)$, $y(n)$ and $z(n)$, $n = 1, \dots, N$ we denote the sequences of the x, y, z co-ordinates of the skeletal joint motion at the n -th frame and occurring within a period of time of N frames. Since Kinect v2 extracts 25 joints, this motion “produces” a set of 75 such sequences. Prior to the creation of the image representation, and in order to address the problem of temporal variability we use a linear interpolation step, setting the number of frames F equal for all activity examples. In this case and upon an experimental process, we set $F = 150$. In order to create the pseudocolored images, spatial co-ordinates x, y, z are assigned to R, G, B color channels of the resulting pseudo-colored image, respectively. E.g., by R denoting the red channel, i the i -th frame, and j the j -th joint, the R -value of pixel (i, j) is $R(i, j) = R(i + 1, j) - R(i, j)$. Similarly, G and B

values are created. In Fig. 5.1, examples of these pseudo-colored images are illustrated.

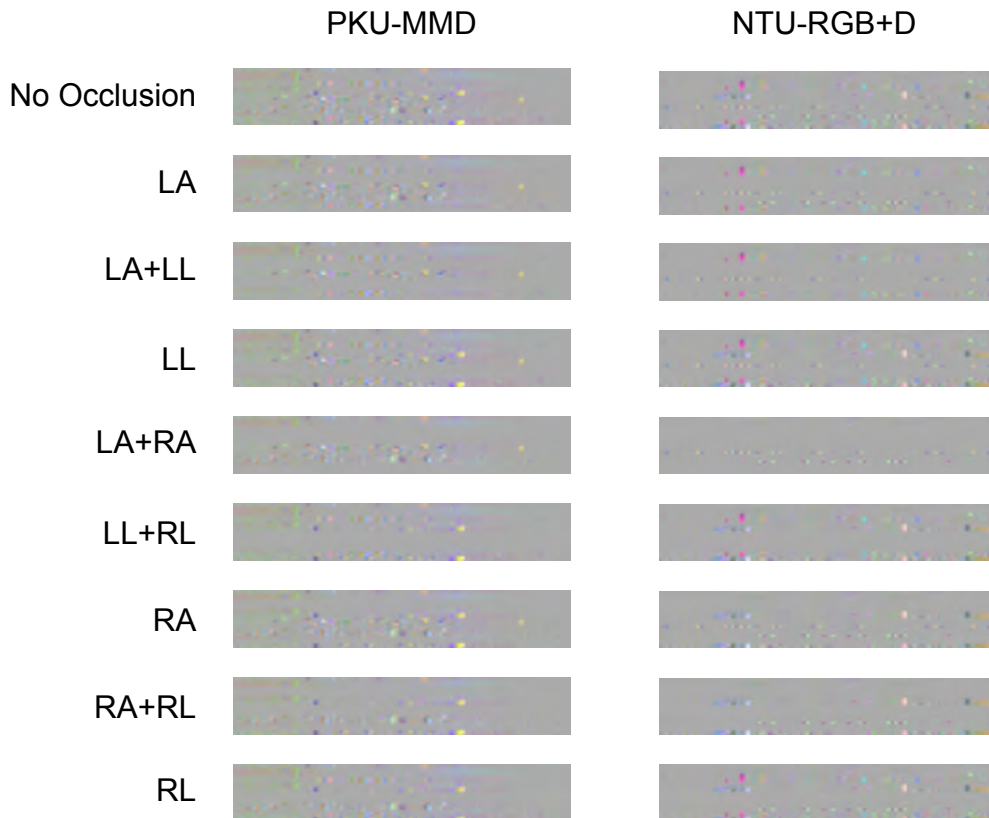


Figure 5.1: Pseudocoloured images for the PKU-MMD [125] and the NTU [172] datasets. LA, RA, LL and RL denote the cases of occlusion of left arm, right arm, left leg and right leg.

5.3.2 Occlusion of Skeletal Data

As we have already discussed, in real-life scenarios occlusion may be one of the most crucial causes of poor performance of HAR approaches. Prior to describe our approach for simulating occlusion, we first discuss the main causes. Let us consider an assistive living scenario, where the subject's behaviour is monitored by cameras. In many occasions, actions take place behind furniture or more than one people are present in the same room. Thus, it is expected that occlusion is often occurring, incommoding the task of recognition algorithms, as it is the main factor for loss of visual information. Even in cases of simple activities that involve one or two body parts, it is obvious that by occluding them, i.e., by losing the information regarding their motion, recognition algorithms are prone to failure.

As we have already mentioned in Chapter 1, amongst the goals of this Thesis is to assess whether the performance drop caused by occlusion may be limited upon adding occluded samples in the training process. First, we should emphasize that most publicly available,

large scale datasets such as the ones we use for the evaluation of this work, i.e., NTU RGB+D [172] and the PKU-MMD [125], have been created in laboratory (i.e., ideal) conditions. Thus, there exist no data samples that are affected by occlusion, while room illumination is appropriate for video capturing. We assume that these datasets are “clean,” i.e., they do not contain any samples with missing/noisy data and the only limitations imposed are due to the capabilities of Kinect v2 which is used for data collection.

Now, and in order to simulate the effect of occlusion, i.e., to create artificially occluded activity samples, we follow the paradigms of [60, 53] and we discard subsets of the skeleton that correspond to meaningful body parts. Specifically, as we have already discussed in section 2.8, we consider arms (each comprising shoulder, elbow, wrist, hand and hand-tip), legs (each comprising hip, knee, ankle and foot) and the torso (comprising head, neck, spine-shoulder, spine-mid and spine-base). This way we form 5 body parts, as illustrated in Fig. 2.12. We should herein note that in our case, we assume that occlusion affects the whole activity, i.e., the body parts that are occluded, remain occluded during the whole activity duration.

Specifically, 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. Intuitively, when two parts are occluded, the most expected case is that they both are from the same side. Moreover, our initial experiments indicated that by removing the torso the accuracy was not significantly affected. Therefore, throughout our experiments, torso is always present. Examples of activities with and without occlusion of a body part are illustrated in Figures 5.2, 5.3. In these examples it is evident that a single body part may carry significant information regarding the activity.

We should herein note that several research works, among those that have been presented in section 5.2 also use occluded samples within the training process [60, 128, 11]. However, in [11], only random skeletal joints are removed; yet these do not correspond to structured body parts as in our case. Moreover, in [60] occlusion is not continuous as in our case; instead it may from partial to full during an activity, yet within several temporal periods, no occlusion is present. Also, contrary to [76], where due to occlusion only some cameras are affected, in our approach, all available cameras are similarly affected.

5.3.3 Classification

For classification we have used a variant of the CNN architecture of Chapter 3. Specifically, it consists of a 2D convolutional layer that filters the 25×150 input image with 5 kernels of 3×3 size, a max-pooling layer that performs 2×2 subsampling, two consecutive convolutional layers of size 3×3 with 10 and 15 kernels, a max-pooling layer performing 2×2 subsampling, a flatten layer that transforms the output of the last pooling layer into a vector, which consists the input to a dense layer upon applying a dropout layer with dropout rate equal to 0.5 and a second dense layer producing the output of the network. The CNN is illustrated in Fig. 5.4.

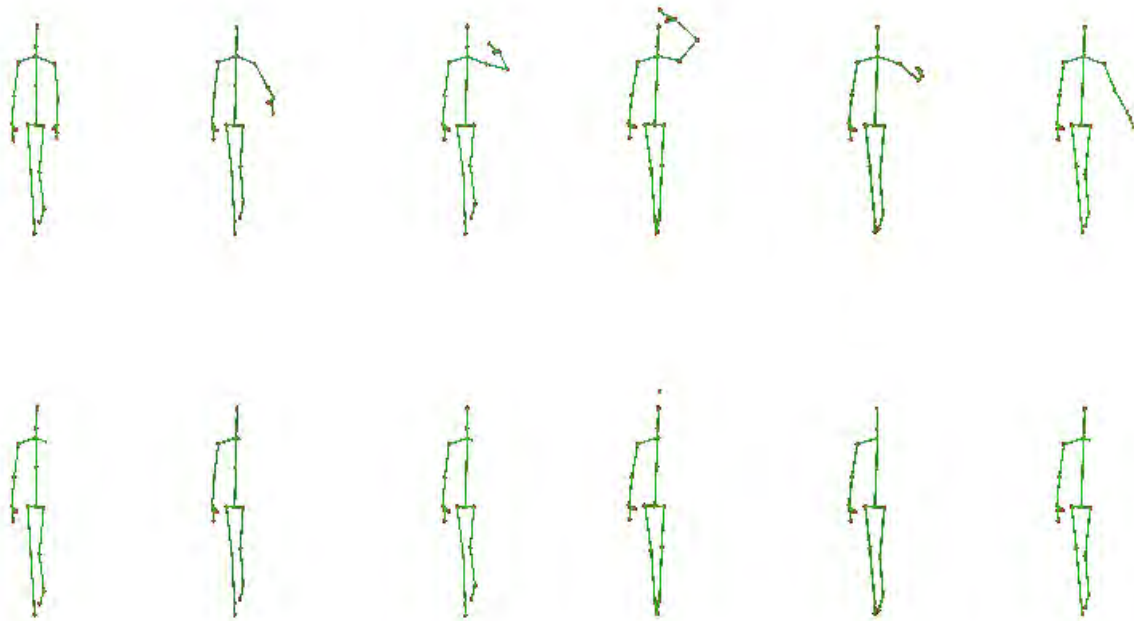


Figure 5.2: Example skeleton sequences of the activity *salute* from the PKU-MMD dataset [125]. First row: skeletons include all 25 joints; Second row: joints corresponding to left arm have been discarded.

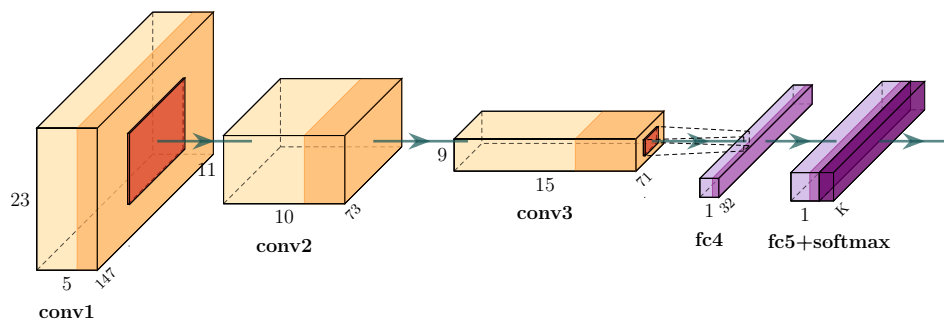


Figure 5.4: The convolutional neural network that has been used for the experimental evaluation of the methodology presented in this Chapter.

5.4 Experiments and Results

In this section we present and discuss the results of the experimental evaluation of the proposed methodology, using two publicly available and well-known human activity recog-

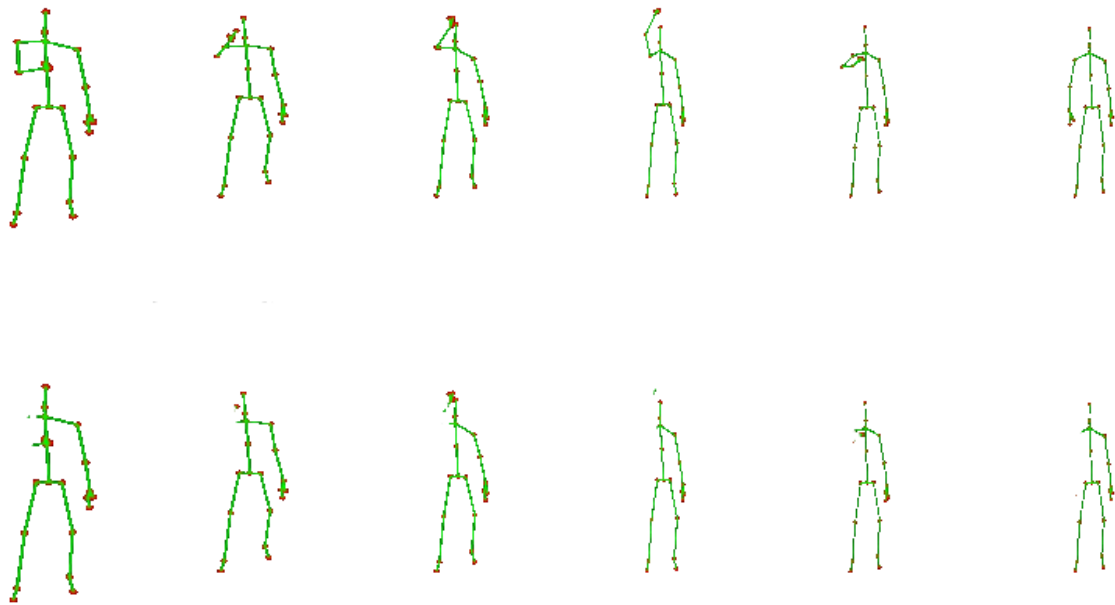


Figure 5.3: Example skeleton sequences of the activity *wearing a hat* from the NTU dataset [172]. First row: skeletons include all 25 joints; Second row: joints corresponding to left arm have been discarded.

dition datasets. Specifically, we have used PKU-MMD [125] and NTU-RGB+D [172]. The goal of this evaluation is to assess whether the inclusion of occluded samples within the training process may improve the performance of classification strategies. Note that the goal of the following experiments was not to provide state-of-the-art results in human activity recognition, but to be used as a baseline so as to verify the effectiveness of our approach. The CNN that has been presented in subsection 5.3.3 was trained using batch size equal to 32 and for 80 epochs. Moreover, the optimizer used was the RMSprop, while the categorical cross-entropy loss was adopted. The data split for training, validation and testing followed exactly the one proposed from the datasets' authors [125, 172].

5.4.1 Evaluation Protocol

For the evaluation of the proposed approach, we studied 8 cases of missing body parts of the skeleton, as we have discussed in section 5.3.2. Specifically, we performed three types of experiments for comparison purposes:

- a. training and test sets consist of full body parts only;
- b. training set consists of full body parts only, test set consists of occluded body parts only;

- c. training set consists of full and occluded body parts, test set consists of occluded body parts.

Moreover, following typical evaluation protocols that have been presented in Chapter 2 and are encountered in multi-camera datasets such as those used herein [125, 172], we further split the aforementioned experiments into three types:

- a. per camera position (single view), where training and test sets derive from the same camera (viewpoint), e.g., both deriving from middle camera;
- b. cross view experiments, where training and test sets derive from different cameras (viewpoints), e.g., training deriving from middle camera, testing deriving from left camera;
- c. cross subject experiments, wherein subjects are split into training and testing groups, i.e., each subject appears exactly into one of these groups.

We performed experiments using full PKU-MMD and NTU-RGB+D datasets. Moreover, we also performed experiments using subsets of both these datasets, as follows:

- a. from PKU-MMD dataset we selected 11 actions that are related to activities of daily living: 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;
- b. from NTU-RGB+D dataset we selected 12 medical conditions: sneeze/cough, staggering, falling down, headache, chest pain, back pain, neck pain, nausea/vomiting, fan self, yawn, stretch oneself, and blow nose.

5.4.2 Results and Discussion

In all cases we measured F_1 score and accuracy. Results of the proposed methodology are depicted in Table 5.1 for the 11 activities of daily living of the PKU-MMD dataset, in Table 5.3 for the 12 medical conditions of NTU-RGB+D dataset and in Tables 5.2 and 5.4 for the full PKU-MMD and NTU-RGB+D datasets, respectively.

Starting with PKU-MMD dataset and based on the results depicted in Tables 5.1 and 5.2 we may observe the following:

- in almost all experiments, both the F_1 score and the accuracy of the proposed approach is improved compared to the “simple” case of occlusion;
- in general, it is also comparable to the baseline case;
- the only exception to the above-mentioned are the cases of cross-view experiments when using:

Table 5.1: Results of the evaluation using the 11 activities of daily living of the PKU-MMD dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases.

Setup	Train	Test	Metric	B	LA		LA+LL		LL		LA+RA		LL+RL		RA		RA+RL		RL	
					S	A	S	A	S	A	S	A	S	A	S	A	S	A	S	A
CS	LRM	LRM	F_1	.68	.56	.71	.53	.71	.62	.74	.18	.59	.47	.69	.53	.70	.49	.69	.66	.73
			Acc.	.70	.59	.74	.55	.75	.64	.77	.28	.64	.52	.72	.55	.73	.50	.71	.67	.75
SV	L	L	F_1	.68	.51	.76	.49	.77	.70	.77	.18	.68	.58	.76	.40	.70	.37	.72	.66	.75
			Acc.	.69	.54	.76	.54	.77	.71	.78	.29	.69	.60	.76	.42	.70	.40	.71	.66	.76
	M	M	F_1	.77	.66	.81	.62	.79	.74	.81	.27	.72	.63	.76	.63	.80	.57	.75	.74	.79
			Acc.	.78	.67	.81	.64	.79	.75	.81	.37	.72	.66	.76	.65	.79	.59	.75	.74	.79
	R	R	F_1	.75	.54	.75	.49	.71	.71	.71	.16	.63	.64	.70	.59	.71	.59	.73	.72	.73
			Acc.	.76	.57	.76	.53	.72	.72	.72	.29	.66	.65	.71	.62	.72	.60	.74	.72	.74
	L	M	F_1	.71	.59	.78	.58	.77	.69	.79	.22	.70	.60	.76	.61	.75	.60	.74	.70	.79
			Acc.	.72	.62	.78	.60	.78	.70	.79	.32	.70	.63	.77	.64	.75	.62	.74	.71	.79
	L	R	F_1	.70	.50	.64	.46	.63	.66	.67	.15	.65	.54	.67	.50	.68	.47	.69	.64	.69
			Acc.	.70	.52	.65	.48	.64	.66	.68	.24	.66	.58	.67	.53	.69	.51	.69	.65	.70
CV	M	L	F_1	.71	.58	.77	.56	.77	.68	.76	.19	.67	.61	.71	.48	.71	.44	.69	.70	.74
			Acc.	.72	.62	.78	.60	.77	.69	.76	.32	.69	.63	.72	.50	.71	.47	.68	.70	.74
	M	R	F_1	.55	.35	.66	.34	.63	.52	.69	.15	.51	.40	.64	.42	.70	.38	.70	.50	.70
			Acc.	.56	.41	.67	.39	.64	.53	.70	.26	.56	.43	.66	.44	.71	.40	.72	.51	.71
	R	L	F_1	.60	.53	.62	.48	.60	.55	.59	.18	.48	.44	.56	.49	.57	.48	.55	.58	.57
			Acc.	.59	.54	.64	.49	.62	.54	.59	.29	.53	.47	.56	.50	.57	.49	.55	.58	.57
	R	M	F_1	.73	.60	.70	.52	.69	.65	.74	.25	.61	.57	.72	.60	.71	.57	.70	.72	.74
			Acc.	.73	.61	.71	.54	.70	.66	.74	.34	.63	.60	.72	.60	.72	.57	.71	.72	.74
	LR	M	F_1	.87	.70	.88	.67	.87	.84	.90	.23	.80	.68	.88	.60	.88	.56	.88	.83	.89
			Acc.	.87	.71	.88	.68	.86	.85	.90	.36	.79	.71	.88	.64	.88	.59	.88	.83	.89
	LM	R	F_1	.79	.67	.65	.65	.62	.76	.68	.21	.52	.64	.64	.56	.66	.53	.67	.77	.70
			Acc.	.79	.68	.66	.66	.63	.76	.69	.34	.54	.66	.65	.59	.66	.55	.67	.77	.70
	RM	L	F_1	.74	.68	.79	.69	.79	.74	.78	.22	.68	.62	.71	.55	.73	.54	.71	.72	.76
			Acc.	.75	.69	.80	.70	.80	.74	.77	.34	.69	.63	.71	.57	.73	.56	.70	.72	.76

Table 5.2: Results of the evaluation using the PKU-MMD dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases.

Setup	Train	Test	Metric	B	LA		LA+LL		LL		LA+RA		LL+RL		RA		RA+RL		RL	
					S	A	S	A	S	A	S	A	S	A	S	A	S	A	S	A
CS	LRM	LRM	F_1	.69	.29	.51	.25	.48	.65	.64	.10	.34	.60	.62	.39	.59	.36	.58	.65	.65
			Acc.	.70	.33	.53	.29	.51	.66	.66	.17	.37	.62	.65	.45	.61	.42	.60	.66	.67
SV	L	L	F_1	.64	.29	.52	.27	.49	.58	.65	.09	.37	.50	.63	.30	.60	.28	.60	.61	.65
			Acc.	.65	.33	.53	.30	.50	.59	.66	.16	.39	.52	.64	.36	.60	.35	.60	.61	.66
	M	M	F_1	.63	.26	.54	.21	.51	.59	.67	.09	.38	.50	.67	.36	.65	.33	.65	.62	.68
			Acc.	.64	.29	.55	.26	.52	.60	.68	.15	.41	.53	.68	.41	.65	.39	.65	.62	.69
	R	R	F_1	.63	.29	.53	.26	.53	.60	.65	.09	.37	.56	.62	.38	.64	.33	.63	.61	.65
			Acc.	.63	.32	.54	.30	.54	.60	.65	.15	.39	.57	.63	.42	.64	.38	.63	.62	.65
CV	L	M	F_1	.61	.23	.50	.22	.47	.55	.64	.10	.34	.49	.63	.34	.60	.32	.60	.59	.65
			Acc.	.62	.27	.51	.26	.48	.56	.65	.15	.37	.53	.64	.39	.61	.37	.61	.59	.66
	L	R	F_1	.58	.23	.34	.21	.33	.53	.44	.08	.27	.45	.44	.36	.42	.32	.42	.56	.46
			Acc.	.59	.28	.37	.25	.35	.54	.46	.14	.30	.49	.46	.40	.44	.36	.45	.57	.48
	M	L	F_1	.61	.28	.53	.26	.52	.54	.63	.08	.34	.49	.62	.34	.58	.33	.58	.58	.64
			Acc.	.62	.31	.54	.28	.53	.56	.64	.14	.37	.51	.63	.39	.59	.38	.58	.60	.65
	M	R	F_1	.50	.20	.47	.18	.46	.45	.62	.09	.34	.39	.62	.29	.61	.24	.61	.47	.63
			Acc.	.51	.24	.49	.22	.47	.47	.63	.14	.36	.41	.62	.32	.61	.27	.61	.47	.63
	R	L	F_1	.47	.18	.38	.16	.37	.41	.47	.08	.25	.37	.44	.26	.44	.26	.43	.44	.48
			Acc.	.48	.22	.40	.19	.39	.43	.49	.16	.28	.40	.47	.32	.46	.33	.45	.46	.50
	R	M	F_1	.60	.25	.48	.22	.47	.54	.62	.08	.33	.46	.61	.38	.58	.33	.57	.57	.64
			Acc.	.61	.30	.50	.24	.48	.54	.63	.16	.35	.48	.62	.42	.58	.37	.58	.57	.64
	LR	M	F_1	.70	.31	.58	.26	.56	.65	.74	.09	.40	.60	.73	.42	.69	.38	.69	.68	.75
			Acc.	.70	.34	.58	.30	.56	.67	.74	.15	.41	.63	.73	.46	.69	.43	.68	.68	.75
	LM	R	F_1	.65	.26	.50	.25	.49	.61	.63	.11	.36	.56	.62	.36	.61	.35	.62	.60	.64
			Acc.	.66	.30	.52	.29	.50	.62	.64	.17	.39	.58	.64	.42	.62	.41	.62	.61	.65
	RM	L	F_1	.65	.27	.50	.26	.49	.60	.62	.09	.34	.55	.60	.36	.56	.35	.55	.61	.62
			Acc.	.65	.30	.51	.30	.49	.61	.63	.15	.36	.56	.60	.40	.57	.40	.56	.62	.63

Table 5.3: Results of the evaluation using the 12 medical conditions of the NTU-RGB+d dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases.

Setup	Train	Test	Metric	B		LA		LA+LL		LL		LA+RA		LL+RL		RA		RA+RL		RL	
						S	A	S	A	S	A	S	A	S	A	S	A	S	A	S	A
CS	LRM	LRM	F_1	.61	.30	.55	.32	.54	.59	.64	.18	.44	.51	.64	.43	.61	.43	.62	.58	.66	
			Acc.	.61	.32	.54	.33	.54	.62	.64	.22	.42	.58	.65	.44	.62	.44	.62	.58	.66	
SV	L	L	F_1	.60	.35	.54	.37	.54	.58	.64	.17	.41	.46	.63	.45	.61	.43	.62	.57	.66	
			Acc.	.61	.38	.55	.39	.54	.59	.64	.24	.44	.48	.64	.47	.62	.45	.62	.57	.66	
	M	M	F_1	.49	.28	.50	.30	.49	.49	.59	.12	.42	.36	.61	.38	.59	.37	.60	.46	.61	
			Acc.	.50	.30	.50	.32	.49	.49	.60	.19	.43	.38	.61	.41	.59	.38	.60	.46	.62	
	R	R	F_1	.53	.29	.48	.30	.48	.51	.59	.11	.37	.44	.62	.39	.54	.37	.55	.52	.61	
			Acc.	.54	.31	.48	.32	.48	.52	.59	.19	.37	.46	.62	.42	.54	.40	.55	.52	.61	
	L	M	F_1	.50	.25	.50	.26	.51	.43	.63	.14	.38	.36	.63	.39	.62	.36	.64	.47	.64	
			Acc.	.50	.28	.50	.27	.51	.43	.63	.19	.40	.39	.63	.39	.62	.37	.64	.46	.64	
	L	R	F_1	.50	.33	.50	.34	.51	.49	.58	.18	.37	.42	.58	.38	.53	.37	.54	.48	.58	
			Acc.	.50	.35	.50	.35	.51	.49	.58	.22	.37	.43	.58	.39	.53	.38	.55	.47	.59	
	M	L	F_1	.59	.31	.51	.33	.51	.55	.64	.13	.36	.47	.63	.45	.58	.44	.58	.55	.64	
			Acc.	.59	.33	.51	.34	.51	.55	.64	.19	.39	.49	.63	.47	.59	.44	.58	.55	.65	
CV	M	R	F_1	.61	.37	.42	.36	.45	.57	.53	.12	.30	.49	.54	.42	.49	.40	.50	.56	.53	
			Acc.	.61	.38	.43	.38	.45	.57	.54	.18	.31	.51	.54	.44	.49	.41	.50	.56	.54	
	R	L	F_1	.59	.28	.51	.31	.51	.57	.61	.11	.39	.49	.61	.43	.56	.42	.58	.55	.62	
			Acc.	.59	.31	.52	.32	.52	.57	.62	.18	.41	.50	.61	.44	.56	.43	.58	.55	.63	
	R	M	F_1	.55	.34	.39	.34	.39	.52	.47	.17	.29	.46	.47	.43	.46	.41	.49	.49	.49	
			Acc.	.55	.35	.40	.34	.39	.52	.47	.21	.31	.49	.47	.42	.46	.42	.48	.51	.49	
	LR	M	F_1	.57	.32	.51	.31	.52	.53	.61	.17	.44	.44	.64	.44	.62	.44	.65	.55	.64	
			Acc.	.58	.33	.52	.31	.52	.53	.62	.22	.45	.46	.64	.46	.63	.46	.65	.56	.65	
	LM	R	F_1	.60	.28	.51	.29	.52	.54	.61	.14	.37	.47	.61	.44	.55	.43	.56	.56	.60	
			Acc.	.60	.31	.52	.30	.53	.54	.61	.18	.38	.50	.61	.45	.55	.43	.56	.56	.61	
	RM	L	F_1	.65	.38	.58	.39	.59	.62	.72	.14	.44	.52	.72	.48	.68	.46	.69	.62	.72	
			Acc.	.65	.39	.58	.39	.59	.62	.73	.21	.45	.55	.72	.50	.69	.48	.69	.62	.73	

Table 5.4: Results of the evaluation using the NTU-RGB+D dataset. Setup cases are Cross-Subject (CS), Single-View (SV) and Cross-View (CV). Camera views are Left (L), Right (R) and Middle (M). In the baseline case (B), all body parts are included in both training and testing, while in the “simple” case of occlusion (S) training set consists of full body parts and test set consists of occluded body parts, and in the “augmented” case of occlusion (A), training set consists of full and occluded body parts, while test set consists of occluded body parts. Moreover, LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Also, F_1 and Acc. denote F_1 score and Accuracy, respectively. In all cases, bold numbers indicate best result between S and A cases.

Setup	Train	Test	Metric	B	LA		LA+LL		LL		LA+RA		LL+RL		RA		RA+RL		RL	
					S	A	S	A	S	A	S	A	S	A	S	A	S	A	S	A
CS	LRM	LRM	F_1	.49	.25	.37	.22	.36	.46	.45	.09	.25	.39	.45	.31	.44	.29	.44	.47	.47
			Acc.	.49	.26	.38	.23	.37	.47	.47	.12	.27	.41	.47	.32	.45	.31	.45	.47	.48
SV	L	L	F_1	.60	.25	.39	.20	.38	.50	.49	.08	.28	.37	.50	.34	.46	.29	.48	.53	.51
			Acc.	.60	.26	.40	.21	.39	.51	.50	.11	.29	.41	.50	.36	.46	.32	.48	.55	.51
	M	M	F_1	.57	.27	.36	.24	.35	.50	.46	.09	.24	.42	.46	.36	.44	.31	.43	.51	.48
			Acc.	.58	.28	.37	.25	.36	.51	.47	.11	.26	.45	.47	.38	.44	.32	.44	.52	.49
	R	R	F_1	.58	.27	.35	.22	.35	.51	.44	.08	.25	.40	.46	.32	.38	.27	.40	.52	.47
			Acc.	.59	.29	.36	.24	.35	.52	.45	.10	.27	.43	.46	.34	.38	.30	.40	.53	.47
CV	L	M	F_1	.38	.14	.35	.11	.34	.35	.45	.04	.24	.28	.46	.23	.43	.20	.44	.35	.47
			Acc.	.37	.16	.36	.13	.35	.35	.46	.06	.26	.30	.46	.23	.43	.20	.44	.34	.47
	L	R	F_1	.41	.24	.35	.20	.35	.37	.44	.12	.23	.31	.42	.26	.39	.25	.39	.39	.44
			Acc.	.41	.24	.35	.20	.35	.37	.44	.12	.23	.33	.43	.29	.39	.28	.39	.40	.44
	M	L	F_1	.46	.23	.39	.20	.38	.44	.49	.08	.25	.35	.49	.31	.45	.29	.45	.45	.50
			Acc.	.46	.24	.39	.21	.38	.44	.50	.10	.26	.37	.49	.32	.45	.32	.45	.45	.51
	M	R	F_1	.46	.26	.33	.25	.32	.43	.40	.09	.22	.37	.39	.30	.37	.28	.37	.45	.41
			Acc.	.46	.28	.33	.26	.32	.44	.40	.11	.23	.39	.40	.32	.37	.30	.37	.46	.41
	R	L	F_1	.44	.23	.33	.22	.33	.41	.44	.08	.21	.35	.43	.27	.39	.25	.39	.42	.45
			Acc.	.43	.24	.34	.23	.33	.41	.44	.10	.22	.37	.44	.28	.38	.27	.39	.42	.46
	R	M	F_1	.50	.23	.31	.20	.30	.47	.38	.09	.23	.42	.39	.32	.38	.28	.39	.47	.40
			Acc.	.50	.24	.32	.21	.31	.48	.39	.10	.25	.43	.39	.33	.39	.30	.39	.47	.41
	LR	M	F_1	.42	.22	.37	.21	.36	.40	.46	.07	.26	.35	.46	.27	.44	.26	.44	.41	.48
			Acc.	.43	.23	.38	.22	.37	.42	.47	.10	.28	.37	.47	.30	.44	.28	.45	.43	.49
	LM	R	F_1	.46	.24	.34	.22	.33	.43	.42	.09	.22	.38	.41	.27	.37	.26	.37	.44	.42
			Acc.	.45	.25	.35	.23	.34	.43	.43	.12	.24	.40	.42	.28	.38	.27	.38	.44	.43
	RM	L	F_1	.52	.29	.37	.24	.36	.48	.48	.07	.25	.40	.48	.32	.45	.28	.46	.48	.51
			Acc.	.52	.29	.38	.25	.36	.49	.49	.10	.26	.43	.48	.34	.45	.31	.46	.50	.51

- only the 11 activities of daily living, and specifically in the case where the left and the middle cameras have been used for training, while the right one has been used for testing and
- the full PKU-MMD dataset, and specifically in the case where the left camera has been used for training, while the right one has been used for testing;
- as expected and due to the activities involved, in cases of occlusion of legs, a drop of performance is not observed, in general.

Moreover, in the case of the NTU-RGB+D dataset and based on the results depicted in Tables 5.3 and 5.4, we may observe the following:

- also in that case both metrics indicated improved performance, compared to the “simple” case of occlusion;
- in general also in that case it is comparable to the baseline case;
- the only exception to the above-mentioned are the cases of the cross-view experiments when using:
 - only the 12 medical conditions, and specifically in the case where the middle camera has been used for training, while the right one has been used for testing or vice versa;
 - the full NTU-RGB+D dataset, and specifically in the case of single view;
 - a few combination of training and testing in the cross view case;
- also in this case and as expected, in case of occlusion of legs a drop of performance is not observed, in general.

A notable observation of our experimental results indicates that due to data augmentation that is indirectly caused by the addition of the occluded samples in the training phase, in many cases and in both datasets we may observe that the results of the proposed approach, i.e., when the training set is augmented with occluded samples, typically leads to higher accuracy and F_1 score. However, this result was not anticipated prior to the experimental evaluation. Of course, since it is well-known that data augmentation may typically boost performance of deep architectures, this result is adequately explained this way.

5.5 Conclusions

In this Chapter we dealt with the problem of HAR in presence of occlusion, which still remains an aspect that has not yet received significant attendance, while it compromises performance. Our goal was twofold: a) we intended to extensively study the effect of

occlusion in human activity recognition tasks; and b) we aimed to assess the effect of the inclusion of occluded samples within the training process. As expected, occlusion is responsible for loss of parts of motion, which in many cases are crucial for the recognition of an activity. Since to the best of our knowledge there do not exist any publicly available data sets, we worked using artificially created occluded data and under the hypothesis that one or more body parts remain occluded during the whole duration of the activity. Of course this is often a harder case than the ones that are encountered in typical, real-life occlusion scenarios, wherein occlusion often does not affect a whole body part, but some of its joints, instead. Also, in such scenarios occlusion is not continuous. Some parts may remain occluded for some time intervals. Thus, to our belief the proposed approach consists a harder problem than typical occlusion cases.

To tackle occlusion we followed a typical recognition approach. We first created 2-D representations of human joint motion. Then we trained a Convolutional Neural Network both with and without artificially occluded samples. We evaluated our approach in typical human activity recognition scenarios, i.e., single-view, cross-view and cross-subject, using two publicly available, large-scale datasets and upon considering 8 occlusion cases involving 1 or 2 arms and/or legs. Prior to the experimental evaluation we expected that (a) the inclusion of occluded samples will boost the performance of recognition in presence of occlusion; and (b) since most activities depend mainly on arms' motion, recognition will be mostly affected in cases of arms' occlusion. Both the aforementioned hypotheses have been experimentally proved. To our surprise, we also observed that augmentation of the training set with occluded samples led to a boost of performance. We consider this as the most important finding of this Chapter, since it may consist a simple and effective approach for data augmentation in future works.

6. A DEEP REGRESSION APPROACH FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION

6.1 Introduction

As we have already mentioned in Chapter 1, in real-world scenarios, Human Activity Recognition (HAR) from video data is often hindered by the occlusion of one or more body parts of the human subjects involved. Despite the fact that the recognition of most activities relies heavily on the motion of certain body parts, which when occluded can compromise the performance of recognition approaches, this issue is often overlooked in current research. In this Chapter, we propose an approach for HAR in the presence of partial occlusion, in cases wherein up to two body parts are involved. As in Chapter 5, we assume that human motion is modeled using a set of 3-D skeletal joints and that occluded body parts remain occluded throughout the activity. However, in this Chapter we address this problem using regression, performed by a novel deep Convolutional Recurrent Neural Network (CRNN). Specifically, given a partially occluded skeleton, we attempt to reconstruct the missing information regarding the motion of its occluded part(s).

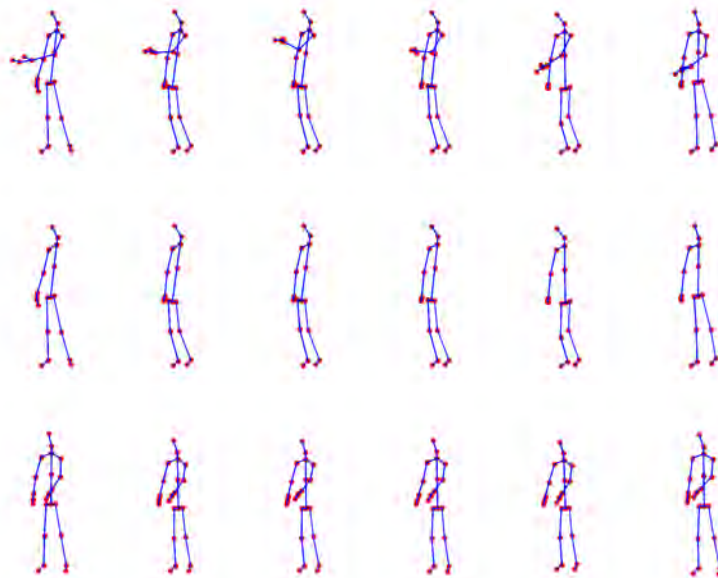
6.2 Methodology

6.2.1 Skeletal Data

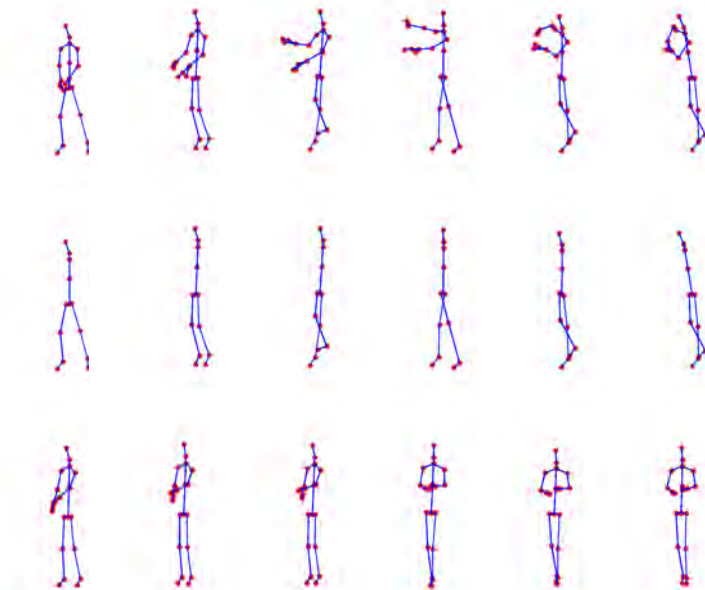
As in previous Chapters, the proposed approach uses as input 3D trajectories of human skeletons 2.8. In the context of this work, an activity is considered to be a temporal sequence of 3D skeleton representations. For the sake of explanation, in Fig. 6.1, we illustrate an example of a successfully reconstructed skeleton which leads to correct classification and an example of an unsuccessfully reconstructed skeleton, which leads to incorrect classification.

6.2.2 Occlusion

As it has already been mentioned in Section 6.1, partial occlusion may compromise the performance of HAR, in real-life scenarios, while its effects vary, depending on the activity performed. For example, the occlusion of both legs when the subject performs the action “kicking” results to a significant loss of visual information, which in turn may result to failure of recognition, while the occlusion of both arms is not expected to compromise recognition for this activity. As in the previous Chapter, we herein manually discard subsets of joints that correspond to body parts, assuming that these parts remain occluded during the whole duration of each activity (see Fig. 2.12). Moreover and for the sake of explanation, a visual example of an activity upon occlusion is illustrated in Fig. 6.1, where the loss of visual information is easily comprehensible.



(a)



(b)

Figure 6.1: Example skeleton sequences of the activities (a) *handshaking* and (b) *hugging other person* 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 (a) left arm; (b) both arms (see Fig. 2.12) have been discarded (i.e., the skeleton is partially occluded); Third row: skeletons have been reconstructed using the proposed deep regression approach. The example of (a) is successfully reconstructed and correctly classified, while the example of (b) is unsuccessfully reconstructed and incorrectly classified.

6.2.3 Regression of Skeletal Data

The input of our approach, as in Chapter 5 consists of temporal sequences of 3D skeleton data, i.e., as described in subsection 2.8. Upon imposing a linear interpolation step between consecutive time frames so as to address temporal variability of activities, we set the length of all activity examples equal to T_m , which denotes the size of the longest one in duration. Note that, if the desired length is not reached upon one interpolation step, the process is repeated until the desired length is reached. As we will mention in Section 6.3, we use datasets that have been captured either using one or three cameras. In the latter case and as we wish to exploit all existing information, we use the corresponding 3 skeleton sequences as input. We also assume that in every case of occlusion, the same missing body part(s) is (are) occluded in all available cameras.

The core philosophy of the herein proposed approach is that since occlusion practically causes missing values (i.e., in our case some of joints of the skeleton have been removed), we may formulate the problem of “reconstructing” those missing values as a *regression* task. More specifically, let $\mathbf{X}$ denote the original skeleton sequence and $\mathbf{X}_o$ the sequence resulting upon occlusion. The goal of regression is ideally to estimate a set of parameters β of a given function f , so that $\mathbf{X} = f(\mathbf{X}_o, \beta) + \epsilon$, where ϵ is some error value, to be minimized.

To this goal, we use a Convolutional Recurrent Neural Network (CRNN) model, whose aim is to implement f and learn β (i.e., its weights), in order to minimize ϵ . Given an occluded skeleton sequence $\mathbf{X}_o$, the network outputs the reconstructed skeleton sequence $\mathbf{X}_r$, which is an estimate for $\mathbf{X}$, upon completing missing (occluded) data (joints). The architecture of the network is as follows: the input of the network constitutes of sequential data from 1 or 3 cameras, depending on the dataset used, as it has already been discussed. We now describe the CRNN architecture for the case of single-camera datasets: in all cases, the duration of each activity is set equal to T_m , each skeleton comprises 20 3-D joints (when using Kinect v1), i.e., 60 co-ordinates, in total. Thus, the input layer size is $T_m \times 60$. This is filtered by a stack of 2 2-D convolutional layers, followed by a max-pooling layer that performs 1×2 sub-sampling. This single tensor is again filtered by a stack of 2 2-D convolutional layers, followed by a max-pooling layer that performs 1×2 sub-sampling. The output of this layer constitutes the input to an LSTM layer, whose goal is to harness temporal information of skeletal data. Then, a dense layer(s) of size $T_m \times 60$ follows and is ultimately reshaped to a $T_m \times 60$ output layer. In case of three-camera datasets, the input of the network constitutes of sequential data from 3 cameras, each providing a skeletal sequence under a different viewpoint. Those three input branches are in this case each filtered by a stack of 2 2-D convolutional layers, followed by a max-pooling layer that performs 1×2 sub-sampling. This process repeats after the three branches are concatenated into a single tensor. This single tensor is again filtered by a stack of 2 2-D convolutional layer, followed by a max-pooling layer that performs 1×2 sub-sampling. The output of this layer also in this case constitutes the input to an LSTM layer, and then, 3 parallel dense layers follow. They are ultimately reshaped to three output layers. The dimension of the kernels of all convolutional layers is 3×3 . For loss computation, in both

cases, the Mean Square Error (MSE) has been used. In Figs. 6.2 and 6.3 we illustrate the architectures of the CRNN and the LSTM networks, respectively and for the three-camera case.

At this point we would like to note that the reason for the use of an asymmetrical kernel (i.e., 1×2) is that while being sub-sampled this way, information is compressed only along the spatial coordinates' axes, leaving temporal information intact. We experimentally verified that this kernel choice led to a significant improvement in the performance of the network. In case of three-camera datasets, input from each camera is independently filtered by a stack of 2 2-D convolutional layers, followed by a max-pooling layer that performs 1×2 sub-sampling, then these three branches are concatenated into a single tensor and the network's architecture remains the same as in the one-camera case.

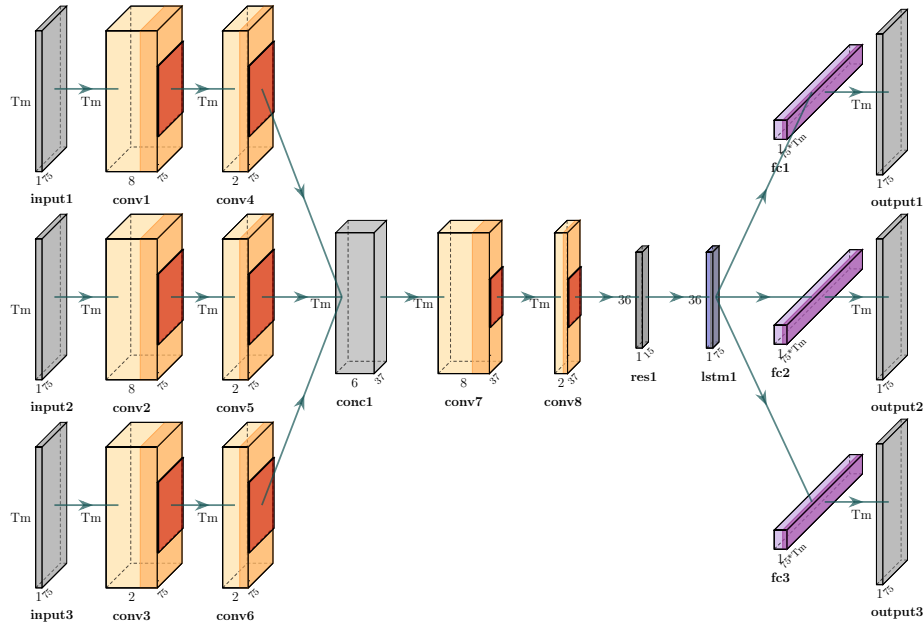


Figure 6.2: The architecture of the CRNN, for the three-camera case.

The occluded data $\mathbf{X}_o$ are given as input in both training and testing phases of the network. Also, the targets of the network are the non-occluded data $\mathbf{X}$; these data are to be estimated by the network, i.e., its output are reconstructed data $\mathbf{X}_r$. Thus, the network is trained to learn β , while minimizing ϵ . As we mentioned in subsection 2.8, in case of using the Kinect SDK in order to extract the skeleton data, each skeleton joint is represented by its own ID. Therefore, in a real-life application, one could easily continuously observe the values of the 3D coordinates and consider those joints with missing/zero values as occluded. This way, the specific case of occlusion could be easily recognized. Bearing this in mind, and upon initial experiments wherein we used a single network for all cases, which demonstrated unsatisfactory results, we finally opted to train one network per occlusion case, ending up with 8 different networks. Therefore, given an input skeletal sequence, first missing joints are identified and then, it is fed to the appropriate trained

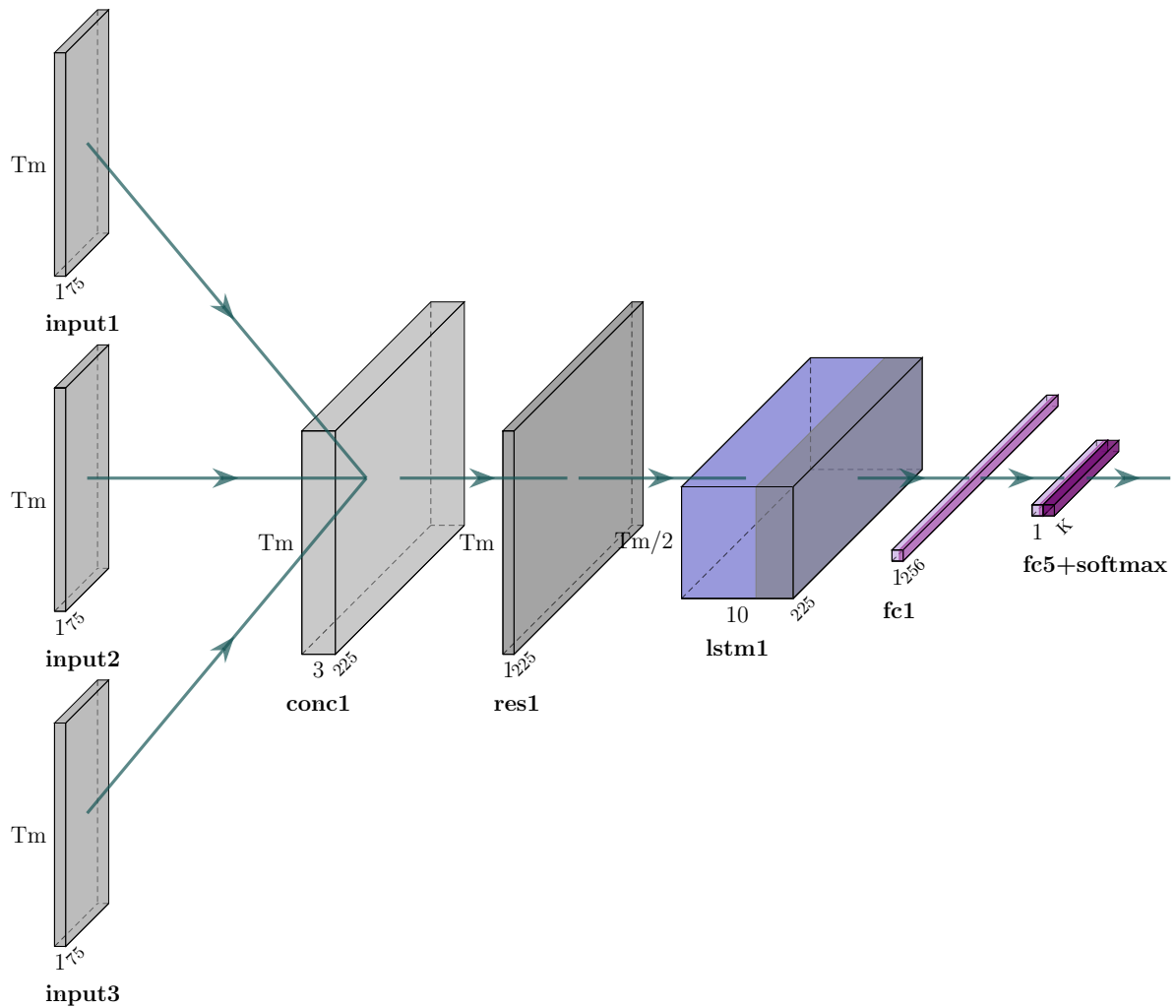


Figure 6.3: The architecture of the LSTM network, for the three-camera case.

network so as to be classified into an activity class, accordingly. Although this approach requires much more memory, its significant increase of performance compared to the use of a single network compensates this requirement.

At this point, an occluded sample (i.e., with missing skeletal joints due to occlusion) may be used as input into the trained CRNN network, which in turn reconstructs the missing skeletal data. For the sake of explanation, a visual example of an activity prior to and upon reconstruction is illustrated in Fig. 6.1. Upon its reconstruction, we are then able to proceed with the classification of the aforementioned into one of the pre-defined classes. For classification we use an LSTM network with 1 layer. Note that the CRNN network is trained using only the non-occluded data $\mathbf{X}$, thus without with any occlusion information. On the contrary, during the testing phase, the input of the LSTM network are the corresponding skeletal sequences $\mathbf{X}_r$, which have been reconstructed by $\mathbf{X}$. A visual overview

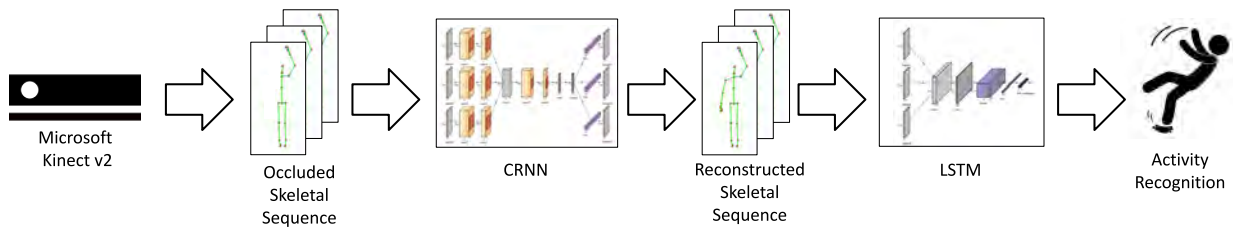


Figure 6.4: A visual overview of the proposed approach.

of the proposed approach is illustrated in Fig. 6.4.

6.3 Network Training

For the training of the estimator, we used the LeakyReLU activation function, except from the LSTM layer wherein the tanh function was used, and the last dense layer wherein linear activation function was used. For the training of the classifier, the LeakyReLU and tanh activation functions were used respectively, except from the last layer, wherein the softmax activation function was used. Moreover, we set the batch size to 5 and 10 for the training of the classifier and the estimator respectively. The Adam optimizer was utilized in both cases, the dropout was set to 0.3, the learning rate to 0.001 and we trained for 50 epochs, using the loss of the validation set calculated via MSE as an early stopping method, in order to prevent overfitting. In all cases, we used 80% of the available data for training, 10% for validation and the remaining 10% for testing.

6.4 Evaluation Protocol and Results

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

- a. removal of structured sets of skeletal joints, corresponding to body parts, to simulate occlusion (see Fig. 2.12). Specifically, we removed (a) left arm; (b) right arm; (c) both arms; (d) left leg; (e) right leg; (f) both legs; (g) left arm & left leg; (h) right arm & right leg. In this experiment we evaluate the performance of classification of reconstructed samples, using an LSTM network that has been trained using non-occluded samples, while reconstruction takes place using the CRNN,
- b. a baseline approach, wherein the LSTM is trained and evaluated using non-occluded samples,
- c. a “reference” approach wherein the LSTM is trained using non-occluded samples and validated using occluded samples, and

- d. inclusion of occluded samples in the training process of the LSTM and validation using occluded samples. In that case, a subset equal to 10% of the non-occluded samples of training set is selected. From those samples we create all 8 cases of occlusion, therefore we “augment” the initial training set by 80%. Note that in this case and contrary to the first of the aforementioned cases, we used a single network for all 8 cases of occlusion.

Experimental results for all datasets are depicted in Tables 6.1 – 6.4. In all cases we extract the following metrics: accuracy per class, F_1 score per class and weighted accuracy, where class weights are calculated based on the class distribution. Moreover, confusion matrices for all datasets in case of the baseline experiment are depicted in Fig. 6.5, while in case of classification of reconstructed samples are depicted in Figs. 6.6 – 6.9.

In case of the **PKU-MMD dataset**, the weighted accuracy (WA) was 0.92 without any body part removal. Specifically, it ranged between 0.21–0.90 in case of some body part removal, while it ranged between 0.70–0.91 upon reconstruction. In 7 out of 8 cases, significant improvement was observed, in terms of WA, while performance was almost equal in case of removal of Left Leg. Moreover, reconstruction outperformed occlusion in 6 out of 8 cases. Intuitively, one should observe that the majority of the activities we used to evaluate our approach mainly consists of upper body motion (i.e., left and/or right arm). Upon careful observation of the samples of the datasets, this assumption has been verified. This is also reflected to the results of Table 6.1, wherein it may be observed that in cases of occluded arms the improvement is significantly large, with most notable example the case of both arms, wherein WA improves from 0.21 to 0.70. Upon careful observation of the confusion matrices depicted in Fig. 6.6, for each occlusion case we should notice the following, when comparing with the case where all joints had been used: a) in case of any occluded arm, class *make a phone call/answer phone* is often confused with *playing with phone/tablet* and class *eat meal/snack* is often confused with *reading*; b) in case of occluded left leg, class *wear jacket* is often confused with *reading* or *standing up*; and c) finally, in case of both arms occluded, 7 classes show adequate performance. Upon observing the performance of individual classes, we should highlight the following: in the baseline experiment, all classes are recognized with high accuracies/ F_1 scores. Exception in this case is the class *make a phone call answer phone*, in case of occluded arm(s) which demonstrates extremely small values of both these metrics, as expected.

In case of **NTU-RGB+D dataset**, the WA was 0.68 without any body part removal and ranged between 0.16–0.59 in case of some body part removal, while it ranged between 0.53–0.62 upon reconstruction. Also in that case, in 7 out of 8 cases, significant improvement was observed, in terms of WA, while performance was almost equal in case of removal of Left Arm. Moreover, reconstruction outperformed occlusion in 7 out of 8 cases. Since activities used to evaluate the proposed approach mainly consisted of upper body motion, also in the case of the NTU-RGB+D dataset, in the results of Table 6.2 it could be observed that in all the remaining cases of occluded arms, the improvement of WA is large, with the most notable example the case of both arms, wherein WA improves from 0.21 to 0.70. Upon careful observation of the confusion matrices depicted in Fig. 6.6 and considering the classification results without occlusion of any part, for each occlusion case

we should notice that occlusion mainly affects activities *back pain*, *chest pain*, *neck pain*, *nausea/vomiting* and *fan self*. Note that these activities are performed in a significantly different way than the remaining ones and also that those of PKU-MMD, i.e., joints involved perform less movements. However, 7 classes in presence of occlusion consistently exhibit comparable performance to the case of absence of occlusion. Upon observing the performance of individual classes, we should highlight the following: in the baseline experiment, almost all classes are recognized with high accuracies/ F_1 scores. Exceptions in this case are the classes *staggering*, *chest pain* and *back pain*, in case of occluded arm(s) which demonstrate extremely small values of both these metrics, as expected.

In case of **SYSU-3D-HOI dataset**, the WA was 0.54 without any body part removal and ranged between 0.10–0.20 in case of some body part removal, while it ranged between 0.39–0.48 upon reconstruction. Moreover, reconstruction outperformed occlusion in 6 out of 8 cases. In every case of occlusion, significant improvement was observed, in terms of WA. We should herein notice that due to the small size of this dataset, performance was inadequate in several classes even without occlusion. As expected, worst performance was observed in case of occlusion of both arms. However, in the majority of cases, most classes in presence of occlusion consistently exhibit comparable performance to the case of absence of occlusion. Upon careful observation of the confusion matrices depicted in Fig. 6.8 and considering the classification results without occlusion of any part, for each occlusion case we should notice that occlusion mainly affects activities *pouring*, *calling phone*, *packing backpacks* and *taking from wallet*, which are based on the motion of arms.

In case of the **UTKinect-Action-3D dataset**, the WA was 0.79 without and body part removal, and ranged between 0.09–0.61 without any body part removal and ranged between 0.40–0.67 upon reconstruction. Moreover, reconstruction outperformed occlusion in 5 out of 8 cases. Also, in that case, in 7 out of 8 cases, significant improvement was observed, in terms of WA, while performance was equal in case of removal of Right Arm. Since activities used to evaluate the proposed approach mainly consisted of upper body motion, also in the case of the UTKinect-Action-3D dataset, in the results of Table 6.4 it could be observed that in all the remaining cases of occluded arms, the improvement of WA is large, with the most notable examples the cases of both arms, wherein WA improves from 0.09 to 0.40 and right arm/right leg wherein WA improves from 0.11 to 0.57. Upon careful observation of the confusion matrices depicted in Fig. 6.9 and considering the classification results without occlusion of any part, for each occlusion case we should notice that occlusion mainly affects activities *pickUp*, *carry*, *pull* and *clapHands*, although most activities that are based on the motion of arms are affected.

In **all cases**, occlusion of some body parts lead to a severe drop of performance; in many occasions, accuracy and F_1 score of some classes dropped to zero/near zero values. In some of these cases, reconstruction was successful, leading to near perfect recognition. Interestingly, addition of occluded samples within the training set, in certain cases leads to an improved performance compared to the baseline case. This has been observed in all 4 datasets and we believe that this is an effect of the augmentation of the training data set that we have previously mentioned. Finally, in several activities and for specific cases of occlusion, it is observed that classification of the reconstructed samples exhibits

slightly inferior performance compared to the occluded samples. It is our belief that in those cases, the specific body part which is occluded is less “relevant” for those activities, although in some cases the reconstruction approach may fail and lead to misleading joint positions, as in the example of Fig.6.1b.

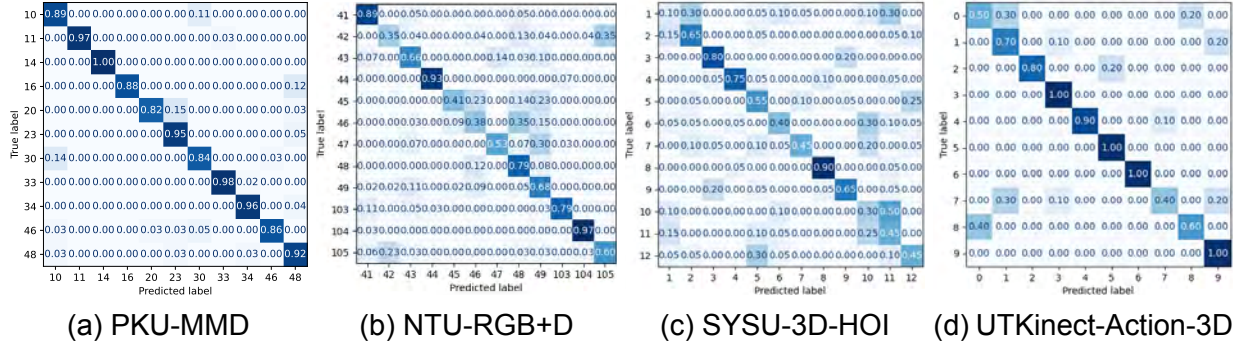


Figure 6.5: Normalized confusion matrices for classification for all datasets, without removing any body part.

Table 6.1: Results on PKU-MMD dataset. “Bas.” / “Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

		None				LA				RA				LA+RA				LL				RL				LL+RL				LA+LL				RA+RL			
Class	Metric	Bas.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.					
10	Acc.	0.90	0.84	0.43	0.11	0.45	0.36	0.03	0.58	0.64	0.00	0.79	0.71	0.92	0.79	0.50	0.66	0.87	0.71	0.74	0.68	0.79	0.21	0.66	0.71	0.24											
	F ₁	0.86	0.75	0.60	0.15	0.57	0.32	0.05	0.55	0.78	0.00	0.86	0.83	0.84	0.81	0.67	0.72	0.82	0.74	0.78	0.70	0.88	0.25	0.69	0.77	0.36											
11	Acc.	0.97	0.97	1.00	0.97	0.97	0.85	0.97	0.97	1.00	0.00	0.97	0.92	0.97	0.95	1.00	0.97	0.97	0.85	0.97	0.97	1.00	1.00	0.97	1.00	0.97											
	F ₁	0.99	0.99	0.96	0.90	0.97	0.88	0.99	0.95	1.00	0.00	0.91	0.92	0.99	0.97	1.00	0.99	0.99	0.88	0.99	0.97	0.96	0.70	0.99	1.00	0.92											
14	Acc.	1.00	0.88	1.00	1.00	1.00	0.29	1.00	1.00	1.00	0.81	1.00	1.00	0.94	1.00	0.86	1.00	1.00	0.57	1.00	1.00	0.94	1.00	0.29	1.00												
	F ₁	1.00	0.93	0.58	0.97	1.00	0.27	1.00	0.97	0.67	0.90	0.97	0.64	0.94	1.00	0.71	0.91	1.00	0.62	0.91	1.00	0.78	0.91	0.97	0.33	0.86											
16	Acc.	0.88	0.94	0.14	0.88	0.94	0.14	0.88	0.88	0.43	0.69	1.00	0.29	0.88	1.00	0.57	0.88	0.94	0.57	0.88	0.94	0.57	0.81	1.00	0.43	0.88											
	F ₁	0.93	0.91	0.25	0.90	0.94	0.25	0.93	0.90	0.50	0.76	0.97	0.44	0.93	0.97	0.73	0.90	0.97	0.67	0.90	0.97	0.62	0.79	1.00	0.50	0.85											
20	Acc.	0.82	0.18	0.91	0.03	0.61	0.73	0.39	0.00	0.73	0.00	0.85	1.00	0.79	0.82	0.73	0.88	0.79	0.55	0.88	0.30	0.91	0.00	0.39	0.64	0.97											
	F ₁	0.89	0.31	0.57	0.06	0.71	0.47	0.19	0.00	0.59	0.00	0.90	0.61	0.87	0.86	0.67	0.62	0.87	0.55	0.90	0.45	0.71	0.00	0.57	0.45	0.27											
23	Acc.	0.95	0.93	0.24	0.98	0.95	0.35	0.83	0.98	0.59	0.05	0.95	0.18	0.93	0.95	0.82	0.02	0.93	0.94	0.05	0.91	0.59	1.00	0.98	0.53	0.05											
	F ₁	0.90	0.74	0.38	0.70	0.83	0.44	0.40	0.56	0.69	0.05	0.92	0.30	0.93	0.91	0.85	0.05	0.91	0.84	0.09	0.75	0.71	0.62	0.80	0.56	0.07											
30	Acc.	0.84	0.60	0.86	0.70	0.87	0.21	0.24	0.16	0.93	0.00	0.97	0.86	0.76	0.87	0.93	0.89	0.89	0.79	0.84	0.81	0.93	0.30	0.76	0.86	0.70											
	F ₁	0.84	0.68	0.71	0.48	0.67	0.32	0.36	0.22	0.79	0.00	0.77	0.80	0.78	0.78	0.74	0.75	0.88	0.73	0.75	0.73	0.84	0.27	0.68	0.77	0.68											
33	Acc.	0.98	0.96	1.00	0.74	0.94	0.95	0.06	0.91	1.00	0.00	0.85	0.95	0.98	0.94	1.00	0.98	0.98	1.00	0.98	0.94	0.95	0.11	0.96	1.00	0.00											
	F ₁	0.98	0.96	1.00	0.84	0.96	0.92	0.11	0.94	1.00	0.00	0.91	0.95	0.98	0.95	1.00	0.98	0.98	0.97	0.98	0.96	0.97	0.20	0.97	1.00	0.00											
34	Acc.	0.96	0.89	0.95	0.96	0.96	0.95	0.19	0.46	1.00	0.00	1.00	0.95	0.94	0.94	1.00	0.90	0.89	1.00	0.85	0.98	1.00	0.54	0.96	1.00	0.00											
	F ₁	0.96	0.91	0.97	0.96	0.94	0.97	0.32	0.89	1.00	0.00	0.93	0.97	0.95	0.94	1.00	0.93	0.92	1.00	0.91	0.96	1.00	0.70	0.95	1.00	0.00											
46	Acc.	0.87	0.84	1.00	0.89	0.87	1.00	0.49	0.84	1.00	0.65	0.87	1.00	0.84	0.87	1.00	0.87	0.87	1.00	0.87	0.84	1.00	0.87	0.87	1.00	0.65											
	F ₁	0.91	0.89	0.97	0.69	0.88	0.79	0.38	0.90	1.00	0.14	0.91	0.94	0.90	0.90	0.97	0.93	0.91	0.97	0.91	0.87	0.97	0.48	0.89	1.00	0.56											
48	Acc.	0.92	0.92	0.80	0.28	0.69	0.73	0.13	0.56	0.67	0.00	0.59	0.80	0.85	0.85	0.87	0.85	0.90	0.80	0.90	0.90	0.73	0.05	0.87	0.73	0.13											
	F ₁	0.88	0.78	0.89	0.43	0.76	0.76	0.22	0.58	0.80	0.00	0.74	0.86	0.87	0.87	0.93	0.72	0.80	0.89	0.74	0.86	0.85	0.10	0.88	0.85	0.23											
all	WA	0.92	0.82	0.78	0.68	0.84	0.65	0.40	0.70	0.83	0.21	0.89	0.80	0.90	0.90	0.87	0.80	0.91	0.84	0.80	0.84	0.87	0.48	0.86	0.80	0.41											

6.5 Conclusions

In this Chapter we presented an approach for HAR under partial occlusion, i.e., occlusion of one/two parts. Our approach was based on the motion of 3D skeletal joints and occlusion was simulated upon manually removing structured sets of joints which corresponded

Table 6.2: Results on NTU-RGB+D dataset. “Bas.” / “Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

		None				LA				RA				LA+RA				LL				RL				LL+RL				LA+LL				RA+RL			
Class	Metric	Bas.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.					
41	Acc.	0.89	0.70	0.87	0.73	0.51	0.74	0.14	0.57	0.84	0.11	0.73	0.84	0.24	0.65	0.81	0.62	0.87	0.74	0.24	0.70	0.84	0.16	0.73	0.71	0.11					0.73	0.71	0.11				
	F ₁	0.82	0.66	0.73	0.74	0.54	0.65	0.19	0.58	0.79	0.11	0.70	0.71	0.35	0.66	0.74	0.59	0.78	0.73	0.38	0.68	0.80	0.24	0.73	0.73	0.19					0.73	0.73	0.19				
42	Acc.	0.35	0.22	0.47	0.13	0.17	0.34	0.26	0.26	0.41	0.00	0.22	0.41	0.09	0.26	0.44	0.09	0.30	0.44	0.00	0.13	0.41	0.00	0.22	0.47	0.09					0.47	0.09					
	F ₁	0.40	0.29	0.42	0.17	0.20	0.33	0.31	0.32	0.39	0.00	0.28	0.40	0.14	0.32	0.40	0.15	0.36	0.41	0.00	0.19	0.39	0.00	0.28	0.45	0.14					0.45	0.14					
43	Acc.	0.66	0.62	0.59	0.72	0.66	0.53	0.83	0.76	0.69	0.62	0.69	0.47	0.52	0.76	0.69	0.76	0.76	0.69	0.72	0.72	0.69	0.62	0.79	0.69	0.55					0.79	0.69	0.55				
	F ₁	0.61	0.46	0.50	0.46	0.47	0.42	0.35	0.49	0.51	0.29	0.53	0.41	0.38	0.54	0.46	0.27	0.57	0.50	0.36	0.51	0.59	0.31	0.56	0.51	0.26					0.56	0.51	0.26				
44	Acc.	0.93	0.87	0.87	0.77	0.90	0.87	0.07	0.90	0.87	0.07	0.90	0.87	0.93	0.90	0.90	0.97	0.90	0.84	1.00	0.90	0.87	0.97	0.90	0.84	0.37					0.90	0.84	0.37				
	F ₁	0.95	0.90	0.89	0.87	0.93	0.89	0.12	0.95	0.89	0.12	0.93	0.87	0.93	0.95	0.90	0.95	0.93	0.88	0.85	0.93	0.87	0.89	0.93	0.88	0.54					0.93	0.88	0.54				
45	Acc.	0.41	0.18	0.28	0.46	0.23	0.19	0.91	0.23	0.28	0.41	0.23	0.25	0.14	0.23	0.34	0.59	0.50	0.22	0.50	0.18	0.41	0.14	0.41	0.16					0.77							
	F ₁	0.51	0.22	0.27	0.40	0.24	0.23	0.26	0.23	0.25	0.23	0.26	0.25	0.17	0.26	0.29	0.32	0.47	0.26	0.23	0.21	0.38	0.19	0.40	0.19	0.23					0.23						
46	Acc.	0.38	0.32	0.47	0.24	0.27	0.47	0.12	0.35	0.56	0.27	0.50	0.53	0.32	0.29	0.44	0.15	0.29	0.50	0.09	0.35	0.59	0.12	0.35	0.66	0.03					0.66	0.03					
	F ₁	0.43	0.39	0.35	0.31	0.30	0.33	0.17	0.39	0.41	0.31	0.47	0.36	0.37	0.33	0.38	0.21	0.35	0.36	0.14	0.37	0.41	0.20	0.40	0.42	0.05					0.42	0.05					
47	Acc.	0.53	0.40	0.42	0.63	0.27	0.42	0.27	0.27	0.58	0.07	0.33	0.45	0.33	0.27	0.52	0.37	0.33	0.65	0.13	0.40	0.71	0.33	0.33	0.65	0.43					0.65	0.43					
	F ₁	0.64	0.53	0.55	0.73	0.39	0.55	0.28	0.38	0.65	0.11	0.45	0.56	0.43	0.39	0.58	0.48	0.45	0.63	0.21	0.51	0.70	0.43	0.45	0.63	0.36					0.63	0.36					
48	Acc.	0.79	0.38	0.10	0.42	0.38	0.07	0.17	0.38	0.03	0.50	0.54	0.07	0.67	0.29	0.10	0.21	0.50	0.10	0.21	0.50	0.10	0.21	0.63	0.38	0.03	0.21					0.63	0.38				
	F ₁	0.55	0.31	0.15	0.47	0.32	0.10	0.18	0.31	0.06	0.14	0.39	0.11	0.31	0.24	0.16	0.22	0.44	0.17	0.14	0.32	0.12	0.32	0.32	0.06	0.15					0.32	0.06					
49	Acc.	0.68	0.48	0.44	0.48	0.46	0.38	0.05	0.36	0.50	0.00	0.46	0.28	0.77	0.48	0.56	0.16	0.48	0.44	0.41	0.46	0.38	0.59	0.52	0.41	0.23					0.59	0.52					
	F ₁	0.59	0.42	0.44	0.47	0.41	0.39	0.07	0.36	0.42	0.00	0.44	0.35	0.46	0.42	0.47	0.24	0.45	0.38	0.31	0.43	0.40	0.40	0.46	0.37	0.26					0.46	0.37					
103	Acc.	0.79	0.76	0.72	0.79	0.82	0.78	0.40	0.76	0.53	0.16	0.76	0.75	0.74	0.84	0.50	0.74	0.82	0.69	0.61	0.79	0.72	0.74	0.79	0.72	0.32					0.79	0.72					
	F ₁	0.85	0.79	0.81	0.79	0.84	0.83	0.54	0.79	0.68	0.26	0.82	0.81	0.84	0.84	0.67	0.85	0.83	0.77	0.72	0.85	0.79	0.82	0.83	0.81	0.48					0.83	0.81					
104	Acc.	0.97	0.94	0.75	0.74	0.94	0.75	0.21	0.88	0.69	0.00	0.88	0.84	0.21	0.94	0.63	0.15	0.88	0.78	0.03	0.94	0.78	0.12	0.91	0.81	0.32					0.91	0.81					
	F ₁	0.96	0.93	0.81	0.82	0.90	0.76	0.34	0.88	0.79	0.00	0.90	0.84	0.33	0.93	0.75	0.24	0.90	0.86	0.06	0.91	0.83	0.21	0.89	0.85	0.48					0.89	0.85					
105	Acc.	0.60	0.66	0.34	0.77	0.51	0.38	0.11	0.63	0.22	0.00	0.57	0.47	0.14	0.60	0.16	0.11	0.60	0.25	0.00	0.69	0.28	0.20	0.54	0.28	0.03					0.54	0.28					
	F ₁	0.66	0.64	0.37	0.65	0.54	0.38	0.17	0.65	0.27	0.00	0.61	0.46	0.24	0.62	0.22	0.19	0.65	0.31	0.00	0.69	0.32	0.26	0.60	0.35	0.05					0.60	0.35					
all	WA	0.68	0.57	0.53	0.59	0.53	0.49	0.27	0.55	0.52	0.16	0.59	0.52	0.44	0.57	0.51	0.41	0.62	0.53	0.33	0.58	0.56	0.40	0.59	0.53	0.25					0.59	0.53					

Table 6.3: Results on SYSU-3D-HOI dataset. “Bas.”/“Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

		None				LA				RA				LA+RA				LL				RL				LL+RL				LA+LL				RA+RL			
Class	Metric	Baseline	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.					
1	Acc.	0.10	0.00	0.13	0.00	0.05	0.13	0.00	0.00	0.25	0.00	0.00	0.10	0.00	0.10	0.28	0.00	0.05	0.13	0.00	0.00	0.28	0.00	0.00	0.28	0.00	0.00	0.05	0.00								
	F ₁	0.10	0.00	0.14	0.00	0.06	0.07	0.00	0.00	0.25	0.00	0.00	0.11	0.00	0.10	0.28	0.00	0.08	0.11	0.00	0.00	0.28	0.00	0.00	0.28	0.00	0.00	0.06	0.00								
2	Acc.	0.65	0.45	0.43	0.00	0.65	0.13	0.05	0.00	0.50	0.00	0.45	0.40	0.45	0.55	0.53	0.20	0.55	0.28	0.10	0.50	0.48	0.30	0.00	0.28	0.00					0.28	0.00					
	F ₁	0.59	0.36	0.46	0.00	0.63	0.12	0.08	0.00	0.54	0.00	0.44	0.45	0.34	0.56	0.57	0.23	0.53	0.34	0.04	0.48	0.49	0.16	0.00	0.19	0.00					0.19	0.00					
3	Acc.	0.80	0.75	0.60	0.55	0.85	0.48	0.05	0.60	0.68	0.45	0.70	0.58	0.00	0.85	0.68	0.00	0.80	0.48	0.00	0.80	0.48	0.00	0.60	0.65	0.25	0.55	0.55	0.10								
	F ₁	0.77	0.69	0.54	0.21	0.77	0.44	0.05	0.57	0.59	0.12	0.65	0.56	0.00	0.71	0.59	0.00	0.73	0.46	0.00	0.56	0.59	0.18	0.61	0.50	0.06					0.61	0.50	0.06				
4	Acc.	0.75	0.70	0.58	0.05	0.80	0.53	0.30	0.75	0.65	0.00	0.75	0.68	0.40	0.75	0.65	0.25	0.70	0.58	0.20	0.75	0.70	0.25	0.80	0.68	0.25					0.80	0.68	0.25				
	F ₁	0.80	0.73	0.61	0.08	0.84	0.57	0.43	0.76	0.71	0.00	0.79	0.70	0.52	0.78	0.72	0.34	0.70	0.62	0.29	0.75	0.76	0.27	0.76	0.66	0.31					0.76	0.66	0.31				
5	Acc.	0.55	0.45	0.28	0.10	0.40	0.53	0.40	0.50	0.40	0.00	0.45	0.38	0.40	0.40	0.40	0.00	0.50	0.25	0.20	0.30	0.33	0.50	0.50	0.30	0.10					0.50	0.30	0.10				
	F ₁	0.48	0.34	0.21	0.10	0.35	0.40	0.15	0.30	0.38	0.00	0.38	0.32	0.10	0.39	0.41	0.00	0.39	0.23	0.19	0.28	0.32	0.20	0.44	0.31	0.07					0.44	0.31	0.07				
6	Acc.	0.40	0.25	0.28	0.05	0.25	0.05	0.00	0.00	0.30	0.00	0.30	0.33	0.00	0.35	0.28	0.00	0.25	0.25	0.00	0.30	0.33	0.00	0.25	0.43	0.00					0.43	0.00					
	F ₁	0.38	0.31	0.27	0.08	0.28	0.03	0.00	0.00	0.28	0.00	0.27	0.31	0.00	0.32	0.26	0.00	0.18	0.22	0.00	0.27	0.30	0.00	0.20	0.38	0.00					0.38	0.00					
7	Acc.	0.45	0.40	0.43	0.50	0.35	0.33	0.30	0.15	0.65	0.00	0.40	0.43	0.00	0.45	0.68	0.00	0.45	0.45	0.00	0.35	0.55	0.20	0.35	0.53	0.00					0.53	0.00					
	F ₁	0.54	0.35	0.39	0.18	0.36	0.29	0.19	0.20	0.61	0.00	0.47	0.42	0.00	0.46	0.60	0.00	0.42	0.42	0.00	0.41	0.52	0.06	0.40	0.43	0.00					0.43	0.00					
8	Acc.	0.90	0.85	0.80	0.10	0.85	0.75	0.65	0.75	0.75	0.00	0.90	0.78	0.80	0.95	0.80	0.60	0.95	0.73	0.60	0.90	0.78	0.50	0.85	0.80	0.35					0.85	0.80	0.35				
	F ₁	0.88	0.81	0.79	0.13	0.83	0.74	0.71	0.77	0.77	0.00	0.84	0.79	0.75	0.87	0.79	0.58	0.88	0.74	0.63	0.88	0.81	0.56	0.80	0.84	0.46					0.84	0.46					
9	Acc.	0.65	0.30	0.35	0.50	0.25	0.30	0.40	0.20	0.33	0.80	0.40	0.40	0.00	0.35	0.38	0.00	0.35	0.28	0.00	0.40	0.35	0.10	0.25	0.30	0.65					0.25	0.30					
	F ₁	0.67	0.39	0.31	0.19	0.31	0.28	0.31	0.28	0.34	0.19	0.47	0.30	0.00	0.45	0.39	0.00	0.38	0.24	0.00	0.44	0.31	0.12	0.26	0.25	0.12					0.26	0.25	0.12				
10	Acc.	0.30	0.15	0.33	0.00	0.30	0.13	0.05	0.00	0.28	0.00	0.30	0.40	0.00	0.30	0.28	0.00	0.25	0.33	0.30	0.00	0.30	0.30	0.05	0.25	0.15	0.05					0.25	0.15	0.05			
	F ₁	0.24	0.12	0.27	0.00	0.21	0.07	0.06	0.00	0.22	0.00	0.20	0.29	0.00	0.22	0.21	0.00	0.18	0.26	0.00	0.31	0.24	0.06	0.20	0.12	0.08					0.20	0.12	0.08				
11	Acc.	0.45	0.40	0.30	0.00	0.45	0.25	0.10	0.30	0.35	0.00	0.60	0.20	0.00	0.50	0.33	0.15	0.55	0.33	0.00	0.25	0.28	0.00	0.50	0.23	0.00					0.50	0.23	0.00				
	F ₁	0.34	0.31	0.24	0.00	0.32	0.14	0.10	0.23	0.26	0.00	0.43	0.15	0.00	0.44	0.25	0.10	0.39	0.31	0.00	0.17	0.21	0.00	0.29	0.19	0.00					0.29	0.19	0.00				
12	Acc.	0.45	0.35	0.40	0.00	0.25	0.40	0.35	0.55	0.23	0.00	0.30	0.35	0.40	0.20	0.35	0.90	0.10	0.38	0.80	0.50	0.23	0.20	0.35	0.20	0.10					0.35	0.20	0.10				
	F ₁	0.45	0.34	0.41	0.00	0.22	0.45	0.18	0.24	0.24	0.00	0.25	0.39	0.09	0.15	0.38	0.18	0.10	0.32	0.16	0.38	0.25	0.14	0.23	0.19	0.02					0.23	0.19	0.02				
all	WA	0.42	0.41	0.15	0.45	0.33	0.22	0.32	0.45	0.10	0.46	0.42	0.20	0.48	0.47	0.18	0.46	0.37	0.16	0.43	0.44	0.20	0.29	0.39	0.37	0.13					0.39	0.37	0.13				

Human Activity Recognition Under Occlusion

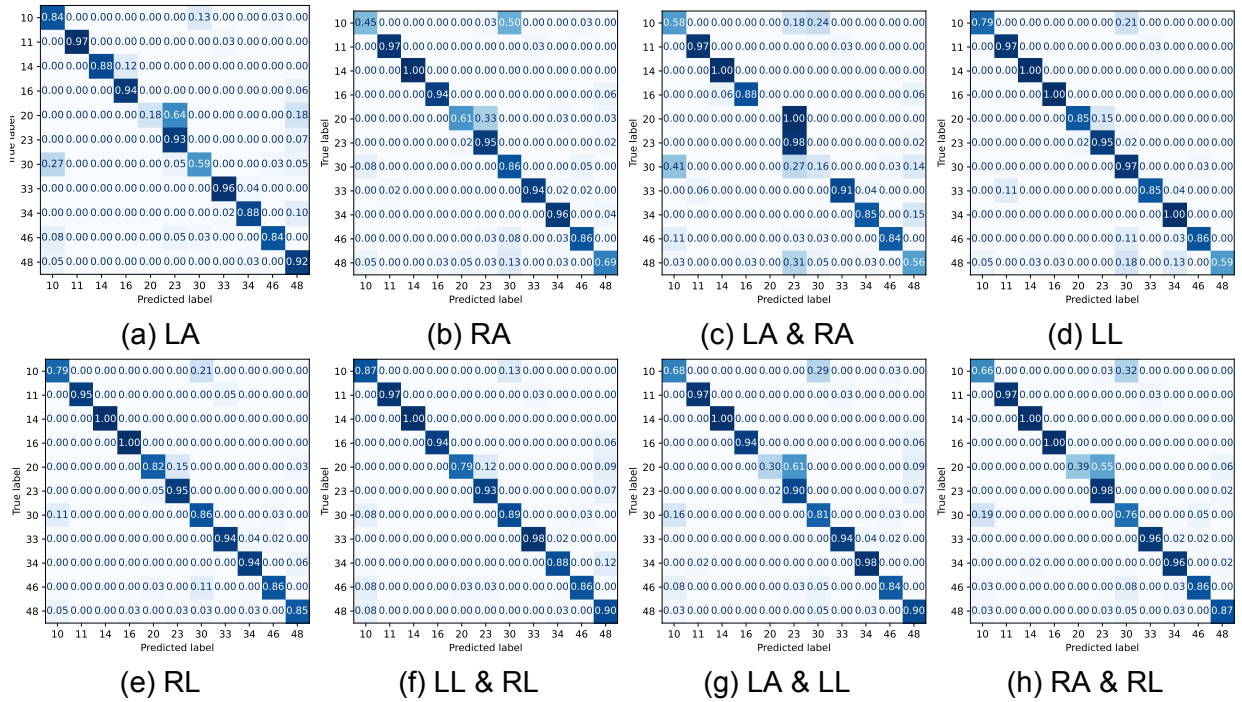


Figure 6.6: Normalized confusion matrices for classification for the PKU-MMD dataset, upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

to arms and/or legs. We managed to reconstruct the missing joints by formulating the problem as a regression task. Specifically, we trained a convolutional recurrent neural network using occluded and non-occluded samples and then used it to reconstruct missing joints of occluded samples. For classification of reconstructed samples, we used a long short term memory network. We then evaluated the proposed approach using four publicly available datasets. Among them, we experimented with several different types of activities, such as simple activities, activities of daily living, medical-related activities and interactions with objects. We showed that this way, we could achieve a significant boost of performance in a classification task, when using reconstructed samples instead of occluded ones, almost in any activity.

The novelties of this approach were as follows: (a) we formulated the problem of missing skeleton joints due to occlusion as a regression problem, within a human motion activity recognition task and implemented a novel deep neural network architecture to reconstruct missing skeletal joints; (b) contrary to other state-of-the-art works, we did not use occluded samples within the training process; (c) the occluded parts were not visible from any camera, even in case of a multi-camera setup.

Human Activity Recognition Under Occlusion

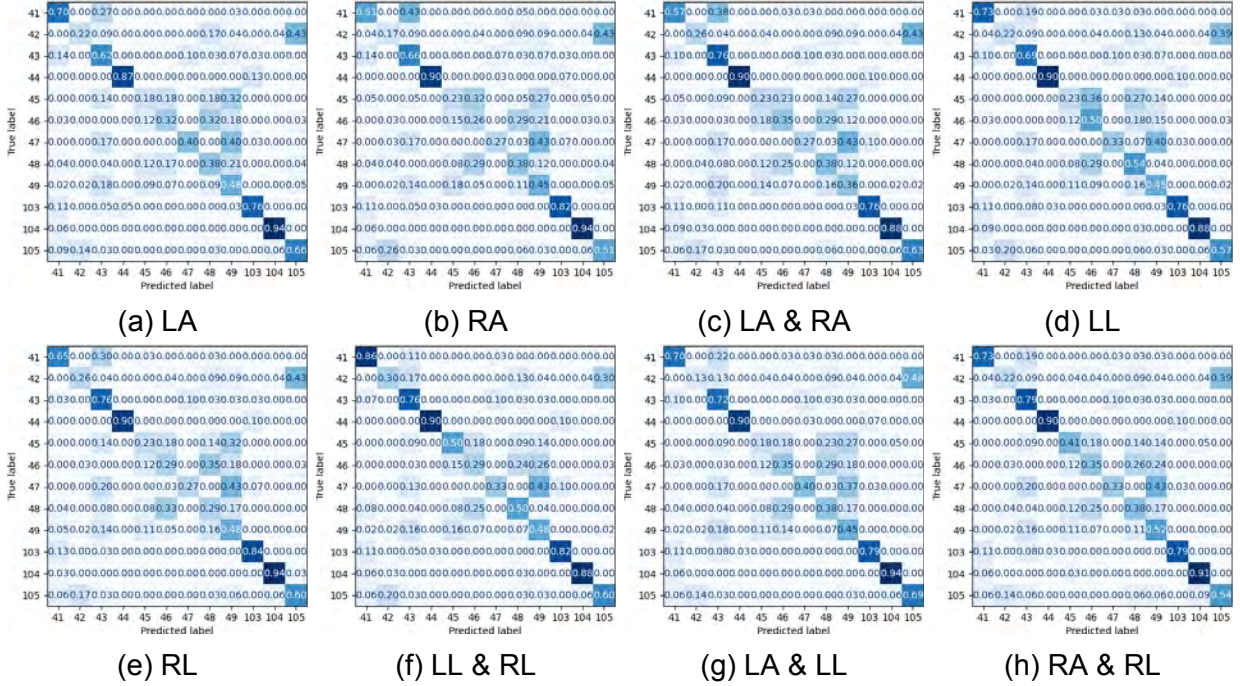


Figure 6.7: Normalized confusion matrices for classification for the NTU-RGB+D dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

Table 6.4: Results on UTKinect-Action3D dataset. “Bas.”/“Rec.”/“Occ.”/“Ref.” denote baseline / reconstructed / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

Class	Metric	None	LA			RA			LA+RA			LL			RL			LL+RL			LA+LL			RA+RL		
		Baseline	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.	Rec.	Occ.	Ref.
0	Acc.	0.50	0.40	0.30	0.30	0.40	0.30	0.00	0.40	0.30	0.00	0.40	0.30	0.20	0.40	0.40	0.50	0.40	0.30	0.20	0.40	0.50	0.70	0.40	0.40	0.00
	F ₁	0.48	0.43	0.35	0.31	0.43	0.35	0.00	0.39	0.26	0.00	0.43	0.31	0.09	0.43	0.41	0.41	0.43	0.28	0.07	0.43	0.44	0.33	0.39	0.43	0.00
1	Acc.	0.70	0.30	0.20	0.60	0.10	0.20	0.30	0.30	0.20	0.90	0.70	0.60	0.00	0.50	0.30	0.40	0.60	0.10	0.00	0.40	0.50	0.20	0.30	0.30	0.20
	F ₁	0.53	0.23	0.16	0.15	0.08	0.10	0.17	0.13	0.11	0.18	0.57	0.51	0.00	0.39	0.26	0.35	0.43	0.10	0.00	0.32	0.39	0.06	0.21	0.23	0.13
2	Acc.	0.80	0.80	0.70	0.00	0.80	0.50	0.00	0.90	0.80	0.00	0.80	0.80	0.10	0.70	0.80	0.60	0.70	0.60	0.00	0.90	0.90	0.20	0.90	0.40	0.00
	F ₁	0.87	0.76	0.66	0.00	0.80	0.41	0.00	0.83	0.76	0.00	0.80	0.67	0.13	0.75	0.66	0.63	0.75	0.46	0.00	0.79	0.85	0.27	0.89	0.26	0.00
3	Acc.	1.00	0.30	0.30	0.00	0.60	0.10	0.20	0.00	0.30	0.00	0.70	0.40	0.40	0.70	0.60	1.00	0.80	0.30	0.70	0.20	0.70	0.00	0.20	0.30	0.30
	F ₁	0.92	0.22	0.26	0.00	0.43	0.05	0.05	0.00	0.33	0.00	0.58	0.41	0.40	0.63	0.57	0.69	0.72	0.18	0.61	0.11	0.63	0.00	0.16	0.29	0.07
4	Acc.	0.90	0.00	0.20	0.00	0.40	0.10	0.60	0.00	0.50	0.00	0.60	0.20	0.90	0.60	0.60	0.50	0.60	0.10	0.80	0.10	0.50	0.00	0.30	0.10	0.60
	F ₁	0.93	0.00	0.11	0.00	0.40	0.13	0.14	0.00	0.41	0.00	0.57	0.13	0.73	0.60	0.57	0.39	0.57	0.08	0.59	0.08	0.46	0.00	0.20	0.13	0.14
5	Acc.	1.00	0.80	0.50	0.00	0.80	0.50	0.00	0.80	0.40	0.00	0.80	0.50	0.90	0.80	0.40	0.70	0.80	0.50	0.80	0.50	0.10	0.80	0.50	0.00	0.00
	F ₁	0.92	0.69	0.49	0.00	0.69	0.43	0.00	0.72	0.47	0.00	0.69	0.50	0.58	0.69	0.43	0.65	0.69	0.40	0.59	0.73	0.60	0.13	0.76	0.43	0.00
6	Acc.	1.00	0.80	0.50	0.00	0.80	0.70	0.00	0.90	0.80	0.00	0.90	0.80	1.00	0.80	0.80	0.90	0.80	0.70	0.80	0.80	1.00	0.20	1.00	0.70	0.00
	F ₁	1.00	0.80	0.49	0.00	0.80	0.66	0.00	0.93	0.61	0.00	0.93	0.72	0.79	0.76	0.68	0.81	0.76	0.63	0.51	0.83	0.86	0.13	0.96	0.63	0.00
7	Acc.	0.40	0.50	0.60	0.70	0.40	0.30	0.00	0.20	0.60	0.00	0.40	0.30	0.00	0.20	0.70	0.10	0.30	0.40	0.00	0.40	0.30	0.40	0.40	0.50	0.00
	F ₁	0.43	0.25	0.22	0.23	0.28	0.21	0.00	0.17	0.35	0.00	0.40	0.19	0.00	0.12	0.54	0.10	0.27	0.34	0.00	0.24	0.21	0.11	0.34	0.28	0.00
8	Acc.	0.60	0.40	0.50	0.00	0.40	0.30	0.00	0.20	0.50	0.00	0.40	0.60	0.50	0.40	0.60	0.40	0.40	0.50	0.30	0.30	0.50	0.20	0.40	0.40	0.00
	F ₁	0.52	0.36	0.44	0.00	0.36	0.29	0.00	0.16	0.47	0.00	0.27	0.55	0.40	0.36	0.55	0.43	0.29	0.53	0.27	0.26	0.53	0.27	0.36	0.44	0.00
9	Acc.	1.00	0.70	0.40	0.00	1.00	0.30	0.40	0.30	1.00	0.00	1.00	0.70	1.00	1.00	1.00	1.00	1.00	0.90	1.00	0.80	1.00	0.20	1.00	0.80	0.00
	F ₁	0.87	0.52	0.23	0.00	0.83	0.10	0.47	0.13	0.88	0.00	0.87	0.59	0.69	0.87	0.96	0.96	0.87	0.81	0.74	0.58	0.92	0.07	0.87	0.76	0.00
all	WA	0.79	0.50	0.42	0.16	0.57	0.33	0.15	0.40	0.54	0.09	0.67	0.52	0.50	0.61	0.62	0.61	0.64	0.44	0.46	0.51	0.64	0.22	0.57	0.44	0.11

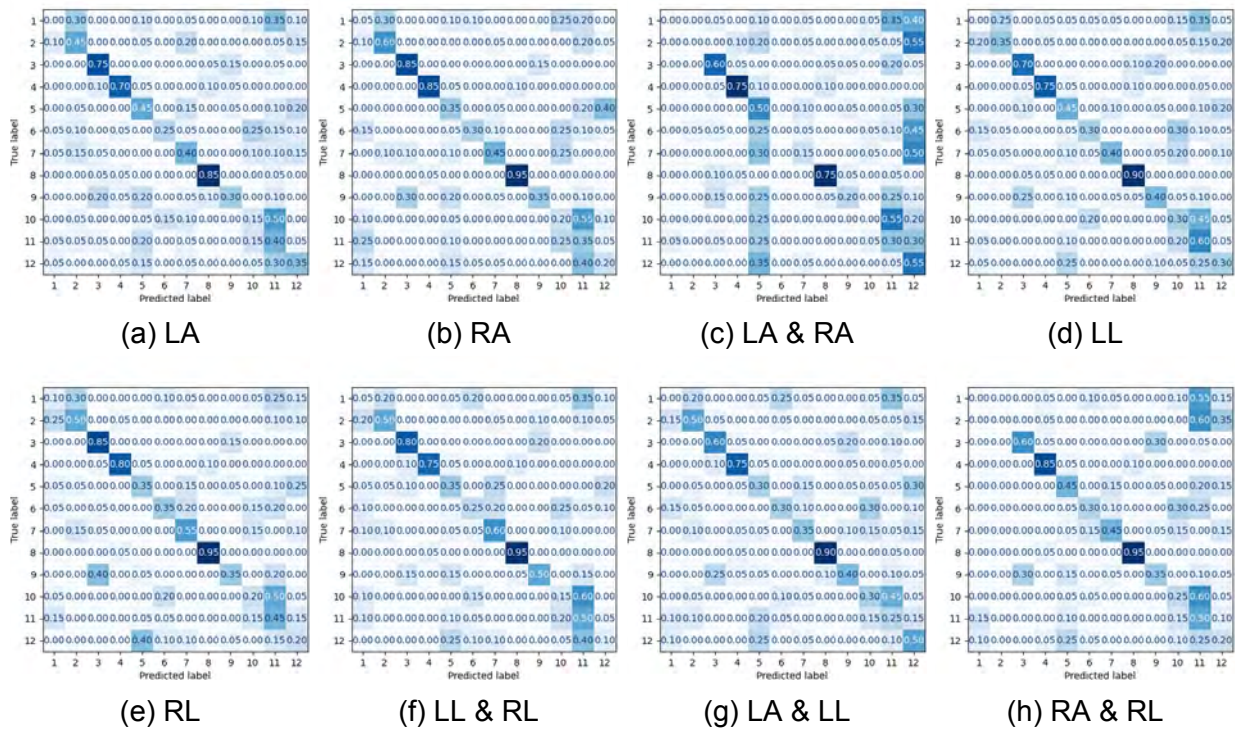


Figure 6.8: Average confusion matrices from five trials, for classification for the SYSU-3D-UOI dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

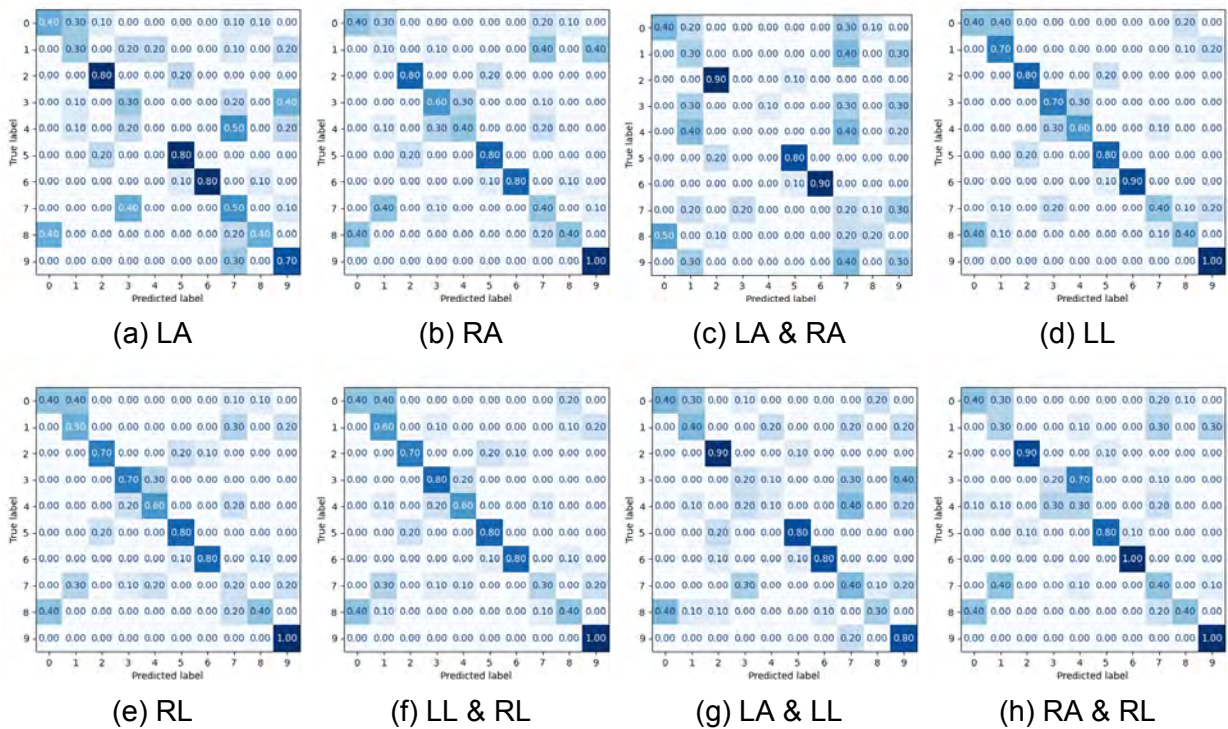


Figure 6.9: Average confusion matrices from five trials, for classification for the UTKinect-Action-3D dataset upon removing the body part(s) denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

7. USING GANS FOR HUMAN ACTIVITY RECOGNITION UNDER OCCLUSION

7.1 Introduction

Up to this point we have already made clear the importance of occlusion in real-life HAR scenarios. As we have seen in Chapters 5 and 6, two effective strategies for dealing with occlusion of one or more body parts are (a) augmentation of the training set with occluded samples and (b) deep regression on occluded samples, respectively. In this Chapter, we propose a novel approach for HAR in the presence of partial occlusion, which may be applied in cases wherein up to two body parts are involved, as in previous Chapters. This approach also works under the assumption that human motion is modeled using a set of 3-D skeletal joints and that occluded body parts remain occluded throughout the activity. Contrary to previous Chapters, in this Chapter we address this problem using a Generative Adversarial Network (GAN). Specifically, we train a Convolutional Recurrent Neural Network (CRNN) to serve as the generator of the GAN, with the aim of producing the complete skeleton by completing the missing parts. The input to the CRNN consists of raw 3D skeleton joint positions, with the joints corresponding to the occluded parts removed. The output of the CRNN is the reconstructed skeleton, which is then fed into a Long Short Term Memory (LSTM) network for classification into one of the predefined activities. For the discriminator we use a simple Long Short Term Memory (LSTM).

7.2 Related work

Over the last few years, there have been notable advancements in deep learning technology for image inpainting. Image inpainting involves restoring and reconstructing images using background information. The aim is to generate images that appear highly realistic and indistinguishable from the original. Deep learning-based image inpainting methods, particularly those leveraging GANs, have shown promising outcomes [196].

Yu et al. [214] introduced a deep generative model-based approach for image inpainting, namely Deepfillv1. This approach combines the strengths of deep learning algorithms with traditional methods to enhance the quality of generated images. Deepfillv1 is capable of automatically repairing images with multiple or large holes, resulting in higher quality images compared to previous methods. It leverages the surrounding image features to synthesize new image structures and utilizes a feed-forward and fully convolutional neural network to process images with multiple holes during testing. The authors also proposed a coarse-to-fine generative image inpainting framework with a novel contextual attention module, which improves the inpainting results by learning feature representations for explicit matching and attending to relevant background patches.

Dolhansky et al. [40] introduced a new technique for inpainting, called Exemplar GANs (ExGANs). Instead of using generic reference images, the authors utilize exemplar in-

formation as a reference image to inpaint a person with closed eyes in a natural image. This approach produces high-quality and personalized inpainting results. Additionally, ExGANs can represent the object with a perceptual code, allowing for the generation of photo-realistic and personalized images in terms of perception and semantics. ExGAN is a type of conditional GAN that enhances the descriptive capability by incorporating additional information at various points within the adversarial network. This method is particularly useful for image generation or inpainting tasks that involve the use of reference images or perceptual codes, as it yields superior perceptual outcomes.

Another image inpainting system called Deepfillv2 was proposed by Yu et al. [215]. This system utilizes deep learning and is capable of completing images using free-form masks and inputs. The method is built upon gated convolutions and is able to handle images with free-form masks of any shape or location. The authors also introduced a GAN loss called SN-PatchGAN, which contributes to faster and more stable training. Compared to previous methods, Deepfillv2 produces more flexible and higher-quality results. It can be employed to remove distracting objects, clear watermarks, edit faces, and fill in missing regions. Additionally, the system, which is based on an end-to-end generative network, can enhance inpainting outcomes with user guidance input.

Nazeri et al. [139] introduced a two-stage adversarial model called EdgeConnect, which offers a fresh approach to image inpainting. The EdgeConnect model is designed to fill in missing regions using hallucinated edges as a priori information. It comprises two components: an edge generator and an image completion network. Together, these components can generate filled regions with intricate details. The edge generator is responsible for obtaining edges of the missing region, which can be either regular or irregular. The image completion network then uses this information to fill in the missing regions. The authors proposed a deep learning-based image inpainting method that can effectively reconstruct reasonable structures for missing regions. This method performs well even when dealing with images that have multiple or irregularly shaped missing regions. It can be used for tasks such as removing unwanted objects from images or as an interactive image editing tool, producing satisfactory results in terms of both quantitative and qualitative evaluations. However, a current limitation is that the edge generator may struggle to accurately depict edges in cases where a large portion of the image is missing or in highly textured regions.

Zeng et al. [217] introduced a deep generative model called Pyramid-context ENcoder Network (PEN-Net) for image inpainting. The goal of PEN-Net is to fill in missing regions of an image with plausible content. The authors proposed a pyramid-context encoder that utilizes a high-level semantic feature map as a guide. This allows the network to progressively learn the region affinity by transferring attention from the high-level features to the low-level features. By applying this method, damaged images can be restored with visually and semantically plausible results. The key idea of PEN-Net is to encode contextual semantics learned from the full resolution input and decode the semantic features to reconstruct the image. This process ensures both visual and semantic coherence of the generated content by transferring attention from deep to shallow features in a pyramid manner. Additionally, the authors introduced a novel loss function to improve training

convergence and generate more realistic results. Compared to previous methods, PEN-Net outperforms in generating images that are both semantically reasonable and visually realistic.

A multi-scale neural patch synthesis method was introduced by Yang et al. [210], which is based on deep learning. This method incorporates image content and texture constraints to optimize the image inpainting task. It not only restores images with semantically plausible contents but also preserves high-frequency details. Yeh et al. [212] proposed a novel approach for semantic image inpainting, which achieves pixel-level photorealism and generates satisfactory results. Finally, Li et al. [113] proposed an effective method for face completion, which is based on a deep generative model. This method is capable of restoring images with a large area of missing pixels and achieves realistic face completion results.

7.3 Methodology

7.3.1 Skeletal Data and Occlusion

As in the approaches that have been presented in previous Chapters, here we also work using 3D trajectories of human skeletons that have been captured using the Kinect v1/v2 camera, comprising 20 or 25 joints, respectively. In Fig. 2.12 we illustrate a skeleton extracted using Kinect v1 and v2 with joints being grouped to form body parts. As in the two previous Chapters, we herein manually discard subsets of joints that correspond to body parts, assuming that these parts remain occluded during the whole duration of each activity (see Fig. 2.12). Moreover and for the sake of explanation, a visual example of an activity upon occlusion is illustrated in Fig. 6.1, where the loss of visual information is easily comprehensible.

7.3.2 Occlusion

7.3.3 Generative Adversarial Networks

A generative adversarial network (GAN) [9] is a framework for machine learning, mainly used in generative AI [82]. In brief, it consists of two neural networks that take part in a game. Within this game, the gain of one network leads to a loss of the other. The one network is the generator, while the other is the discriminator. The GAN aims to learn how to generate “new” examples which follow the same statistics as those that have been used for training. For example, in our case, as we shall later see, we will use a GAN trained on skeleton sequences; the goal of its generator will be to generate new skeleton sequences that will be “realistic” compared to the real ones that have been used for training. This is achieved based on training through feedback by the discriminator, which is able to discriminate between “realistic” and “non-realistic” images. Both generator and discriminator are dynamically updated, i.e., they are trained in a way that the former learns to

“fool” the latter. Specifically, a given sample that is generated by the generator should be mapped to a specific distribution; each of them should be recognized as belonging to the true distribution or not by the discriminator.

7.3.3.1 Generator

We formulate the generator as a regression task, i.e., the one presented in Chapter 6. Specifically, let $\mathbf{X}$ denote the original skeleton sequence and $\mathbf{X}_o$ the sequence resulting from occlusion. The aim of regression, i.e., of the generator is to estimate a set of parameters β of a given function f , so that $\mathbf{X} = f(\mathbf{X}_o, \beta) + \epsilon$, where ϵ is some error value, to be minimized. As in Chapter 6 we employ the same Convolutional Recurrent Neural Network (CRNN) model as the generator model for our GAN framework to achieve our objective. Therefore, the purpose of this model is to execute f and learn β (i.e., its weights) in order to reduce ϵ . When given an occluded skeleton sequence $\mathbf{X}_o$, the network generates a reconstructed skeleton sequence $\mathbf{X}_r$, which is an approximation of $\mathbf{X}$, by filling in the missing (occluded) data (joints). Also in this case, the CRNN includes an LSTM whose goal is to capture temporal information of skeletal data. Figures 7.5 and 7.4 illustrate the architectures of the CRNN and the LSTM networks, respectively and for the three-camera case for the action gan network 7.1.

7.3.3.2 Discriminator

The Discriminator takes a generated image and a real image and reshapes them according to the timeframes that have been set. Its architecture is quite simple: it first feeds the aforementioned images into an LSTM layer, followed by two dense layers, the first with 256 neurons and the last with 1 neuron that produces a binary result to determine whether the image is real or not. The discriminator loss is a sigmoid cross-entropy loss of the real and generated images. The architecture of the Discriminator is illustrated in Fig. 7.4.

7.3.4 Classification

An occluded sample (i.e., with missing skeletal joints due to occlusion) may be used as input into the trained CRNN network, which in turn reconstructs the missing skeletal data. For the sake of explanation, a visual example of an activity prior to and upon reconstruction is illustrated in Fig. 7.10. Upon its reconstruction, we are then able to proceed with the classification of the aforementioned into one of the pre-defined classes. For classification we use an LSTM network with 1 layer. Note that the CRNN network is trained using only the non-occluded data $\mathbf{X}$, thus without with any occlusion information. On the contrary, during the testing phase, the input of the LSTM network are the corresponding skeletal sequences $\mathbf{X}_r$, which have been reconstructed by $\mathbf{X}$. A visual overview of the proposed approach is illustrated in Fig. 6.4.

To assess the performance of the herein proposed approach, we use a Classifier to measure the accuracy of the model. The architecture of the Classifier is illustrated in Fig. 7.3. Specifically, when three cameras are used, the input from each camera is combined into one, reshaped to fit the predetermined timeframes, and then fed into an LSTM layer. Following this, two dense layers are used, the first with 256 neurons and the last with 1 neuron that produces a result that indicates the user's activity (see Figure 7.3). For datasets with one camera, the same network is used, but with only one input 7.2.

7.3.5 The GAN objective

In this work we use the objective of the Pix2Pix GAN framework [79]. At the following we provide the notion behind the choice of this objective, from the authors of Pix2Pix.

Firstly, the objective L_{cGAN} of a conditional Generative Adversarial Network (GAN) is to create data that meet a specific requirement. This may be described by the following:

$$L_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z)))] \quad (7.1)$$

wherein the Generator G attempts to reduce this objective in opposition to an adversarial Discriminator D that attempts to maximize it, i.e. $G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D)$. To evaluate the significance of conditioning the Discriminator, the authors of [79] also compare it to an unconditional variant in which the discriminator does not observe x :

$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z)))] \quad (7.2)$$

Previous approaches have suggested to combine the GAN objective with a more traditional loss, such as the L2 distance. The Discriminator's role remains the same, but the Generator is now responsible for not only deceiving the Discriminator but also being close to the real output in terms of L2 distance. Thus, they also investigate this option, using L1 distance instead of L2:

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1] \quad (7.3)$$

The final objective is:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \quad (7.4)$$

Note that without z , the net would still be able to learn a mapping from x to y , but the outputs would be deterministic and thus unable to match any distribution other than a delta function. Previous conditional GANs have addressed this by providing Gaussian noise z as an input to the Generator, in addition to x (e.g., [193]). However, it has been experimentally showed that this strategy was not effective as the Generator simply learned to ignore the noise as it has been stated by Mathieu et al. [132].

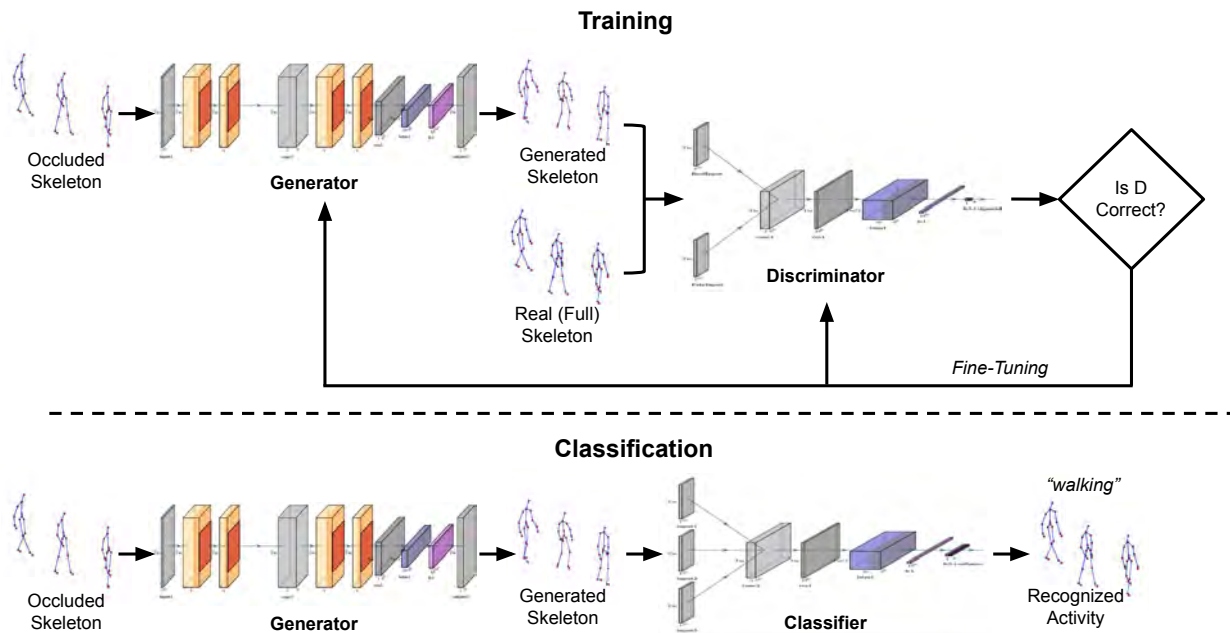


Figure 7.1: A visual overview of the proposed approach.

7.3.6 Network Training and Evaluation

As in previous Chapters, the herein presented approach takes temporal sequences of 3D skeleton data as input, (see subsection 2.8). Interpolation is also applied to set the length of all activity examples to T_m , which is the size of the longest example, following the procedure described in subsection 3.3. We once again use single- and multi-view datasets. In the latter case, we exploit all available information by feeding the network with all 3 skeleton sequences. As in the approach presented in Chapter 6, we opted to train one network per occlusion case, resulting in 8 different networks. Thus, given an input skeletal sequence, missing joints are identified and then fed to the appropriate trained network to be classified into an activity class. Although this approach requires more memory, its significant increase in performance compared to the use of a single network compensates for this requirement. Note that the classifier is trained exclusively using non-occluded samples.

For training of the generator, the LeakyReLU activation function was used, except for the LSTM layer where the tanh function was used, and the last dense layer where the linear activation function was used. For the training of the classifier, the LeakyReLU and tanh activation functions were used respectively, except for the last layer, where the sigmoid activation function was used. The batch size was set to 5 and 10 for the training of the classifier and the gan, respectively. The Adam optimizer was utilized in both cases, the dropout was set to 0.25, the learning rate to 0.001 and the training was done for 200 epochs, using the loss of the validation set calculated via MSE as an early stopping method, to prevent overfitting. In all cases, 80% of the available data was used for training, 10% for validation and the remaining 10% for testing. The importance of these parameters should

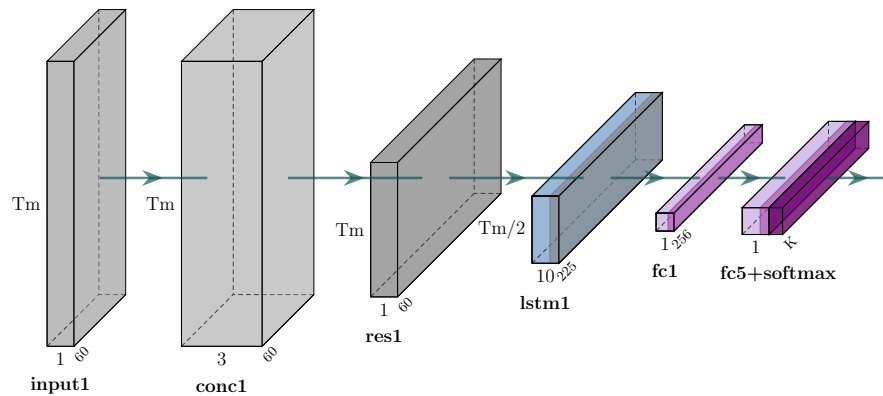


Figure 7.2: The architecture of the Classifier of the proposed GAN architecture., for the one-camera case.

not be overstated, as GANs are extremely sensitive to any changes. In our case, the aforementioned parameters have produced the most favorable results upon a trial-and-error experimental process.

At this stage, a sample with missing skeletal joints due to occlusion can be used as input into the trained Generator, which then reconstructs the missing skeletal data. An example of this activity before and after reconstruction is shown in Figure 7.10. After reconstruction, the data can be classified into one of the pre-defined classes using the Classifier. During the testing phase, the input of the Classifier is a reconstructed skeletal sequence $\mathbf{X}_r$. A visual representation of the proposed approach is presented in Figure 7.10.

7.3.7 Evaluation Protocol and Results

To experimentally evaluate the proposed methodology, we considered the following:

- removal of structured sets of skeletal joints, corresponding to body parts, to simulate occlusion (see Fig. 2.12). Specifically, as in previous Chapters, we removed (a) left arm; (b) right arm; (c) both arms; (d) left leg; (e) right leg; (f) both legs; (g) left arm & left leg; (h) right arm & right leg. We used an LSTM network that had been trained using non-occluded samples, while reconstruction took place using the GAN.
- a baseline approach, wherein the LSTM was trained and evaluated using non-occluded samples,
- a “reference” approach wherein the LSTM was trained using non-occluded samples and validated using occluded samples, and
- inclusion of occluded samples in the training process of the LSTM and validation using occluded samples. In this case, a subset equal to 10% of the non-occluded samples of training set was selected. From those samples, we created all 8 cases

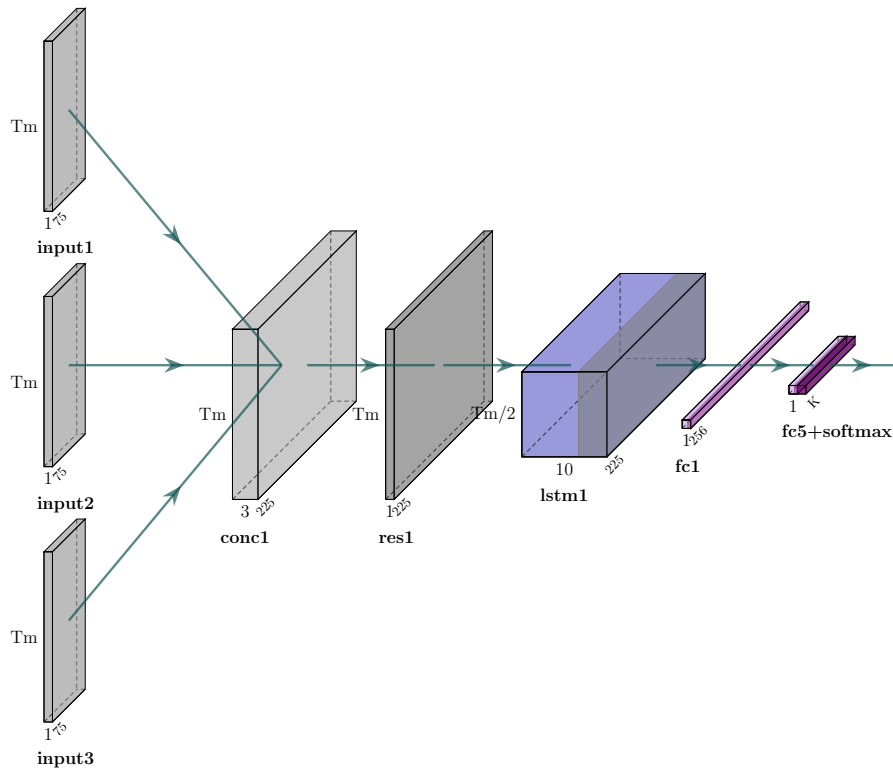


Figure 7.3: The architecture of the Classifier of the proposed GAN architecture., for the three-camera case.

of occlusion, thus “augmenting” the initial training set by 80%. We used a single network for all 8 cases of occlusion.

The experimental results for all datasets are depicted in Tables 7.1 – 7.3. We extracted the following metrics: accuracy per class, F_1 score per class and weighted accuracy, where class weights were calculated based on the class distribution. Moreover, confusion matrices for all datasets in case of the baseline experiment are depicted in Fig. 6.5, while in case of classification of reconstructed samples are depicted in Figs. 7.8 – 7.7.

In case of the PKU-MMD dataset, the weighted accuracy (WA) was 0.92 without any body part removal. Specifically, it ranged between 0.21–0.90 in case of some body part removal, while it ranged between 0.78–0.93 upon reconstruction with the GAN. In all cases, significant improvement was observed, in terms of WA. Moreover, reconstruction with the GAN outperformed the regression approach (see Chapter 6) in all cases, and the augmentation approach (see Chapter 5) in 6 out of 8 cases. Since as we have stated in subsection 2.8 the majority of the activities we used to evaluate our approach mainly consists of upper body motion (i.e., left and/or right arm), this is also reflected to the results of Table 7.1, wherein it may be observed that in cases of occluded arms the improvement is significantly large, with most notable example the case of both arms, wherein WA improves from 0.21 to 0.78. Upon careful observation of the confusion matrices depicted in Fig. 7.8, for each

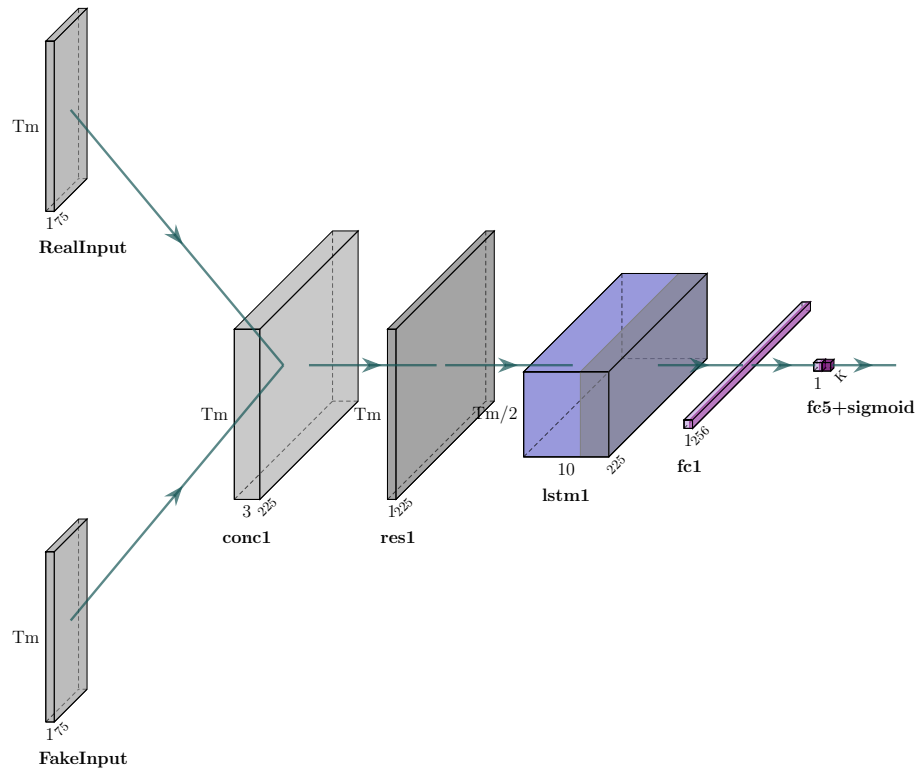


Figure 7.4: The architecture of the Discriminator of the proposed GAN architecture.

occlusion case we should notice the following, when comparing with the case where all joints had been used:

- in case of any occluded arm, class *make a phone call/answer phone* is often confused with *playing with phone/tablet*. In case of occluded right arm, class *eat meal/snack* is very often confused with *reading*. This happens less often in case of occluded left arm. Also in a few cases class *handshaking* is confused with *hugging* and class *reading* with *typing on a keyboard*;
- in case of any occluded leg, class *make a phone call/answer phone* is often confused with *playing with phone/tablet*. Also, in case of occluded left leg class *reading* is often confused with *eat meal/snack*, while in case of occluded right leg class *eat meal/snack* is very often confused with *reading*.
- in case of occluded left arm and left leg, class *make a phone call/answer phone* in the majority of testing examples is confused with *playing with phone/tablet* and class *hugging* with *playing with phone/tablet* or *handshaking*.
- in case of occluded right arm and right leg, class *make a phone call/answer phone* is often confused with *playing with phone/tablet* or *handshaking*, *eat meal/snack* is very often confused with *reading* and class *handshaking* is often confused with *make a phone call/answer phone*.

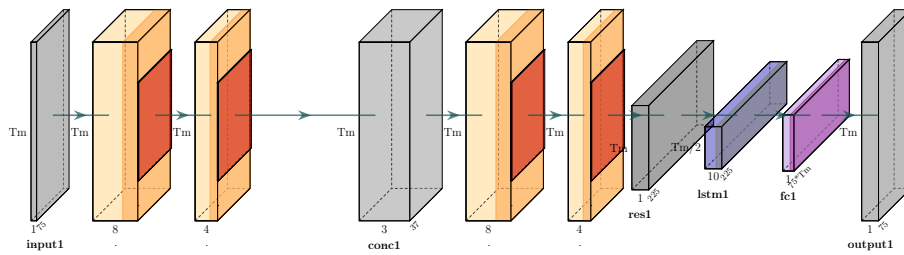


Figure 7.5: The architecture of the Generator of the proposed GAN architecture.

In all cases, the majority of classes demonstrate excellent performance, equivalent to the case without any occlusion. Compared to the augmentation approach that has been presented in Chapter 6, we should note that reconstruction using GANs was superior in 6 out of the 8 cases of occlusion. Specifically, augmentation showed superior performance in case of both occluded arms and in case of occluded left arm and left leg, although in that case reconstruction using GANs exhibited equivalent performance.

In case of NTU-RGB+D dataset, the WA was 0.68 without any body part removal and ranged between 0.16–0.59 in case of some body part removal, while it ranged between 0.35–0.71 upon reconstruction with the GAN. In that case, in 5 out of 8 cases, significant improvement was observed, in terms of WA, while performance was almost equal with the other two approaches in case of removal of Right Arm or Right Arm and Right Leg. In the remaining 3 cases the regression approach demonstrated best performance. Moreover, reconstruction with GAN outperformed the augmentation approach in 7 out of 8 cases. Since activities used to evaluate the proposed approach mainly consisted of upper body motion, also in the case of the NTU-RGB+D dataset, in the results of Table 6.2 it could be observed that in all the remaining cases of occluded arms, the improvement of WA is large, with the most notable example the case of both arms, wherein WA improves from 0.16 to 0.35, although the other two approaches demonstrated superior performance in that case. Upon careful observation of the confusion matrices depicted in Fig. 7.6 and considering the classification results without occlusion of any part, for each occlusion case we should notice that most activities are affected by the occlusion of the Right Arm and performance is far from being excellent in the majority of these cases. Notably reconstruction with GANs exhibits best performance in case of occluded legs.

In case of SYSU-3D-HOI dataset, the WA was 0.54 without any body part removal and ranged between 0.10–0.22 in case of some body part removal, while it ranged between 0.44–0.52 upon reconstruction with GAN. Moreover, reconstruction with GANs outperformed the other two methods in 7 out of 8 cases. In every case of occlusion, significant improvement was observed, in terms of WA, when compared to the reference case. We should herein notice that due to the small size of this dataset, performance was inadequate in several classes even without occlusion, while reconstruction in several cases exhibited superior performance. It was quite a surprise to observe that reconstruction with GANs steadily exhibited almost constant performance for all occlusion cases, whilst in many occasions it was also superior to the baseline approach, i.e., to the absence of occlusion.

Human Activity Recognition Under Occlusion

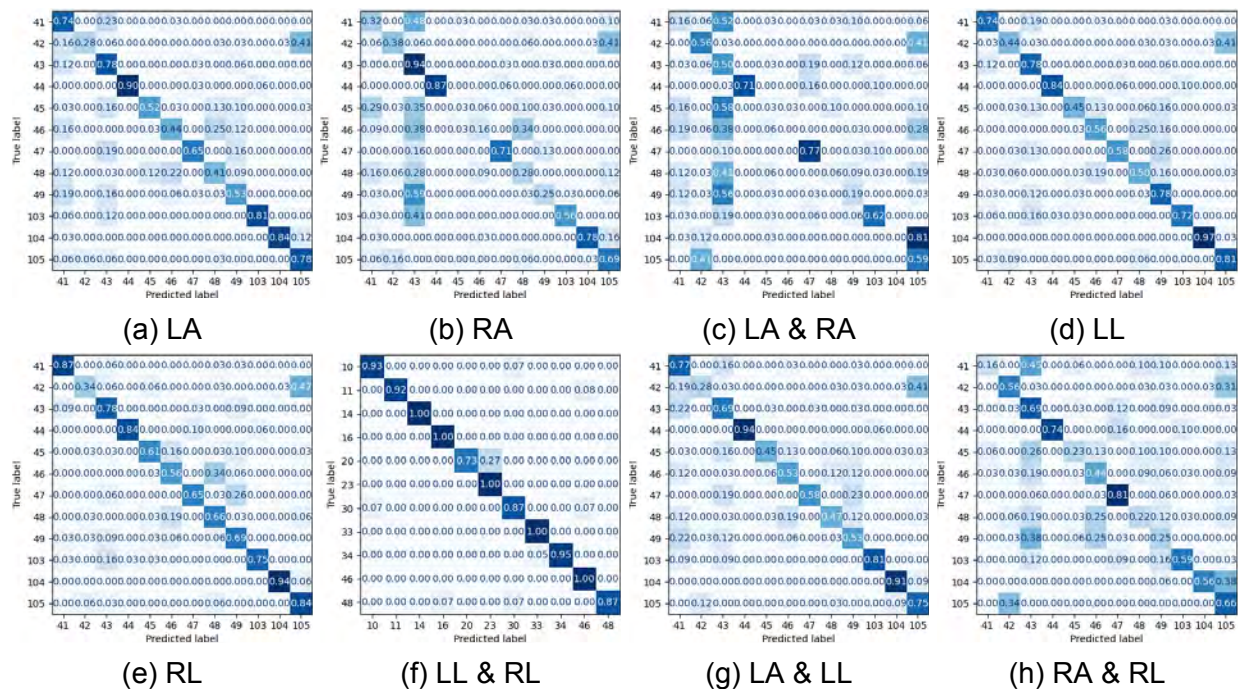


Figure 7.6: Normalized confusion matrices for classification for the NTU-RGB+D dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

Also, due to the small size of the dataset, often several classes fail to be recognized.

In case of the UTKinect-Action-3D dataset, the WA was 0.79 without and body part removal, and ranged between 0.09-0.61 with some body part removal and also between 0.55–0.80 upon reconstruction with GAN. Moreover, reconstruction with GAN outperformed both other approaches in 7 out of 8 cases. Since activities used to evaluate the proposed approach mainly consisted of upper body motion, also in the case of the UTKinect-Action-3D dataset, in the results of Table 6.4 it could be observed that in all the remaining cases of occluded arms, the improvement of WA is exceptional, with the most notable examples the cases of occluded Right Arm and Right Leg, wherein WA improves from 0.11 to 0.70 and Right Arm and Left Argm wherein WA improves from 0.09 to 0.55. Upon careful observation of the confusion matrices depicted in Fig. 7.7 and considering the classification results without occlusion of any part, for each occlusion case we should notice that occlusion affects half of the activities, especially those that are based on the motion of arms

In all cases, occlusion of some body parts lead to a severe drop of performance; in many occasions, accuracy and F_1 score of some classes dropped to zero/near zero values. In some of these cases, reconstruction was successful, leading to near perfect recognition. Finally, also in this case, in several activities and for specific cases of occlusion, it is observed that classification of the reconstructed samples exhibits slightly inferior performance compared to the occluded samples. It is our belief that in those cases, the specific

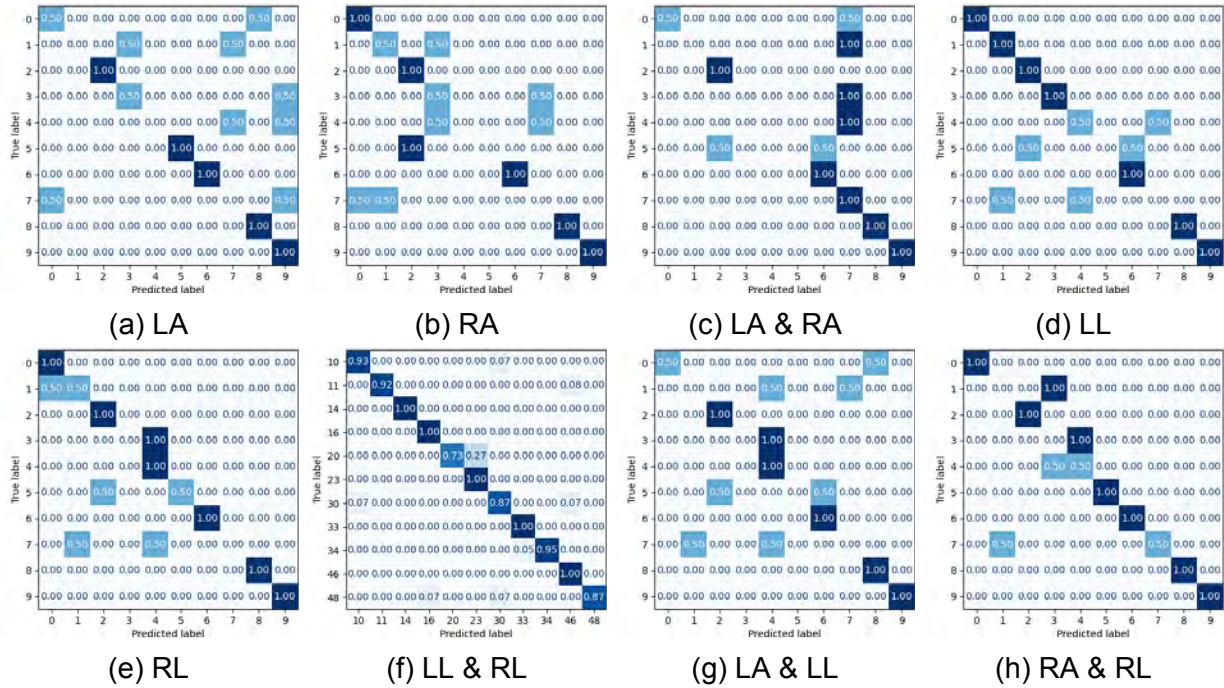


Figure 7.7: Normalized confusion matrices for classification for the UTKinect-Action3D dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

body part which is occluded is less “relevant” for those activities, although in some cases the reconstruction approach may fail and lead to misleading joint positions, as in the example of Fig.6.1b.

7.4 Conclusions

In this Chapter, we have further developed the method for reconstructing human skeletons, following the same assumptions as in previous methodologies. Specifically, we have addressed this problem by employing a generative adversarial network. The generator in this network is a convolutional recurrent neural network, which has been trained to generate the complete skeleton by filling in the missing parts. The input to the generator consists of the raw 3D positions of the skeleton joints, with the occluded joints removed. The output of the generator is a reconstructed skeleton, which is then passed through a long short-term memory network for classification into one of the predefined activities. For the discriminator, we have used a simple long short-term memory network trained only on non-occluded samples. Through experimental evaluation, we have demonstrated that this approach offers improved performance compared to our previous methods.

Human Activity Recognition Under Occlusion

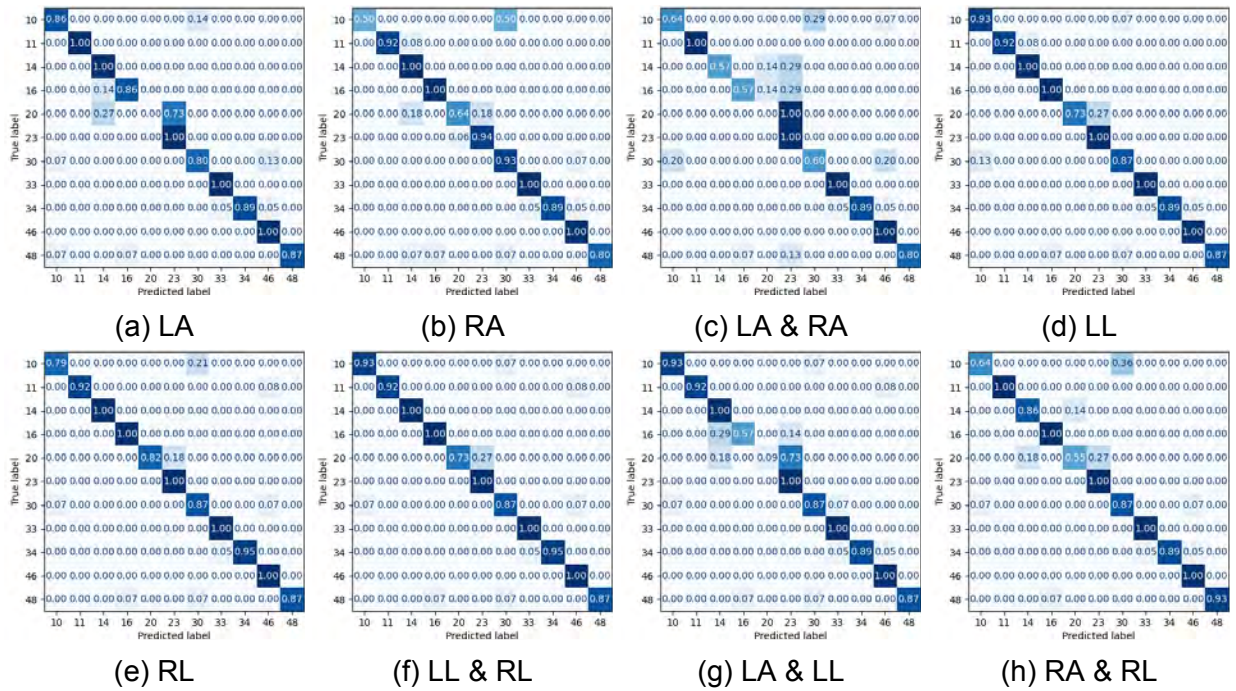


Figure 7.8: Normalized confusion matrices for classification for the PKU-MMD dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

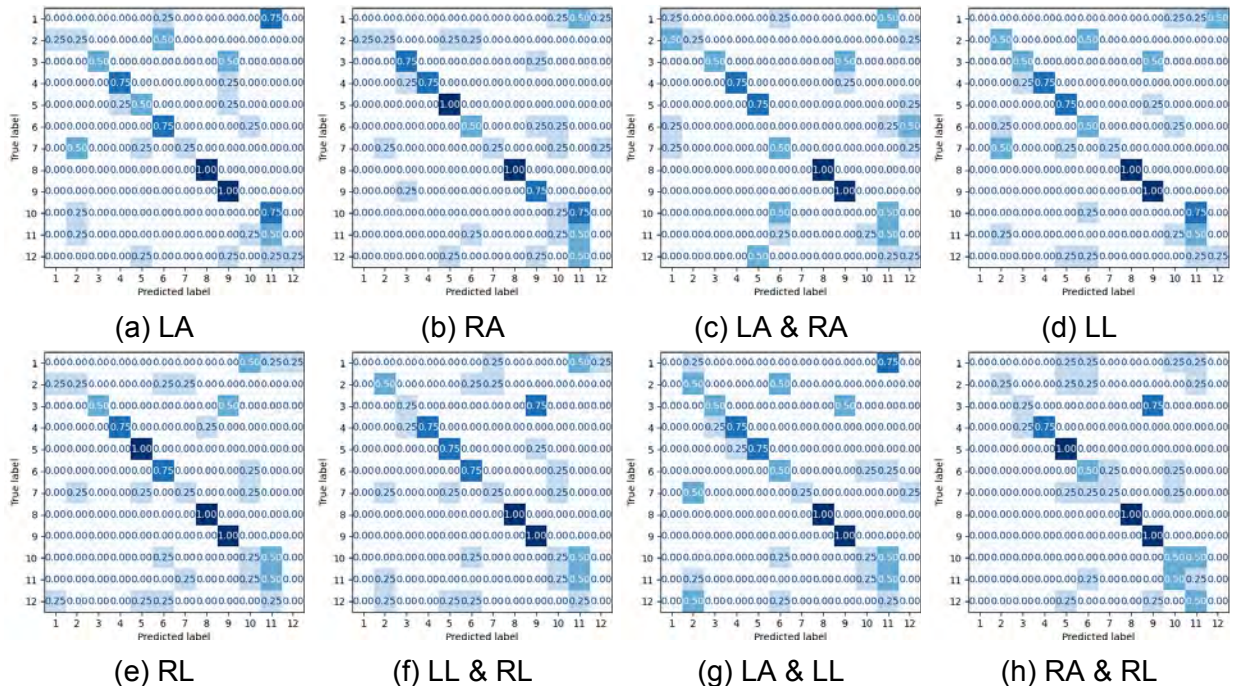


Figure 7.9: Normalized confusion matrices for classification for the SYSU-3D-HOI dataset denoted in the caption of the corresponding subfigure. LA, RA, LL and RL correspond to cases of occluded Left Arm, Right Arm, Left Leg and Right Leg, respectively.

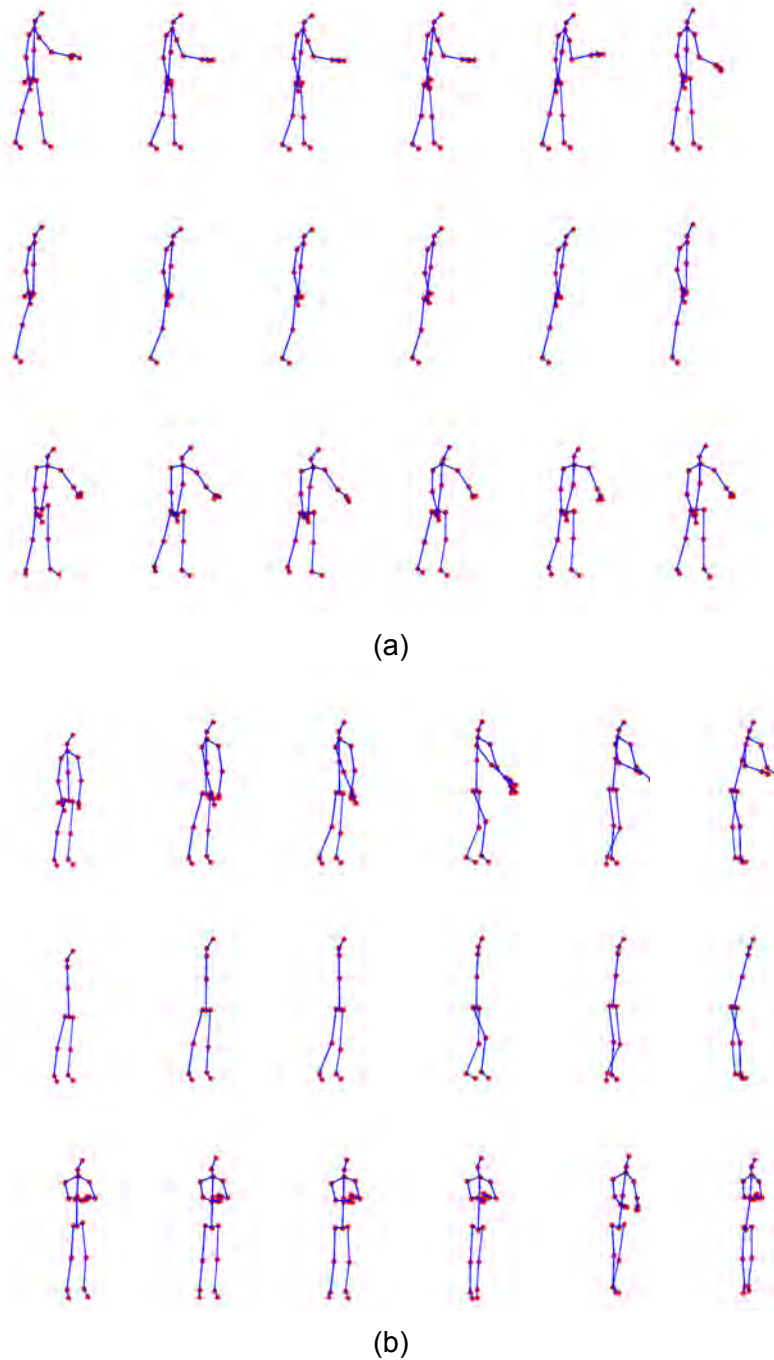


Figure 7.10: Example skeleton sequences of the activities (a) *handshaking* and (b) *hugging other person* from the PKU-MMD dataset, captured by Microsoft Kinect v2. First row: original skeletons, including all 25 joints (i.e., without any occlusion); Second row: joints corresponding to (a) left arm; (b) both arms (see Fig. 2.12) have been discarded (i.e., the skeleton is partially occluded); Third row: skeletons have been reconstructed using the proposed deep regression approach. The example of (a) is successfully reconstructed and correctly classified, while the example of (b) is unsuccessfully reconstructed and incorrectly classified.

Table 7.1: Results on PKU-MMD dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

Class	Metric	None			LA			RA			LA+RA			LL			RL			LL+RL			LA+LL			RA+RL								
		Bas.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.								
10	Acc.	0.90	0.86	0.84	0.43	0.11	0.50	0.45	0.36	0.03	0.64	0.58	0.64	0.00	0.93	0.79	0.71	0.92	0.79	0.79	0.50	0.66	0.93	0.87	0.71	0.74	0.93	0.68	0.79	0.21	0.64	0.66	0.71	0.24
	F_1	0.86	0.86	0.75	0.60	0.15	0.67	0.57	0.32	0.05	0.69	0.55	0.78	0.00	0.90	0.86	0.83	0.84	0.85	0.81	0.67	0.72	0.93	0.82	0.74	0.78	0.93	0.70	0.88	0.25	0.75	0.69	0.77	0.36
	Acc.	0.97	1.00	0.97	1.00	0.97	0.92	0.97	0.85	0.97	1.00	0.97	1.00	0.00	0.92	0.97	0.92	0.97	0.92	0.95	1.00	0.97	0.92	0.97	0.85	0.97	1.00	1.00	1.00	0.97	1.00	0.97	1.00	0.97
11	F_1	0.99	1.00	0.99	0.96	0.90	0.96	0.97	0.88	0.99	1.00	0.95	1.00	0.00	0.96	0.91	0.92	0.99	0.96	0.97	1.00	0.99	0.96	0.99	0.88	0.99	0.96	0.97	0.96	0.70	1.00	0.99	1.00	0.92
	Acc.	1.00	1.00	0.88	1.00	1.00	1.00	0.97	1.00	0.29	1.00	0.57	1.00	1.00	0.81	1.00	1.00	0.94	1.00	1.00	0.86	1.00	1.00	1.00	0.57	1.00	1.00	1.00	1.00	0.94	0.86	1.00	0.29	1.00
	F_1	1.00	0.78	0.93	0.58	0.97	0.78	1.00	0.27	1.00	0.73	0.97	0.67	0.90	0.93	0.97	0.64	0.94	1.00	1.00	0.71	0.91	1.00	1.00	0.62	0.91	0.78	1.00	0.78	0.91	0.80	0.97	0.33	0.86
16	Acc.	0.88	0.86	0.94	0.14	0.88	0.57	0.88	0.43	0.69	1.00	0.00	0.73	0.85	1.00	0.29	0.88	1.00	0.57	0.88	1.00	0.94	0.57	0.88	0.57	0.81	1.00	1.00	0.94	0.43	0.88	0.88		
	F_1	0.93	0.86	0.91	0.25	0.90	0.93	0.94	0.25	0.93	0.67	0.90	0.50	0.76	0.93	0.97	0.44	0.93	0.97	0.73	0.90	0.93	0.97	0.67	0.90	0.67	0.97	0.62	0.79	0.93	1.00	0.50	0.85	
	Acc.	0.82	0.00	0.18	0.91	0.03	0.64	0.61	0.73	0.39	0.00	0.00	0.73	0.00	0.73	0.85	1.00	0.79	0.82	0.82	0.73	0.88	0.73	0.79	0.55	0.88	0.00	0.57	0.39	0.64	0.97			
20	F_1	0.89	0.00	0.31	0.57	0.06	0.74	0.71	0.47	0.19	0.00	0.00	0.59	0.00	0.84	0.90	0.61	0.87	0.90	0.86	0.67	0.62	0.84	0.87	0.55	0.90	0.17	0.45	0.71	0.00	0.67	0.57	0.45	0.27
	Acc.	0.95	1.00	0.93	0.24	0.98	0.94	0.95	0.35	0.83	1.00	0.98	0.59	0.05	1.00	0.95	0.18	0.93	1.00	0.95	0.82	0.02	1.00	0.93	0.94	0.05	1.00	0.91	0.59	1.00	0.98	0.53	0.05	
	F_1	0.90	0.81	0.74	0.38	0.70	0.91	0.83	0.44	0.40	0.67	0.56	0.69	0.05	0.92	0.92	0.30	0.93	0.94	0.91	0.85	0.05	0.92	0.91	0.84	0.09	0.79	0.75	0.71	0.62	0.92	0.80	0.56	0.07
30	Acc.	0.84	0.80	0.60	0.86	0.70	0.93	0.87	0.21	0.24	0.60	0.16	0.93	0.00	0.87	0.97	0.86	0.76	0.87	0.87	0.93	0.89	0.87	0.89	0.79	0.84	0.87	0.81	0.93	0.30	0.87	0.76	0.86	0.70
	F_1	0.84	0.83	0.68	0.71	0.48	0.76	0.67	0.32	0.36	0.64	0.22	0.79	0.00	0.87	0.77	0.80	0.78	0.81	0.78	0.74	0.75	0.87	0.88	0.73	0.75	0.87	0.73	0.84	0.27	0.79	0.68	0.77	0.68
	Acc.	0.98	1.00	0.96	1.00	0.74	1.00	0.94	0.95	0.06	1.00	0.91	1.00	0.00	1.00	0.85	0.95	0.98	1.00	0.94	1.00	0.98	1.00	0.98	1.00	0.98	1.00	0.94	0.95	1.00	0.96	1.00	0.00	0.00
33	F_1	0.98	0.97	0.96	1.00	0.84	0.97	0.96	0.92	0.11	0.97	0.94	1.00	0.00	0.97	0.91	0.95	0.98	0.97	0.95	1.00	0.98	0.97	0.98	0.97	0.98	0.95	0.96	0.97	0.20	0.97	0.97	1.00	0.00
	Acc.	0.96	0.90	0.89	0.95	0.96	0.90	0.96	0.95	0.19	0.90	0.46	1.00	0.00	0.90	1.00	0.95	0.94	0.95	0.94	1.00	0.90	0.95	0.98	1.00	0.85	0.90	0.98	1.00	0.54	0.90	0.96	1.00	0.00
	F_1	0.96	0.94	0.91	0.97	0.96	0.94	0.94	0.97	0.32	0.94	0.89	1.00	0.00	0.94	0.93	0.97	0.95	0.97	0.94	1.00	0.93	0.97	0.92	1.00	0.91	0.94	0.96	1.00	0.70	0.94	0.95	1.00	0.00
46	Acc.	0.87	1.00	0.84	1.00	0.89	1.00	0.87	1.00	0.49	1.00	0.84	1.00	0.65	1.00	0.87	1.00	0.84	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.87	1.00	0.84	1.00	0.87	1.00	0.87	1.00	0.65
	F_1	0.91	0.90	0.89	0.97	0.69	0.93	0.88	0.79	0.38	0.85	0.90	1.00	0.14	0.97	0.91	0.94	0.90	0.93	0.90	0.97	0.93	0.93	0.91	0.97	0.91	0.93	0.87	0.97	0.48	0.93	0.89	1.00	0.56
	Acc.	0.92	0.87	0.92	0.80	0.28	0.80	0.69	0.73	0.13	0.80	0.56	0.67	0.00	0.87	0.59	0.80	0.85	0.87	0.85	0.87	0.85	0.87	0.80	0.80	0.87	0.90	0.73	0.05	0.93	0.87	0.73	0.13	
48	F_1	0.88	0.93	0.78	0.89	0.43	0.89	0.76	0.76	0.22	0.89	0.58	0.80	0.00	0.93	0.74	0.86	0.87	0.93	0.87	0.93	0.72	0.93	0.80	0.89	0.74	0.93	0.86	0.85	0.10	0.97	0.88	0.85	0.23
	WA	0.92	0.86	0.82	0.78	0.68	0.87	0.84	0.65	0.40	0.78	0.70	0.83	0.21	0.93	0.89	0.80	0.90	0.93	0.90	0.87	0.80	0.93	0.91	0.84	0.80	0.86	0.84	0.87	0.48	0.89	0.86	0.80	0.41
	all	WA	0.92	0.86	0.82	0.78	0.68	0.87	0.84	0.65	0.40	0.78	0.70	0.83	0.21	0.93	0.89	0.80	0.90	0.93	0.90	0.87	0.80	0.93	0.91	0.84	0.80	0.86	0.84	0.87	0.48	0.89	0.86	0.80

Table 7.2: Results on NTU-RGB+D dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

Class	Metric	None			LA			RA			LA+RA			LL			RL			LL+RL			LA+LL			RA+RL								
		Bas.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.	Ref.	Gains	Reg.	Occ.					
41	Acc.	0.89	0.74	0.70	0.87	0.73	0.32	0.51	0.74	0.14	0.16	0.57	0.84	0.11	0.74	0.73	0.84	0.24	0.87	0.65	0.81	0.62	0.90	0.87	0.74	0.24	0.77	0.70	0.84	0.16	0.16	0.73	0.71	0.11
F ₁	0.82	0.55	0.66	0.73	0.74	0.31	0.54	0.65	0.19	0.17	0.58	0.79	0.11	0.72	0.70	0.71	0.35	0.86	0.66	0.74	0.59	0.86	0.78	0.73	0.38	0.58	0.68	0.80	0.24	0.26	0.73	0.73	0.19	
42	Acc.	0.35	0.28	0.22	0.47	0.13	0.38	0.17	0.34	0.26	0.56	0.26	0.41	0.00	0.44	0.22	0.41	0.09	0.34	0.26	0.44	0.09	0.31	0.30	0.44	0.00	0.28	0.13	0.41	0.00	0.56	0.22	0.47	0.09
F ₁	0.40	0.42	0.29	0.42	0.17	0.46	0.20	0.33	0.31	0.48	0.32	0.39	0.00	0.53	0.28	0.40	0.14	0.46	0.32	0.40	0.15	0.41	0.36	0.41	0.00	0.39	0.19	0.39	0.00	0.55	0.28	0.45	0.14	
43	Acc.	0.66	0.78	0.62	0.59	0.72	0.94	0.66	0.53	0.83	0.50	0.76	0.69	0.62	0.78	0.69	0.47	0.52	0.78	0.76	0.69	0.76	0.78	0.76	0.69	0.72	0.69	0.62	0.69	0.62	0.69	0.55		
F ₁	0.61	0.56	0.46	0.50	0.46	0.41	0.47	0.42	0.35	0.24	0.49	0.51	0.29	0.62	0.53	0.41	0.38	0.68	0.54	0.46	0.27	0.71	0.57	0.50	0.36	0.55	0.51	0.59	0.31	0.41	0.56	0.51	0.26	
44	Acc.	0.93	0.90	0.87	0.87	0.77	0.87	0.90	0.87	0.07	0.71	0.90	0.87	0.07	0.84	0.90	0.87	0.93	0.84	0.90	0.90	0.97	0.94	0.90	0.84	1.00	0.94	0.90	0.87	0.97	0.74	0.90	0.84	0.37
F ₁	0.95	0.95	0.90	0.89	0.87	0.93	0.93	0.89	0.12	0.83	0.95	0.89	0.12	0.90	0.93	0.87	0.93	0.90	0.95	0.90	0.95	0.95	0.93	0.88	0.85	0.97	0.93	0.87	0.89	0.85	0.93	0.88	0.54	
45	Acc.	0.41	0.52	0.18	0.28	0.46	0.03	0.23	0.19	0.91	0.32	0.23	0.28	0.41	0.45	0.23	0.25	0.14	0.61	0.23	0.34	0.59	0.52	0.50	0.22	0.50	0.45	0.18	0.41	0.14	0.23	0.41	0.16	0.77
F ₁	0.51	0.62	0.22	0.27	0.40	0.06	0.24	0.23	0.26	0.05	0.23	0.25	0.23	0.57	0.26	0.25	0.17	0.68	0.26	0.29	0.32	0.63	0.47	0.26	0.23	0.57	0.21	0.38	0.19	0.31	0.40	0.19	0.23	
46	Acc.	0.38	0.44	0.32	0.47	0.24	0.16	0.27	0.47	0.12	0.00	0.35	0.56	0.27	0.56	0.50	0.53	0.32	0.56	0.29	0.44	0.15	0.50	0.29	0.50	0.09	0.53	0.35	0.59	0.12	0.44	0.35	0.66	0.03
F ₁	0.43	0.49	0.39	0.35	0.31	0.23	0.30	0.33	0.17	0.00	0.39	0.41	0.31	0.59	0.47	0.36	0.37	0.57	0.33	0.38	0.21	0.50	0.35	0.36	0.14	0.55	0.37	0.41	0.20	0.42	0.40	0.42	0.05	
47	Acc.	0.53	0.65	0.40	0.42	0.63	0.71	0.27	0.42	0.27	0.77	0.27	0.58	0.07	0.58	0.33	0.45	0.33	0.65	0.27	0.52	0.37	0.65	0.33	0.65	0.13	0.58	0.40	0.71	0.33	0.81	0.33	0.65	0.43
F ₁	0.64	0.74	0.53	0.55	0.73	0.79	0.39	0.55	0.28	0.69	0.38	0.65	0.11	0.69	0.45	0.56	0.43	0.73	0.39	0.58	0.48	0.77	0.45	0.63	0.21	0.72	0.51	0.70	0.43	0.72	0.45	0.63	0.36	
48	Acc.	0.79	0.41	0.38	0.10	0.42	0.28	0.38	0.07	0.17	0.62	0.38	0.03	0.50	0.50	0.54	0.07	0.67	0.66	0.29	0.10	0.21	0.50	0.50	0.10	0.21	0.47	0.38	0.07	0.63	0.22	0.38	0.03	0.21
F ₁	0.55	0.43	0.31	0.15	0.47	0.30	0.32	0.10	0.18	0.11	0.31	0.06	0.14	0.52	0.39	0.11	0.31	0.58	0.24	0.16	0.22	0.46	0.44	0.17	0.14	0.54	0.32	0.12	0.32	0.29	0.32	0.06	0.15	
49	Acc.	0.68	0.53	0.48	0.44	0.48	0.25	0.46	0.38	0.05	0.19	0.36	0.50	0.00	0.78	0.46	0.28	0.77	0.69	0.48	0.56	0.16	0.69	0.48	0.44	0.41	0.53	0.46	0.38	0.59	0.25	0.52	0.41	0.23
F ₁	0.59	0.51	0.42	0.44	0.47	0.34	0.41	0.39	0.07	0.23	0.36	0.42	0.00	0.60	0.44	0.33	0.46	0.61	0.42	0.47	0.24	0.69	0.45	0.38	0.31	0.49	0.43	0.40	0.40	0.25	0.46	0.37	0.26	
103	Acc.	0.79	0.81	0.76	0.72	0.79	0.56	0.82	0.78	0.40	0.63	0.76	0.53	0.16	0.72	0.76	0.75	0.74	0.75	0.84	0.50	0.74	0.84	0.82	0.69	0.61	0.81	0.79	0.72	0.74	0.59	0.79	0.72	0.32
F ₁	0.85	0.87	0.79	0.81	0.79	0.68	0.84	0.83	0.54	0.68	0.79	0.68	0.26	0.79	0.82	0.81	0.84	0.83	0.84	0.67	0.85	0.89	0.83	0.77	0.72	0.87	0.85	0.79	0.82	0.68	0.83	0.81	0.48	
104	Acc.	0.97	0.84	0.94	0.75	0.74	0.78	0.94	0.75	0.21	0.00	0.88	0.69	0.00	0.97	0.88	0.84	0.21	0.94	0.94	0.63	0.15	0.94	0.88	0.78	0.03	0.91	0.94	0.78	0.12	0.56	0.91	0.81	0.32
F ₁	0.96	0.90	0.93	0.81	0.82	0.85	0.90	0.76	0.34	0.00	0.88	0.79	0.00	0.97	0.90	0.84	0.33	0.95	0.93	0.75	0.24	0.95	0.90	0.86	0.06	0.88	0.91	0.83	0.21	0.71	0.89	0.85	0.48	
105	Acc.	0.60	0.78	0.66	0.34	0.77	0.69	0.51	0.38	0.11	0.59	0.63	0.22	0.00	0.81	0.57	0.47	0.14	0.84	0.60	0.16	0.11	0.84	0.60	0.25	0.00	0.75	0.69	0.28	0.20	0.66	0.54	0.28	0.03
F ₁	0.66	0.67	0.64	0.37	0.65	0.52	0.54	0.38	0.17	0.34	0.65	0.27	0.00	0.70	0.61	0.46	0.24	0.68	0.62	0.22	0.19	0.67	0.65	0.31	0.00	0.65	0.69	0.32	0.26	0.46	0.60	0.35	0.05	
all	WA	0.68	0.64	0.57	0.53	0.59	0.50	0.53	0.49	0.27	0.35	0.55	0.52	0.16	0.68	0.59	0.52	0.44	0.71	0.57	0.51	0.41	0.70	0.62	0.53	0.33	0.64	0.58	0.56	0.40	0.49	0.59	0.53	0.25

Table 7.3: Results on UTKinect-Action3D dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

Class	Metric	None			LA			RA			LA+RA			LL			RL			LL+RL			LA+LL			RA+RL								
		Baseline	Gan	Ref.	Occ.	Reg.	Acc.	Reg.	Occ.	Ref.	Occ.	Reg.	Acc.	Reg.	Occ.	Ref.	Occ.	Reg.	Acc.	Reg.	Occ.	Ref.	Occ.	Reg.	Acc.	Reg.	Occ.	Ref.						
0	Acc.	0.50	0.40	0.30	1.00	0.40	0.30	0.00	0.50	0.40	0.30	0.00	1.00	0.40	0.30	0.20	1.00	0.40	0.40	0.50	1.00	0.40	0.30	0.20	0.50	0.40	0.50	0.70	1.00	0.40	0.40	0.00		
	F ₁	0.48	0.50	0.43	0.35	0.31	0.80	0.43	0.35	0.00	0.67	0.39	0.26	0.00	1.00	0.43	0.31	0.09	0.80	0.43	0.41	1.00	0.43	0.28	0.07	0.67	0.43	0.44	0.33	1.00	0.39	0.43	0.00	
1	Acc.	0.70	0.00	0.30	0.20	0.60	0.50	0.10	0.20	0.30	0.00	0.30	0.20	0.90	1.00	0.70	0.60	0.00	0.50	0.50	0.30	0.40	0.50	0.10	0.00	0.40	0.50	0.20	0.00	0.30	0.30	0.20		
	F ₁	0.53	0.00	0.23	0.16	0.15	0.50	0.08	0.10	0.17	0.00	0.13	0.11	0.18	0.80	0.57	0.51	0.00	0.50	0.39	0.26	0.35	0.50	0.43	0.10	0.00	0.32	0.39	0.06	0.00	0.21	0.23	0.13	
2	Acc.	0.80	1.00	0.80	0.70	0.00	1.00	0.80	0.50	0.00	1.00	0.90	0.80	0.00	1.00	0.80	0.80	0.10	1.00	0.70	0.80	0.60	1.00	0.70	0.60	0.00	1.00	0.90	0.90	0.20	1.00	0.90	0.40	0.00
	F ₁	0.87	1.00	0.76	0.66	0.00	0.67	0.80	0.41	0.00	0.80	0.83	0.76	0.00	1.00	0.80	0.67	0.13	0.80	0.75	0.66	0.63	0.80	0.75	0.46	0.00	0.80	0.79	0.85	0.27	1.00	0.89	0.26	0.00
3	Acc.	1.00	0.50	0.30	0.30	0.00	0.50	0.60	0.10	0.20	0.00	0.00	0.30	0.00	1.00	0.70	0.40	0.40	0.00	0.70	0.60	1.00	1.00	0.80	0.30	0.70	0.00	0.20	0.70	0.00	0.20	0.30	0.30	
	F ₁	0.92	0.50	0.22	0.26	0.00	0.40	0.43	0.05	0.05	0.00	0.00	0.33	0.00	1.00	0.58	0.41	0.40	0.00	0.63	0.57	0.69	0.67	0.72	0.18	0.61	0.00	0.11	0.63	0.00	0.16	0.29	0.07	
4	Acc.	0.90	0.00	0.00	0.20	0.00	0.40	0.10	0.60	0.00	0.00	0.50	0.00	0.60	0.20	0.90	1.00	0.60	0.60	0.50	1.00	0.60	0.10	0.80	1.00	0.10	0.50	0.00	0.50	0.30	0.10	0.60		
	F ₁	0.93	0.00	0.00	0.11	0.00	0.00	0.40	0.13	0.14	0.00	0.00	0.41	0.00	0.50	0.57	0.13	0.73	0.57	0.60	0.57	0.39	1.00	0.57	0.08	0.59	0.50	0.08	0.40	0.20	0.13	0.14		
5	Acc.	1.00	1.00	0.80	0.50	0.00	0.00	0.80	0.50	0.00	0.00	0.80	0.40	0.00	0.00	0.80	0.50	0.00	0.90	0.50	0.80	0.40	0.70	0.50	0.80	0.50	0.10	1.00	0.80	0.50	0.00	0.00		
	F ₁	0.92	1.00	0.69	0.49	0.00	0.00	0.69	0.43	0.00	0.00	0.72	0.47	0.00	0.00	0.69	0.50	0.58	0.67	0.69	0.43	0.65	0.67	0.69	0.40	0.59	0.00	0.73	0.60	0.13	1.00	0.76	0.43	0.00
6	Acc.	1.00	1.00	0.80	0.50	0.00	1.00	0.80	0.70	0.00	1.00	0.90	0.80	0.00	1.00	0.90	0.80	0.00	1.00	0.80	0.80	0.90	1.00	0.80	0.70	0.80	1.00	0.80	1.00	0.20	1.00	0.70	0.00	
	F ₁	1.00	1.00	0.80	0.49	0.00	1.00	0.80	0.66	0.00	0.80	0.93	0.61	0.00	0.80	0.93	0.72	0.79	1.00	0.76	0.68	0.81	1.00	0.76	0.63	0.51	0.80	0.83	0.86	0.13	1.00	0.96	0.63	0.00
7	Acc.	0.40	0.00	0.50	0.60	0.70	0.00	0.40	0.30	0.00	1.00	0.20	0.60	0.00	0.40	0.30	0.00	0.00	0.20	0.70	0.10	0.00	0.30	0.40	0.00	0.40	0.30	0.40	0.50	0.40	0.50	0.00	0.00	
	F ₁	0.43	0.00	0.25	0.22	0.23	0.00	0.28	0.21	0.00	0.36	0.17	0.35	0.00	0.40	0.19	0.00	0.00	0.12	0.54	0.10	0.00	0.27	0.34	0.00	0.24	0.21	0.11	0.67	0.34	0.28	0.00	0.00	
8	Acc.	0.60	1.00	0.40	0.50	0.00	1.00	0.40	0.30	0.00	1.00	0.20	0.50	0.00	1.00	0.40	0.60	0.50	1.00	0.40	0.60	0.40	1.00	0.40	0.50	0.30	1.00	0.30	0.50	0.20	1.00	0.40	0.40	0.00
	F ₁	0.52	0.80	0.36	0.44	0.00	1.00	0.36	0.29	0.00	1.00	0.16	0.47	0.00	1.00	0.27	0.55	0.40	1.00	0.36	0.55	0.43	1.00	0.29	0.53	0.27	0.80	0.26	0.53	0.27	1.00	0.36	0.44	0.00
9	Acc.	1.00	1.00	0.70	0.40	0.00	1.00	1.00	0.30	0.40	1.00	0.30	1.00	0.00	1.00	1.00	0.70	1.00	1.00	1.00	0.90	1.00	1.00	0.90	1.00	0.20	1.00	1.00	0.80	1.00	0.80	0.00	0.00	
	F ₁	0.87	0.57	0.52	0.23	0.00	1.00	0.83	0.10	0.47	1.00	0.13	0.88	0.00	1.00	0.87	0.59	0.69	1.00	0.87	0.96	0.96	1.00	0.87	0.81	0.74	1.00	0.92	0.07	1.00	0.87	0.76	0.00	
all	WA	0.79	0.60	0.50	0.42	0.16	0.60	0.57	0.33	0.15	0.55	0.40	0.54	0.09	0.75	0.67	0.52	0.50	0.70	0.61	0.62	0.61	0.80	0.64	0.44	0.46	0.55	0.51	0.64	0.22	0.70	0.57	0.44	0.11

Table 7.4: Results on SYSU-3D-HOI dataset. “Bas.”/“Gan”/“Reg.”/“Occ.”/“Ref.” denote baseline / Generative adversarial networks/ Regression / training with occluded samples / reference case, Acc, F_1 and WA denote Accuracy, F_1 score and Weighted Accuracy, respectively. “None” denotes the case without occlusion. LA, RA, LL, RL denote the occlusion of left arm, right arm, left leg, right leg, respectively. Numbers in bold indicate best performance between “Rec.”/“Occ.”/“Ref.”.

Class	Metric	None		LA		RA		LA+RA		LL		RL		LL+RL		LA+LL		RA+RL	
		Baseline	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.	Ref.	Gan	Reg.	Occ.
1	Acc.	0.10	0.00	0.00	0.13	0.00	0.00	0.05	0.13	0.00	0.25	0.00	0.25	0.00	0.00	0.00	0.28	0.00	0.00
	F_1	0.10	0.00	0.00	0.14	0.00	0.00	0.06	0.07	0.00	0.22	0.00	0.25	0.00	0.00	0.00	0.28	0.00	0.00
2	Acc.	0.65	0.25	0.45	0.43	0.00	0.25	0.65	0.13	0.05	0.25	0.00	0.50	0.00	0.45	0.45	0.55	0.53	0.20
	F_1	0.59	0.22	0.36	0.46	0.00	0.33	0.63	0.12	0.08	0.40	0.00	0.54	0.00	0.40	0.44	0.45	0.34	0.33
3	Acc.	0.80	0.50	0.75	0.60	0.55	0.75	0.85	0.48	0.05	0.50	0.60	0.68	0.45	0.50	0.70	0.58	0.00	0.50
	F_1	0.77	0.67	0.69	0.54	0.21	0.67	0.77	0.44	0.05	0.67	0.57	0.59	0.12	0.57	0.65	0.56	0.00	0.67
4	Acc.	0.75	0.70	0.58	0.05	0.75	0.80	0.53	0.30	0.75	0.75	0.65	0.00	0.75	0.75	0.68	0.40	0.75	0.65
	F_1	0.80	0.75	0.73	0.61	0.08	0.86	0.84	0.57	0.43	0.86	0.76	0.71	0.00	0.86	0.79	0.70	0.52	0.86
5	Acc.	0.55	0.50	0.45	0.28	0.10	1.00	0.40	0.53	0.40	0.75	0.50	0.40	0.00	0.75	0.45	0.38	0.40	1.00
	F_1	0.48	0.50	0.34	0.21	0.10	0.80	0.35	0.40	0.15	0.67	0.30	0.38	0.00	0.67	0.30	0.32	0.10	0.80
6	Acc.	0.40	0.75	0.25	0.28	0.05	0.50	0.25	0.05	0.00	0.00	0.00	0.30	0.00	0.50	0.30	0.33	0.00	0.75
	F_1	0.38	0.60	0.31	0.27	0.08	0.50	0.28	0.03	0.00	0.00	0.00	0.28	0.00	0.40	0.27	0.31	0.00	0.60
7	Acc.	0.45	0.25	0.40	0.43	0.50	0.25	0.35	0.33	0.30	0.00	0.15	0.65	0.00	0.25	0.40	0.43	0.00	0.25
	F_1	0.54	0.40	0.35	0.39	0.18	0.40	0.36	0.29	0.19	0.00	0.20	0.61	0.00	0.40	0.47	0.42	0.00	0.29
8	Acc.	0.90	1.00	0.85	0.80	0.10	1.00	0.85	0.75	0.65	1.00	0.75	0.65	0.00	1.00	0.90	0.78	0.80	1.00
	F_1	0.88	1.00	0.81	0.79	0.13	1.00	0.83	0.74	0.71	1.00	0.77	0.77	0.00	1.00	0.84	0.79	0.75	0.89
9	Acc.	0.65	1.00	0.30	0.35	0.50	0.75	0.25	0.30	0.40	1.00	0.20	0.33	0.80	1.00	0.40	0.40	0.00	1.00
	F_1	0.67	0.62	0.39	0.31	0.19	0.60	0.31	0.28	0.31	0.73	0.28	0.34	0.19	0.73	0.47	0.30	0.00	0.80
10	Acc.	0.30	0.00	0.15	0.23	0.00	0.25	0.30	0.17	0.05	0.00	0.00	0.28	0.00	0.00	0.20	0.29	0.00	0.20
	F_1	0.24	0.00	0.12	0.27	0.00	0.22	0.21	0.07	0.06	0.00	0.00	0.22	0.00	0.00	0.20	0.40	0.00	0.20
11	Acc.	0.45	0.50	0.40	0.30	0.00	0.50	0.45	0.25	0.10	0.50	0.30	0.35	0.00	0.50	0.60	0.20	0.00	0.50
	F_1	0.34	0.31	0.31	0.24	0.00	0.31	0.32	0.14	0.10	0.33	0.23	0.23	0.00	0.36	0.43	0.15	0.00	0.40
12	Acc.	0.45	0.25	0.35	0.40	0.00	0.25	0.40	0.35	0.25	0.55	0.23	0.00	0.25	0.30	0.45	0.35	0.35	0.40
	F_1	0.45	0.40	0.34	0.41	0.00	0.00	0.22	0.45	0.18	0.20	0.24	0.24	0.00	0.29	0.25	0.39	0.09	0.00
all	WA	0.54	0.48	0.42	0.41	0.15	0.50	0.45	0.33	0.22	0.44	0.32	0.45	0.10	0.50	0.46	0.42	0.20	0.52

8. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this Thesis we presented our research within the area of human activity recognition from visual data. Specifically, we worked under the assumptions that human motion is (a) represented by a structured set of 3D skeleton joints; and (b) segmented, i.e., the beginning/ending of each given activity has been previously determined. We focused on the problem of occlusion, which, as we have argued, remains one of the main challenges in the broader field of research and it has not been given attention within the latest research efforts. Moreover, contemporary datasets have been created in ideal conditions, ignoring the specific challenges and limitations of a real environment.

Specifically, the first research part of this Thesis aimed to propose a novel approach for human activity recognition based on human motion. We considered 3D skeletal data as input to our methodology. To deal with the limitation of the input layer of convolutional neural networks, i.e., that it should be a still image, contrary to the temporal dimension of the skeletal data, we proposed a novel representation of skeletal motion. Our representation was created so as to capture and to maintain the spatio-temporal properties of the skeleton trajectories, and to adhere to the graph-based skeleton representation, as it was based on inter-joint distances of skeleton joints and also the structure of the image strictly followed the order of skeleton joints, based on their ids. We showed that this representation may be successfully applied to the problem of human activity recognition and was able to differentiate between different activities. We also showed that this representation achieved superior results, both in single-view and in cross-view cases compared to previous state-of-the-art approach.

In the second research part of this Thesis we proposed a transfer learning-based approach for human activity recognition, in an effort to go deeper in the area. Therefore, we proposed an approach for the fusion of handcrafted and learnt features from skeletal data, both resulting from human motion in the 3D space. The former were extracted based on angles, displacements and lengths over sets of joints, as they move during an activity, while the latter were learnt by the convolutional layers of a trained convolutional neural network. Our goal was to transfer the knowledge from a given problem to another, similar problem, both in the broader area of human activity recognition. Specifically, the problem at hand is the one of arm gesture recognition. We extracted handcrafted features from a relatively small arm gesture dataset. Then, we used a convolutional neural network which had been trained in a large-scale action dataset and used the output of its last dense layer as feature vector. This network served as feature extractor, providing context-aware deep features. We applied a late fusion approach using both feature vectors and trained a support vector machine for classification. We experimentally showed that the combination of both types of features led to a performance boosting, since learnt features were able to introduce a highly diverse representation, not captured by the handcrafted ones, thus knowledge was successfully transferred from a more generalized action recognition task to the more specific task of arm gesture recognition.

Starting from the third research part of this Thesis, our goal was to focus on the problem

of occlusion, typically encountered in real-life, practical human activity recognition applications and in many cases responsible for significant loss of information that is crucial for the recognition of activities. Since to the best of our knowledge there do not exist any publicly available data sets, we simulated occlusion by removing structured skeleton joints corresponding to body parts and under the strict assumptions that they remain occluded during the whole duration of the activity. We should herein emphasize that we consider our approach as a significantly harder case than the ones that are encountered in typical, real-life scenarios, wherein occlusion often does not affect a whole body part, but some of its joints, instead, while it may not affect the whole duration of an activity. So, in the third part of this Thesis, we firstly focused on studying the effect of occlusion in human activity recognition tasks and then proposed a data augmentation approach by including occluded samples within the training process. We experimentally verified that the inclusion of occluded samples was able to boost the performance of recognition in presence of occlusion and also that recognition was mostly affected in cases of arms' occlusion.

Then, in the fourth research part of this Thesis we presented an approach for human activity recognition under partial occlusion, wherein we managed to reconstruct missing skeleton joints by formulating the problem as a regression task. Specifically, we trained a convolutional recurrent neural network using occluded and non-occluded samples and then used it to reconstruct missing joints of occluded samples. For classification of reconstructed samples, we used a long short term memory network and evaluated using four publicly available datasets, experimenting with several different types of activities, demonstrating a significant boost of performance in a classification task, when using reconstructed samples instead of occluded ones, almost in any activity. The novelties of this work were: (a) the formulation of the problem of missing skeleton joints due to occlusion as a regression problem, implemented using novel deep neural network architecture; (b) contrary to other state-of-the-art works, occluded samples had not been part of the training process of the classifier; (c) the occluded parts were not visible from any camera, even in case of a multi-camera setup.

In the fifth and final research part of this Thesis we further evolved the approach for reconstructing human skeletons, strictly under the same assumptions as in the previous methodologies. Specifically, we addressed this problem using a generative adversarial network. Specifically, we trained a convolutional recurrent neural network to serve as the generator, with the aim of producing the complete skeleton by completing the missing parts. Its input consisted of raw 3D skeleton joint positions, with the joints corresponding to the occluded parts removed, while its output was a reconstructed skeleton, which was ultimately fed into a long short-term memory network for classification into one of the pre-defined activities. As the discriminator we used a simple long short-term memory network, trained using only non-occluded samples. We experimentally demonstrated that this approach could provide improved performance compared to those that we have previously presented.

We believe that the proposed approaches for dealing with partial occlusion could benefit several HAR tasks which take place into dynamic environments and are prone to occlusion. Apart from applications within ambient assisted environment i.e., where the goal is

to detect activities of daily living (ADLs)[99], or medical-related events that require some kind of attention or intervention (e.g., staggering, falling, etc.). In such scenarios, as we have already mentioned, occlusion may take place due to e.g., furniture or other people present. Moreover, it could also be of great utilization in AR environments and applications, since human motion is tied to the overall user experience and, HAR may act as a means of assessing user engagement e.g., when a visitor of a museum makes a phone call while interacting with an AR application, this should be an indicator of low engagement, while when she/he is reading in front of an AR screen, this should be an indicator of high engagement. Moreover, another important field of application is the one of surveillance, where the goal is to identify certain types of unwanted activity or abnormal events. In cases where a subject acts in public places, occlusion may take place due to obstacles or the presence of other people/crowd.

Future research work may target several aspects of the problem of HAR, focusing on occlusion. Our regression approach could be enhanced and improved e.g., by replacing the interpolation step herein used with a temporal augmentation approach such as the one that has been proposed by Kwon et al. [98]. Moreover, other state-of-the-art architectures such as transformers [111] or techniques that work with sequences, such as seq2seq [178] or time-series such as multivariate CNNs [122] could be introduced within the classification process. Also, the use of modern classifiers could be investigated for classification. For example, other possible research directions may include experiments with approaches such as Neural Dynamic Classification,[164] Dynamic Ensemble Learning,[10] Finite Element Machine for fast learning[151] or self-supervised learning [165]. Additionally, in the training phase it would be interesting to investigate the inclusion of occluded samples upon reconstruction.

Also, when dealing with the occlusion aspect of HAR, cases such as temporally partial occlusion could be investigated. Moreover, since handcrafted features have been experimentally proven to be able to boost recognition performance of deep approaches,[95] it would be interesting to experiment with other methodologies for feature extraction that are based on the geometry and the motion of skeletons, as the one proposed by Avola et al. [14]. Moreover, our approaches could be further evaluated upon performing experiments using larger datasets. We believe that it would be extremely interesting to perform real-life experiments into an assistive living environment or simulate occlusions occurring within such an environment by constructing masks using 3d models of real objects [150].

Finally, it is our belief that the findings of this study could be applied to real-world applications, provided there exists a reliable methodology for extraction a skeleton from visual data, under occlusion. It is evident that the Kinect sensor [92] is typically unable to achieve this, but deep learning approaches like OpenPose [25] demonstrate greater resilience to such challenges. Therefore, we anticipate that in the coming years, methodologies capable of extracting the skeleton even in occluded scenarios will be presented, rendering the herein proposed approaches practical and applicable to consumer applications.

ABBREVIATIONS - ACRONYMS

API	Application Programming Interface
ADL	Activity of Daily Living
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BoW	bag-of-features algorithms
CCTV	Closed Circuit Television
CF	Contextual Features
CNN	Convolutional Neural Network
CTC	Connectionist Temporal Classification
CRNN	Convolutional Recurrent Neural Network
CRC	Collaborative Representative Classifier
CS	Cross-Subject
CV	Cross-View
DL	Deep Learning
DWT	discrete wavelet transform
DMM	Depth Motion Maps
FNN	Feedforward Neural Network
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
GLCM	Grey-Level Co-Occurrence Matrix
HAR	Human Activity Recognition
HOG	Histogram of Oriented Gradients
HOF	Histograms of Optical Flow
HCI	Human Computer Interaction
HF	Handcrafted Features
HOI	Human-Object Interaction

HMM	Hidden Markov Models
LBP	Local Binary Patterns
JTM	Joint Trajectory Maps
LA	Left Arm
LDA	Linear Discriminant Analysis
LL	Left Leg
LSTM	Long Short Term Memory
MRF	Markov Random Fields
MP	Max Pooling
ML	Machine Learning
MTL	Multi task learning
MHI	Motion History Images
NN	Neural Network
PCA	Principal Component Analysis
RA	Right Arm
RNN	Recurrent Neural Networks
RL	Right Leg
RFID	Radio Frequency Identification
TSSI	Tree Structure Skeleton Image
TPSMMs	Temporal Pyramid Skeleton Motion Maps
VMT	Volume Motion Templates
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
STIP	Space Time Interest Points
STIF	Spatio Temporal Image Formation
SV	Single-View
SVM	Support Vector Machines
SLAM	Simultaneous Localization and Mapping
WA	Weighted Accuracy

WHDMM	Weighted Hierarchical Depth Motion Maps
ToF	Time of Flight

BIBLIOGRAPHY

- [1] Amor, B. B., Su, J., & Srivastava, A. (2015). Action recognition using rate-invariant analysis of skeletal shape trajectories. *IEEE transactions on pattern analysis and machine intelligence*, 38(1), 1-13.
- [2] Antunes, M., Aouada, D., & Ottersten, B. (2016, March). A revisit to human action recognition from depth sequences: Guided svm-sampling for joint selection. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1-8). IEEE.
- [3] Anxionnat, A., Voros, S., & Troccaz, J. (2018, November). Comparative study of RGB-D sensors based on controlled displacements of a ground-truth model. In *2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV)* (pp. 125-128). IEEE.
- [4] Al-Berry, M. N., Salem, M. A. M., Ebeid, H. M., Hussein, A. S., & Tolba, M. F. (2016). Fusing directional wavelet local binary pattern and moments for human action recognition. *IET Computer Vision*, 10(2), 153-162.
- [5] 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).
- [6] Abu-Bakar, S. A. (2019). Advances in human action recognition: an updated survey. *IET Image Processing*, 13(13), 2381-2394.
- [7] Ahonen, T., Hadid, A., & Pietikäinen, M. (2004). Face recognition with local binary patterns. In *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8* (pp. 469-481). Springer Berlin Heidelberg.
- [8] Aggarwal, J. K., & Ryoo, M. S. (2011). Human activity analysis: A review. *Acm Computing Surveys (Csur)*, 43(3), 1-43.
- [9] Aggarwal, A., Mittal, M., & Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. *International Journal of Information Management Data Insights*, 1(1), 100004.
- [10] Alam, K. M. R., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, 32, 8675-8690.
- [11] 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.

- [12] Antoshchuk, S., Kovalenko, M., & Sieck, J. (2018). Gesture recognition-based human–computer interaction interface for multimedia applications. *Digitisation of Culture: Namibian and International Perspectives*, 269-286.
- [13] Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., & Darrell, T. (2018). Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7564-7573).
- [14] 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.
- [15] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3), 346-359.
- [16] Batabyal, T., Chattopadhyay, T., & Mukherjee, D. P. (2015, September). Action recognition using joint coordinates of 3d skeleton data. In *2015 IEEE International Conference on Image Processing (ICIP)* (pp. 4107-4111). IEEE.
- [17] Belhadj, L. C., & Mignotte, M. (2016, September). Spatio-temporal fastmap-based mapping for human action recognition. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 3046-3050). IEEE.
- [18] Bradski, G. (2000). The openCV library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11), 120-123.
- [19] Bre, F., Gimenez, J. M., & Fachinotti, V. D. (2018). Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy and Buildings*, 158, 1429-1441.
- [20] Brock, A., Donahue, J., & Simonyan, K. (2018). Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.
- [21] Bulat, A., Yang, J., & Tzimiropoulos, G. (2018). To learn image super-resolution, use a gan to learn how to do image degradation first. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 185-200).
- [22] Bian, W., Tao, D., & Rui, Y. (2011). Cross-domain human action recognition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(2), 298-307.
- [23] Caetano, C., Sena, J., Brémond, F., Dos Santos, J. A., & Schwartz, W. R. (2019, September). Skelemotion: A new representation of skeleton joint sequences based on motion information for 3d action recognition. In *2019 16th IEEE international conference on advanced video and signal based surveillance (AVSS)* (pp. 1-8). IEEE.
- [24] Cao, X. Q., & Liu, Z. Q. (2014). Type-2 fuzzy topic models for human action recognition. *IEEE Transactions on Fuzzy Systems*, 23(5), 1581-1593.

- [25] 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).
- [26] Cai, X., Zhou, W., Wu, L., Luo, J., & Li, H. (2015). Effective active skeleton representation for low latency human action recognition. *IEEE Transactions on Multimedia*, 18(2), 141-154.
- [27] Chen, C., Jafari, R., & Kehtarnavaz, N. (2014). Improving human action recognition using fusion of depth camera and inertial sensors. *IEEE Transactions on Human-Machine Systems*, 45(1), 51-61.
- [28] Chen, C., Jafari, R., & Kehtarnavaz, N. (2015). A real-time human action recognition system using depth and inertial sensor fusion. *IEEE Sensors Journal*, 16(3), 773-781.
- [29] Chen, C., Liu, K., & Kehtarnavaz, N. (2016). Real-time human action recognition based on depth motion maps. *Journal of real-time image processing*, 12, 155-163.
- [30] Chen, L., & Nugent, C. D. (2019). Human activity recognition and behaviour analysis. Berlin/Heidelberg, Germany: Springer International Publishing.
- [31] Cheng, J., Sundholm, M., Zhou, B., Hirsch, M., & Lukowicz, P. (2016). Smart-surface: Large scale textile pressure sensors arrays for activity recognition. *Pervasive and Mobile Computing*, 30, 97-112.
- [32] Chen, C., Jafari, R., & Kehtarnavaz, N. (2016, March). Fusion of depth, skeleton, and inertial data for human action recognition. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 2712-2716). IEEE.
- [33] Chen, Y., Wang, L., Li, C., Hou, Y., & Li, W. (2020). ConvNets-based action recognition from skeleton motion maps. *Multimedia Tools and Applications*, 79, 1707-1725.
- [34] Chun, S., & Lee, C. S. (2016). Human action recognition using histogram of motion intensity and direction from multiple views. *IET Computer vision*, 10(4), 250-257.
- [35] Cippitelli, E., Gambi, E., Spinsante, S., & Flórez-Revuelta, F. (2016). Evaluation of a skeleton-based method for human activity recognition on a large-scale RGB-D dataset.
- [36] Chollet, F. (2015). Keras.[online] Available at: <https://github.com/fchollet/keras>. Accessed, 12(01), 2021.
- [37] Das, S., Dai, R., Koperski, M., Minciullo, L., Garattoni, L., Bremond, F., & Francesca, G. (2019). Toyota smarthome: Real-world activities of daily living. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 833-842).
- [38] Das Dawn, D., & Shaikh, S. H. (2016). A comprehensive survey of human action recognition with spatio-temporal interest point (STIP) detector. *The Visual Computer*, 32, 289-306.

- [39] Dogan, E., Eren, G., Wolf, C., & Baskurt, A. (2015, September). Activity recognition with volume motion templates and histograms of 3d gradients. In 2015 IEEE International Conference on Image Processing (ICIP) (pp. 4421-4425). IEEE.
- [40] Dolhansky, B., & Ferrer, C. C. (2018). Eye in-painting with exemplar generative adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7902-7911).
- [41] Dollár, P., Rabaud, V., Cottrell, G., & Belongie, S. (2005, October). Behavior recognition via sparse spatio-temporal features. In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (pp. 65-72). IEEE.
- [42] Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., & Del Bimbo, A. (2014). 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold. *IEEE transactions on cybernetics*, 45(7), 1340-1352.
- [43] 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.
- [44] Du, R., Turner, E., Dzitsiuk, M., Prasso, L., Duarte, I., Dourgarian, J., ... & Kim, D. (2020, October). DepthLab: Real-time 3D interaction with depth maps for mobile augmented reality. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (pp. 829-843).
- [45] Ding, S., & Qu, S. (2016, May). An improved interest point detector for human action recognition. In 2016 Chinese Control and Decision Conference (CCDC) (pp. 4355-4360). IEEE.
- [46] Egede, J., Valstar, M., & Martinez, B. (2017, May). Fusing deep learned and hand-crafted features of appearance, shape, and dynamics for automatic pain estimation. In 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017) (pp. 689-696). IEEE.
- [47] Emami, H., Aliabadi, M. M., Dong, M., & Chinnam, R. B. (2020). Spa-gan: Spatial attention gan for image-to-image translation. *IEEE Transactions on Multimedia*, 23, 391-401.
- [48] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018, April). Synthetic data augmentation using GAN for improved liver lesion classification. In 2018 IEEE 15th international symposium on biomedical imaging (ISBI 2018) (pp. 289-293). IEEE.
- [49] Foix, S., Alenya, G., & Torras, C. (2011). Lock-in time-of-flight (ToF) cameras: A survey. *IEEE Sensors Journal*, 11(9), 1917-1926.

- [50] Gallego, G., Delbrück, T., Orchard, G., Bartolozzi, C., Taba, B., Censi, A., ... & Scaramuzza, D. (2020). Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1), 154-180.
- [51] Gori, I., Aggarwal, J. K., Matthies, L., & Ryoo, M. S. (2016). Multitype activity recognition in robot-centric scenarios. *IEEE Robotics and Automation Letters*, 1(1), 593-600.
- [52] Giannakopoulos, Th., Spyrou, E., & Perantonis, S. (2019). Recognition of urban sound events using deep context-aware feature extractors and handcrafted features. In *Proc. of Mining Humanistic Data Workshop (MHDW)*, located at the Int'l Conference on Artificial Intelligence Applications and Innovations (AIAI).
- [53] Giannakos, I., Mathe, E., Spyrou, E., & Mylonas, P. (2021, June). A study on the Effect of Occlusion in Human Activity Recognition. In *The 14th Pervasive Technologies Related to Assistive Environments Conference* (pp. 473-482).
- [54] Gkalelis, N., Kim, H., Hilton, A., Nikolaidis, N., & Pitas, I. (2009, November). The i3dpost multi-view and 3d human action/interaction database. In *2009 Conference for Visual Media Production* (pp. 159-168). IEEE.
- [55] Gonzalez-Jorge, H., Riveiro, B., Vazquez-Fernandez, E., Martínez-Sánchez, J., & Arias, P. (2013). Metrological evaluation of microsoft kinect and asus xtion sensors. *Measurement*, 46(6), 1800-1806.
- [56] Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., ... & Memisevic, R. (2017). The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision* (pp. 5842-5850).
- [57] Guo, Y., Tao, D., Liu, W., & Cheng, J. (2016). Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4), 617-627.
- [58] Guo, Y., Chen, Y., Deng, J., Li, S., & Zhou, H. (2022). Identity-Preserved Human Posture Detection in Infrared Thermal Images: A Benchmark. *Sensors*, 23(1), 92.
- [59] Gu, C., Sun, C., Ross, D. A., Vondrick, C., Pantofaru, C., Li, Y., ... & Malik, J. (2018). Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6047-6056).
- [60] 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.
- [61] Graves, A., Fernández, S., Gomez, F., & Schmidhuber, J. (2006, June). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning* (pp. 369-376).

- [62] Halim, A. A., Dartigues-Pallez, C., Precioso, F., Riveill, M., Benslimane, A., & Ghoneim, S. (2016, September). Human action recognition based on 3d skeleton part-based pose estimation and temporal multi-resolution analysis. In 2016 IEEE international conference on image processing (ICIP) (pp. 3041-3045). IEEE.
- [63] Hai, P. T., & Kha, H. H. (2016, July). An efficient star skeleton extraction for human action recognition using hidden Markov models. In 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE) (pp. 351-356). IEEE.
- [64] Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- [65] Holte, M. B., Tran, C., Trivedi, M. M., & Moeslund, T. B. (2012). Human pose estimation and activity recognition from multi-view videos: Comparative explorations of recent developments. *IEEE Journal of selected topics in signal processing*, 6(5), 538-552.
- [66] 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.
- [67] Hu, J. F., Zheng, W. S., Lai, J., Gong, S., & Xiang, T. (2015). Exemplar-based recognition of human-object interactions. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(4), 647-660.
- [68] 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).
- [69] Hu, B., Yuan, J., & Wu, Y. (2016). Discriminative action states discovery for online action recognition. *IEEE Signal Processing Letters*, 23(10), 1374-1378.
- [70] Huynh-The, T., Banos, O., Le, B. V., Bui, D. M., Lee, S., Yoon, Y., & Le-Tien, T. (2015, October). PAM-based flexible generative topic model for 3D interactive activity recognition. In 2015 International Conference on Advanced Technologies for Communications (ATC) (pp. 117-122). IEEE.
- [71] Hussain, Z., Sheng, M., & Zhang, W. E. (2019). Different approaches for human activity recognition: A survey. *arXiv preprint arXiv:1906.05074*.
- [72] 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.
- [73] Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9(03), 90-95.

- [74] Imran, J., & Kumar, P. (2016, September). Human action recognition using RGB-D sensor and deep convolutional neural networks. In 2016 international conference on advances in computing, communications and informatics (ICACCI) (pp. 144-148). IEEE.
- [75] Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2013). Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7), 1325-1339.
- [76] 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.
- [77] Iosifidis, A., Marami, E., Tefas, A., Pitas, I., & Lyroudia, K. (2015). The MOBISERV-AIIA eating and drinking multi-view database for vision-based assisted living. *Journal of Information Hiding and Multimedia Signal Processing*, 6(2), 254-273.
- [78] Iosifidis, A., Tefas, A., & Pitas, I. (2015). Graph embedded extreme learning machine. *IEEE transactions on cybernetics*, 46(1), 311-324.
- [79] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
- [80] 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).
- [81] Ji, X., Cheng, J., & Feng, W. (2016, February). Spatio-temporal cuboid pyramid for action recognition using depth motion sequences. In 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI) (pp. 208-213). IEEE.
- [82] Jovanovic, M., & Campbell, M. (2022). Generative artificial intelligence: Trends and prospects. *Computer*, 55(10), 107-112.
- [83] Kashif, M. N., Raza, S. E. A., Sirinukunwattana, K., Arif, M., & Rajpoot, N. (2016, April). Handcrafted features with convolutional neural networks for detection of tumor cells in histology images. In 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI) (pp. 1029-1032). IEEE.
- [84] Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., ... & Zisserman, A. (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.
- [85] Ke, S. R., Thuc, H. L. U., Lee, Y. J., Hwang, J. N., Yoo, J. H., & Choi, K. H. (2013). A review on video-based human activity recognition. *Computers*, 2(2), 88-131.

- [86] Keselman, L., Iselin Woodfill, J., Grunnet-Jepsen, A., & Bhowmik, A. (2017). Intel realsense stereoscopic depth cameras. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops (pp. 1-10).
- [87] 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.
- [88] 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).
- [89] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
- [90] Krishnan, K., Prabhu, N., & Babu, R. V. (2016, June). ARRNET: action recognition through recurrent neural networks. In 2016 International Conference on Signal Processing and Communications (SPCOM) (pp. 1-5). IEEE.
- [91] 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.
- [92] Microsoft Corporation. (2014). Kinect for Windows Software Development Kit (SDK) [Software]. Retrieved from <https://www.microsoft.com/en-us/download/details.aspx?id=40278>
- [93] Kong, Y., & Fu, Y. (2015). Close human interaction recognition using patch-aware models. *IEEE Transactions on Image Processing*, 25(1), 167-178.
- [94] Koutník, J., Schmidhuber, J., & Gomez, F. (2014). Online evolution of deep convolutional network for vision-based reinforcement learning. In From Animals to Animats 13: 13th International Conference on Simulation of Adaptive Behavior, SAB 2014, Castellón, Spain, July 22-25, 2014. Proceedings 13 (pp. 260-269). Springer International Publishing.
- [95] Koutrintzes, D., Mathe, E., & Spyrou, E. (2022, February). Boosting the Performance of Deep Approaches through Fusion with Handcrafted Features. In ICPRAM (pp. 370-377).
- [96] Krohn, J., Beyleveld, G., & Bassens, A. (2019). Deep Learning Illustrated. Addison-Wesley Professional.
- [97] 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.

- [98] 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).
- [99] 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.
- [100] Leo, M., Spagnolo, P., D'Orazio, T., & Distant, A. (2004, November). Human activity recognition in archaeological sites by hidden markov models. In *Pacific-Rim Conference on Multimedia* (pp. 1019-1026). Berlin, Heidelberg: Springer Berlin Heidelberg.
- [101] Le, C. Q., Ngo, T. D., Le, D. D., Satoh, S. I., & Duong, D. A. (2015, October). Human action recognition from depth videos using multi-projection based representation. In *2015 IEEE 17th International Workshop on Multimedia Signal Processing (MMSP)* (pp. 1-6). IEEE.
- [102] Lee, J., Min, H. S., Seo, J. J., De Neve, W., & Ro, Y. M. (2014, July). Sub-sampled dictionaries for coarse-to-fine sparse representation-based human action recognition. In *2014 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.
- [103] Lee, S. H., Lee, D. W., & Kim, M. S. (2023). A Deep Learning-Based Semantic Segmentation Model Using MCNN and Attention Layer for Human Activity Recognition. *Sensors*, 23(4), 2278.
- [104] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.
- [105] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [106] Leightley, D., McPhee, J. S., & Yap, M. H. (2016). Automated analysis and quantification of human mobility using a depth sensor. *IEEE journal of biomedical and health informatics*, 21(4), 939-948.
- [107] Li, X., Song, D., Zhang, P., Yu, G., Hou, Y., & Hu, B. (2016, December). Emotion recognition from multi-channel EEG data through convolutional recurrent neural network. In *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)* (pp. 352-359). IEEE.
- [108] Lillo, I., Niebles, J. C., & Soto, A. (2016). A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1981-1990).
- [109] Li, C., Zhong, Q., Xie, D., & Pu, S. (2017, July). Skeleton-based action recognition with convolutional neural networks. In *2017 IEEE international conference on multimedia & expo workshops (ICMEW)* (pp. 597-600). IEEE.

- [110] 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.
- [111] Li, B., Cui, W., Wang, W., Zhang, L., Chen, Z., & Wu, M. (2021, May). Two-stream convolution augmented transformer for human activity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 35, No. 1, pp. 286-293).
- [112] 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.
- [113] Li, Y., Liu, S., Yang, J., & Yang, M. H. (2017). Generative face completion. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3911-3919).
- [114] Li, Z., & Li, D. (2022). Action recognition of construction workers under occlusion. *Journal of Building Engineering*, 45, 103352.
- [115] Liang, C., Qi, L., Chen, E., & Guan, L. (2016, September). Depth-based action recognition using multiscale sub-actions depth motion maps and local auto-correlation of space-time gradients. In *2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (pp. 1-7). IEEE.
- [116] Liu, A. A., Su, Y. T., Jia, P. P., Gao, Z., Hao, T., & Yang, Z. X. (2014). Multiple/single-view human action recognition via part-induced multitask structural learning. *IEEE transactions on cybernetics*, 45(6), 1194-1208.
- [117] Liu, F., Xu, X., Qiu, S., Qing, C., & Tao, D. (2015). Simple to complex transfer learning for action recognition. *IEEE Transactions on Image Processing*, 25(2), 949-960.
- [118] Liu, L., Shao, L., Li, X., & Lu, K. (2015). Learning spatio-temporal representations for action recognition: A genetic programming approach. *IEEE transactions on cybernetics*, 46(1), 158-170.
- [119] Liu, T., Wang, X., Dai, X., & Luo, J. (2016, March). Deep recursive and hierarchical conditional random fields for human action recognition. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1-9). IEEE.
- [120] Liu, A. A., Su, Y. T., Nie, W. Z., & Kankanhalli, M. (2016). Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(1), 102-114.
- [121] Liu, A. A., Xu, N., Nie, W. Z., Su, Y. T., Wong, Y., & Kankanhalli, M. (2016). Benchmarking a multimodal and multiview and interactive dataset for human action recognition. *IEEE Transactions on cybernetics*, 47(7), 1781-1794.

- [122] 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.
- [123] 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.
- [124] Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68, 346-362.
- [125] Liu, C., Hu, Y., Li, Y., Song, S., & Liu, J. (2017). Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*.
- [126] 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).
- [127] Liu, J., Shahroudy, A., Perez, M., Wang, G., Duan, L. Y., & Kot, A. C. (2019). Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(10), 2684-2701.
- [128] Liu, T., Sun, J. J., Zhao, L., Zhao, J., Yuan, L., Wang, Y., ... & Adam, H. (2022). View-invariant, occlusion-robust probabilistic embedding for human pose. *International Journal of Computer Vision*, 130(1), 111-135.
- [129] Lu, T., Zhang, S., Hao, Q., & Yang, J. H. (2012). Activity recognition in ubiquitous learning environment. *Journal of advances in information technology*, 3(1), 29-35.
- [130] Luo, H., & Lu, H. (2016, August). Multi-level sparse coding for human action recognition. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 1, pp. 460-463). IEEE.
- [131] McKinney, W. (2010, June). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, No. 1, pp. 51-56).
- [132] Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*.
- [133] Majumder, S., Mondal, T., & Deen, M. J. (2017). Wearable sensors for remote health monitoring. *Sensors*, 17(1), 130.
- [134] Mathe, E., Mitsou, A., Spyrou, E., & Mylonas, Ph. (2018). Arm Gesture Recognition using a Convolutional Neural Network. In *Proc. of Int'l Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*.

- [135] Mathe, E., Maniatis, A., Spyrou, E., & Mylonas, P. (2020). A deep learning approach for human action recognition using skeletal information. In *GeNeDis 2018: Computational Biology and Bioinformatics* (pp. 105-114). Springer International Publishing.
- [136] Masood, M., Nawaz, M., Malik, K. M., Javed, A., Irtaza, A., & Malik, H. (2023). Deep-fakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward. *Applied intelligence*, 53(4), 3974-4026.
- [137] Mould, N., Regens, J. L., Jensen III, C. J., & Edger, D. N. (2014). Video surveillance and counterterrorism: the application of suspicious activity recognition in visual surveillance systems to counterterrorism. *Journal of Policing, Intelligence and Counter Terrorism*, 9(2), 151-175.
- [138] Miao, J., Jia, X., Mathew, R., Xu, X., Taubman, D., & Qing, C. (2016, September). Efficient action recognition from compressed depth maps. In *2016 IEEE International Conference on Image Processing (ICIP)* (pp. 16-20). IEEE.
- [139] Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., & Ebrahimi, M. (2019). Edge-connect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*.
- [140] Nanni, L., Ghidoni, S., & Brahnam, S. (2017). Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71, 158-172.
- [141] Nguyen, T. V., Song, Z., & Yan, S. (2014). STAP: Spatial-temporal attention-aware pooling for action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(1), 77-86.
- [142] Nguyen, D. T., Pham, T. D., Baek, N. R., & Park, K. R. (2018). Combining deep and handcrafted image features for presentation attack detection in face recognition systems using visible-light camera sensors. *Sensors*, 18(3), 699.
- [143] NVidia Developer. Convolutional Neural Network (CNN). Retrieved from <https://developer.nvidia.com/discover/convolutional-neural-network>, 30 November, 2023
- [144] Ofodile, I., Helmi, A., Clapés, A., Avots, E., Peensoo, K. M., Valdma, S. M., ... & Anbarjafari, G. (2019). Action recognition using single-pixel time-of-flight detection. *Entropy*, 21(4), 414.
- [145] 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.
- [146] Papadopoulos, G. T., & Daras, P. (2016). Human action recognition using 3d reconstruction data. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8), 1807-1823.

- [147] Papadakis, A., Mathe, E., Vernikos, I., Maniatis, A., Spyrou, E., & Mylonas, P. (2019). Recognizing human actions using 3d skeletal information and CNNs. In Int'l Conf. on Engineering Applications of Neural Networks. Springer, Cham.
- [148] A. Papadakis, E. Mathe, E. Spyrou and P. Mylonas, A geometric approach for cross-view human action recognition using deep learning, *IEEE International Symposium on Image and Signal Processing and Analysis (ISPA)* (2019), pp. 258–263.
- [149] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research, 12, 2825-2830.
- [150] Peng, K., Roitberg, A., Yang, K., Zhang, J., & Stiefelhausen, R. (2023). Delving deep into one-shot skeleton-based action recognition with diverse occlusions. *IEEE Transactions on Multimedia*.
- [151] 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.
- [152] 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.
- [153] Pietikäinen, M. (2010). Local binary patterns. *Scholarpedia*, 5(3), 9775.
- [154] Qin, J., Liu, L., Zhang, Z., Wang, Y., & Shao, L. (2015). Compressive sequential learning for action similarity labeling. *IEEE Transactions on Image Processing*, 25(2), 756-769.
- [155] Rashid, M., Abu-Bakar, S. A. R., Mokji, M., & Abdu, A. (2012, November). Human action concentric video retrieval system using features weight updating method as relevance feedback. In 2012 IEEE International Conference on Control System, Computing and Engineering (pp. 366-370). IEEE.
- [156] Raschka, S. (2015). Python machine learning. Packt publishing Ltd.
- [157] Rahman, S., & See, J. (2016, March). Spatio-temporal mid-level feature bank for action recognition in low quality video. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1846-1850). IEEE.
- [158] Rahmani, H., Mahmood, A., Huynh, D., & Mian, A. (2016). Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(12), 2430-2443.
- [159] Ranasinghe, S., Al Machot, F., & Mayr, H. C. (2016). A review on applications of activity recognition systems with regard to performance and evaluation. *International Journal of Distributed Sensor Networks*, 12(8), 1550147716665520.

- [160] Rubine, D. (1991). Specifying gestures by example (Vol. 25, No. 4, pp. 329-337). ACM.
- [161] Roitberg, A., Perzylo, A., Somani, N., Giuliani, M., Rickert, M., & Knoll, A. (2014, December). Human activity recognition in the context of industrial human-robot interaction. In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific (pp. 1-10). IEEE.
- [162] Rodomagoulakis, I., Kardaris, N., Pitsikalis, V., Mavroudi, E., Katsamanis, A., Tsi-ami, A., & Maragos, P. (2016, March). Multimodal human action recognition in assistive human-robot interaction. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 2702-2706). IEEE.
- [163] Rosenfeld, A., & Ullman, S. (2016, June). Hand-object interaction and precise localization in transitive action recognition. In 2016 13th Conference on Computer and Robot Vision (CRV) (pp. 148-155). IEEE.
- [164] Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. IEEE transactions on neural networks and learning systems, 28(12), 3074-3083.
- [165] Rafiei, M. H., Gauthier, L. V., Adeli, H., & Takabi, D. (2022). Self-supervised learning for electroencephalography. IEEE Transactions on Neural Networks and Learning Systems.
- [166] Schuldt, C., Laptev, I., & Caputo, B. (2004, August). Recognizing Human Actions: A Local SVM Approach. In Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3-Volume 03 (pp. 32-36). IEEE Computer Society.
- [167] See, J., & Rahman, S. (2015, November). On the effects of low video quality in human action recognition. In 2015 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 1-8). IEEE.
- [168] 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.
- [169] Subetha, T., & Chitrakala, S. (2016, January). An adaptive framework for silhouette based human action recognition. In 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16) (pp. 1-5). IEEE.
- [170] Sheng, J. (2003). A study of adaboost in 3d gesture recognition. Department of Computer Science, University of Toronto.
- [171] Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.
- [172] 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).

- [173] Silva, V., Soares, F., Leão, C. P., Esteves, J. S., & Vercelli, G. (2021). Skeleton driven action recognition using an image-based spatial-temporal representation and convolution neural network. *Sensors*, 21(13), 4342.
- [174] Siddiqui, N., & Chan, R. H. (2017, July). A wearable hand gesture recognition device based on acoustic measurements at wrist. In 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 4443-4446). IEEE.
- [175] Sjarif, N. N. A., & Shamsuddin, S. M. (2015, December). Human action invariance-ness for human action recognition. In 2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA) (pp. 1-7). IEEE.
- [176] 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.
- [177] Srinivasan, V., Gul, S., Bosse, S., Meyer, J. T., Schierl, T., Hellge, C., & Samek, W. (2016, October). On the robustness of action recognition methods in compressed and pixel domain. In 2016 6th European Workshop on Visual Information Processing (EUVIP) (pp. 1-6). IEEE.
- [178] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- [179] Tasnim, N., Islam, M. K., & Baek, J. H. (2021). Deep learning based human activity recognition using spatio-temporal image formation of skeleton joints. *Applied Sciences*, 11(6), 2675.
- [180] Tölgyessy, M., Dekan, M., Chovanec, L., & Hubinský, P. (2021). Evaluation of the azure kinect and its comparison to kinect v1 and kinect v2. *Sensors*, 21(2), 413.
- [181] Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, 18(11), 1473-1488.
- [182] Traver, V. J., Latorre-Carmona, P., Salvador-Balaguer, E., Pla, F., & Javidi, B. (2016). Three-dimensional integral imaging for gesture recognition under occlusions. *IEEE Signal Processing Letters*, 24(2), 171-175.
- [183] Tripathi, R. K., Jalal, A. S., & Agrawal, S. C. (2018). Suspicious human activity recognition: a review. *Artificial Intelligence Review*, 50, 283-339.
- [184] Verma, P., Sah, A., & Srivastava, R. (2020). Deep learning-based multi-modal approach using RGB and skeleton sequences for human activity recognition. *Multimedia Systems*, 26(6), 671-685.

- [185] Vernikos, I., Mathe, E., Papadakis, A., Spyrou, E., & Mylonas, P. (2019, June). An image representation of skeletal data for action recognition using convolutional neural networks. In Proceedings of the 12th ACM International conference on pervasive technologies related to assistive environments (pp. 325-326).
- [186] Vishwakarma, D. K., & Singh, K. (2016). Human activity recognition based on spatial distribution of gradients at sublevels of average energy silhouette images. *IEEE Transactions on Cognitive and Developmental Systems*, 9(4), 316-327.
- [187] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & Van Mulbregt, P. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3), 261-272.
- [188] Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2, 28.
- [189] Wang, H., & Schmid, C. (2013). Action recognition with improved trajectories. In Proceedings of the IEEE international conference on computer vision (pp. 3551-3558).
- [190] 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).
- [191] 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.
- [192] Wang, C., Yang, H., & Meinel, C. (2016, July). Exploring multimodal video representation for action recognition. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 1924-1931). IEEE.
- [193] Wang, X., & Gupta, A. (2016, September). Generative image modeling using style and structure adversarial networks. In European conference on computer vision (pp. 318-335). Cham: Springer International Publishing.
- [194] 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).
- [195] 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.
- [196] Wang, L., Chen, W., Yang, W., Bi, F., & Yu, F. R. (2020). A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, 8, 63514-63537.
- [197] Wu, Q., Wang, Z., Deng, F., Chi, Z., & Feng, D. D. (2013). Realistic human action recognition with multimodal feature selection and fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4), 875-885.

- [198] Wu, S., Chen, Y. C., Li, X., Wu, A. C., You, J. J., & Zheng, W. S. (2016, March). An enhanced deep feature representation for person re-identification. In 2016 IEEE winter conference on applications of computer vision (WACV) (pp. 1-8). IEEE.
- [199] Wu, T., Gurram, P., Rao, R. M., & Bajwa, W. U. (2016, July). Clustering-aware structure-constrained low-rank representation model for learning human action attributes. In 2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP) (pp. 1-5). IEEE.
- [200] Xu, Z., Hu, R., Chen, J., Chen, C., Chen, H., Li, H., & Sun, Q. (2017). Action recognition by saliency-based dense sampling. *Neurocomputing*, 236, 82-92.
- [201] Xie, H., & Wu, Q. (2015, August). Human action recognition based on variation energy images features. In 2015 11th International Conference on Natural Computation (ICNC) (pp. 479-484). IEEE.
- [202] Xie, L., Wang, C., Liu, A. X., Sun, J., & Lu, S. (2017). Multi-touch in the air: Concurrent micromovement recognition using RF signals. *IEEE/ACM Transactions on Networking*, 26(1), 231-244.
- [203] Yadav, G. K., Shukla, P., & Sethfi, A. (2016, March). Action recognition using interest points capturing differential motion information. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1881-1885). IEEE.
- [204] Yadav, S. K., Tiwari, K., Pandey, H. M., & Akbar, S. A. (2021). A review of multi-modal human activity recognition with special emphasis on classification, applications, challenges and future directions. *Knowledge-Based Systems*, 223, 106970.
- [205] Yang, Y., Deng, C., Tao, D., Zhang, S., Liu, W., & Gao, X. (2016). Latent max-margin multitask learning with skeletons for 3-D action recognition. *IEEE transactions on cybernetics*, 47(2), 439-448.
- [206] Yang, Y., Deng, C., Gao, S., Liu, W., Tao, D., & Gao, X. (2016). Discriminative multi-instance multitask learning for 3D action recognition. *IEEE Transactions on Multimedia*, 19(3), 519-529.
- [207] 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.
- [208] 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.
- [209] Zhang, J., Gu, R., Ye, Q., & Ji, Y. (2012, December). Monocular human action recognition utilizing silhouette feature extraction and skin color detection. In 2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies (pp. 745-748). IEEE.

- [210] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., & Li, H. (2017). High-resolution image inpainting using multi-scale neural patch synthesis. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 6721-6729).
- [211] Yang, D., Wang, Y., Dantcheva, A., Garattoni, L., Francesca, G., & Brémont, 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.
- [212] Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M., & Do, M. N. (2017). Semantic image inpainting with deep generative models. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5485-5493).
- [213] Yuvaraju, M., Sheela, K., & Rani, S. S. (2015). Feature extraction of real-time image using Sift algorithm. International Journal of Research in Electrical & Electronics Engineering, 3(4), 01-07.
- [214] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2018). Generative image inpainting with contextual attention. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 5505-5514).
- [215] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., & Huang, T. S. (2019). Free-form image inpainting with gated convolution. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 4471-4480).
- [216] Yi, Z., Zhang, H., Tan, P., & Gong, M. (2017). Dualgan: Unsupervised dual learning for image-to-image translation. In Proceedings of the IEEE international conference on computer vision (pp. 2849-2857).
- [217] Zeng, Y., Fu, J., Chao, H., & Guo, B. (2019). Learning pyramid-context encoder network for high-quality image inpainting. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 1486-1494).
- [218] Zhang, Z., Wang, C., Xiao, B., Zhou, W., & Liu, S. (2013). Attribute regularization based human action recognition. IEEE Transactions on information forensics and security, 8(10), 1600-1609.
- [219] Zhang, W., Zhu, M., & Derpanis, K. G. (2013). From actemes to action: A strongly-supervised representation for detailed action understanding. In Proceedings of the IEEE international conference on computer vision (pp. 2248-2255).
- [220] Zhang, Z. (2012). Microsoft kinect sensor and its effect. IEEE multimedia, 19(2), 4-10.
- [221] Zhang, D. (2015, May). Recognizing human actions via silhouette image analysis. In The 27th Chinese Control and Decision Conference (2015 CCDC) (pp. 5870-5874). IEEE

- [222] Zhang, X., Zhang, H., Zhang, Y., Yang, Y., Wang, M., Luan, H., ... & Chua, T. S. (2015). Deep fusion of multiple semantic cues for complex event recognition. *IEEE Transactions on Image Processing*, 25(3), 1033-1046.
- [223] Zhang, Y., Cheng, L., Wu, J., Cai, J., Do, M. N., & Lu, J. (2016). Action recognition in still images with minimum annotation efforts. *IEEE Transactions on Image Processing*, 25(11), 5479-5490.
- [224] Zhang, S., Gao, C., Chen, F., Luo, S., & Sang, N. (2016). Group sparse-based mid-level representation for action recognition. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(4), 660-672.
- [225] 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.
- [226] Zhang, H. B., Lei, Q., Chen, D. S., Zhong, B. N., Peng, J., Du, J. X., & Su, S. Z. (2016). Probability-based method for boosting human action recognition using scene context. *IET Computer Vision*, 10(6), 528-536.
- [227] Zhang, H. B., Ma, Y., Guo, F., Zhang, M., & Lin, Y. (2016, May). Motion difference histogram for recognizing human action in video. In *2016 International Conference on Applied System Innovation (ICASI)* (pp. 1-4). IEEE.
- [228] Zhao, Y., Di, H., Zhang, J., Lu, Y., & Lv, F. (2016, July). Recognizing human actions from low-resolution videos by region-based mixture models. In *2016 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.
- [229] Zhu, L., Zhou, Q., & Li, Z. (2016, August). A new method of feature description for human action recognition. In *2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)* (Vol. 2, pp. 396-400). IEEE.
- [230] Zhu, S., Chen, W., Liu, F., Zhang, X., & Han, X. (2023). Human Activity Recognition Based on a Modified Capsule Network. *Mobile Information Systems*, 2023.
- [231] Zubair, A. R., & Alo, O. A. (2019). Grey level co-occurrence matrix (GLCM) based second order statistics for image texture analysis. *International Journal of Science and Engineering Investigations*, 8(93), 64-73.

APPENDIX A. SHORT CV, DECEMBER 2023

Ioannis Vernikos is a BSc graduate of the Department of Computer Science and Telecommunications of the University of Thessaly (2019). From 2019 until today he is working on his PhD thesis entitled “Human Activity Recognition under Occlusion” at the Department of Computer Science and Telecommunications of the University of Thessaly. He has worked as a research assistant at the Institute of Informatics and Telecommunications of the National Research Centre for Physical Sciences - “Demokritos” (2019-present), and as a fellow/external collaborator at the University of Thessaly (2020-2021). Since 2022 he is a fellow of the State Scholarship Foundation. He has 5 publications in international scientific journals and 13 publications in international scientific conferences. His research interests are in the areas of computer vision, machine learning and artificial intelligence.

APPENDIX B. LIST OF RELEVANT PUBLICATIONS

B.1 Publications in international scientific journals

J3. **I. Vernikos**, E. Spyrou, I.A. Kostis, E. Mathe and Ph. Mylonas, A Deep Regression Approach for Human Activity Recognition Under Partial Occlusion. *International Journal of Neural Systems*, 33(9), 2350047, World Scientific, 2023

Impact Factor 2022: 8.0, DOI: <https://doi.org/10.1142/S0129065723500478>

J2. **I. Vernikos**, T. Spyropoulos, E. Spyrou and Ph. Mylonas, Human Activity Recognition in Presence of Occlusion. *Sensors*, 23(10), 4899, MDPI, 2023

Impact Factor 2022: 3.9, DOI: <https://doi.org/10.3390/s23104899>

J1. 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*, 32, pp. 17181-17195, Springer, 2020

Impact Factor 2022: 6.0, DOI: <https://doi.org/10.1007/s00521-020-05162-5>

B.2 Publications in international scientific conferences

C9. **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

DOI: <https://doi.org/10.1145/3549737.3549786>

C8. 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

DOI: https://doi.org/10.1007/978-3-030-80568-5_10

C7. 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

DOI: <https://doi.org/10.1145/3453892.3461006>

C6. 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

DOI: <https://doi.org/10.1145/3389189.3397653>

C5. 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

DOI: <https://doi.org/10.1109/SEEDA-CECNSM.2019.8908363>

C4. **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

C3. **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

DOI: <https://doi.org/10.1109/SMAP.2019.8864848>

C2. **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

DOI: <https://doi.org/10.1145/3316782.3322740>

C1. 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

DOI: https://doi.org/10.1007/978-3-030-20257-6_44