Vision-based human gestures recognition for human-robot interaction

Osama Mazhar

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Systèmes Automatiques et Microélectroniques

École doctorale Information Structures Systèmes (I2S)

Unité de recherche Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier UMR 5506

VISION-BASED HUMAN GESTURES RECOGNITION FOR HUMAN-ROBOT INTERACTION

Présentée par Osama MAZHAR
Le 24 octobre 2019

Sous la direction de Andrea CHERUBINI et Sofiane RAMDANI

Devant le jury composé de

Antonis ARGYROS	PU	University of Crete	Rapporteur
Frédéric BOUCHARA	MCF-HDR	Université de Toulon	Rapporteur
Atilla BASKURT	PU	INSA Lyon	Examinateur
Christine AZEVEDO COSTE	DR	INRIA	Examinateur (président)
Andrea CHERUBINI	MCF-HDR	Université de Montpellier	Directeur de thèse
Sofiane RAMDANI	MCF-HDR	Université de Montpellier	Co-encadrant de thèse
Arash AJOUDANI	Chercheur	Istituto Italiano di Tecnologia	Invité
Vincent BONNET	MCF	Université de Paris Est Créteil	Invité

UNIVERSITÉ DE MONTPELLIER

I would like to dedicate this thesis to my loving parents ...

Acknowledgements

I would like to express my sincere gratitude to my thesis director Assoc. Prof. Andrea Cherubini for his continuous support of my PhD research, for his patience and motivation. It was a real privilege and an honour for me to have a share of his immense knowledge but also his extraordinary human qualities. I learned a great deal from him during this memorable journey of my life. I would also like to thank my co-supervisor Assoc. Prof. Sofiane Ramdani for his constant support, availability and constructive suggestions, which were determinant for the accomplishment of the work presented in this thesis. I gratefully acknowledge the funding received towards my PhD from French Education Ministry scholarship.

I am very grateful to Abderrehmane Kheddar for his valuable suggestions and guidance throughout my PhD. Many thanks to Prof. Abdul Rahim Ruzairi who motivated me during our discussion at Universiti Teknologi Malaysia, to purse a doctoral degree. Further acknowledgement and thanks is due to Prof. Antonis Argyros and Assoc. Prof. Frédéric Bouchara for taking time out of their busy schedules to review and provide feedback on my thesis. I would also like to warmly thank Prof. Atilla Baskurt, Christine Azevedo Coste, Assoc. Prof. Vincent Bonnet and Arash Ajoudani for that they accepted to be jury members of this thesis.

My deep appreciation goes out to our research team members and colleagues; Robin Passama and Benjamin Navarro. They helped me solve many scientific and technical problems with their practical support, suggestions and guidance. Benjamin's help in setting the robot experiment for my gesture-based robot control framework has made an invaluable contribution towards my PhD. I greatly appreciate the support and insightful discussions I had with Stephane Caron on many technical aspects of my research work.

I am indebted to my closer friends for their unconditional friendship, support and patience throughout these years.

Last, but not least, I would like to say my heartfelt thank to my father Mazharul Haque and my mother Khalida for their wise counsel and sympathetic ear. Finally to my wife Fatima, my sisters Afifa and Madiha, my brother Yousuf and to my niece Zainub, for their unconditional support, encouragement and love without which I would not have come this far.

Titre

Reconnaissance des gestes humains basée sur la vision pour l'interaction homme-robot

Résumé

Dans la perspective des usines du futur, pour garantir une interaction productive, sure et efficace entre l'homme et le robot, il est impératif que le robot puisse interpréter l'information fournie par le collaborateur humain. Pour traiter cette problématique nous avons exploré des solutions basées sur l'apprentissage profond et avons développé un framework pour la détection de gestes humains. Le framework proposé permet une détection robuste des gestes statiques de la main et des gestes dynamiques de la partie supérieure du corps.

Pour la détection des gestes statiques de la main, *openpose* est associé à la caméra Kinect V2 afin d'obtenir un pseudo-squelette humain en 3D. Avec la participation de 10 volontaires, nous avons constitué une base de données d'images, *opensign*, qui comprend les images RGB et de profondeur de la Kinect V2 correspondant à 10 gestes alphanumériques statiques de la main, issus de l'American Sign Language. Un réseau de neurones convolutifs de type « Inception V3 » est adapté et entrainé à détecter des gestes statiques de la main en temps réel.

Ce framework de détection de gestes est ensuite étendu pour permettre la reconnaissance des gestes dynamiques. Nous avons proposé une stratégie de détection de gestes dynamiques basée sur un mécanisme d'attention spatiale. Celle-ci utilise un réseau profond de type « Convolutional Neural Network - Long Short-Term Memory » pour l'extraction des dépendances spatio-temporelles dans des séquences vidéo pur RGB. Les blocs de construction du réseau de neurones convolutifs sont pré-entrainés sur notre base de données opensign de gestes statiques de la main, ce qui permet une extraction efficace des caractéristiques de la main. Un module d'attention spatiale exploite la posture 2D de la partie supérieure du corps pour estimer, d'une part, la distance entre la personne et le capteur pour la normalisation de l'échelle et d'autre part, les paramètres des cadres délimitant les mains du sujet sans avoir recourt à un capteur de profondeur. Ainsi, le module d'attention spatiale se focalise sur les grands mouvements des membres supérieurs mais également sur les images des mains, afin de traiter les petits mouvements de la main et des doigts pour mieux distinguer les classes de gestes. Les informations extraites d'une caméra de profondeur sont acquises de la base de données opensign. Par conséquent, la stratégie proposée pour la reconnaissance des gestes peut être adoptée par tout système muni d'une caméra de profondeur.

Ensuite, nous explorons brièvement les stratégies d'estimation de postures 3D à l'aide de caméras monoculaires. Nous proposons d'estimer les postures 3D chez l'homme par une approche hybride qui combine les avantages des estimateurs discriminants de postures 2D avec les approches utilisant des modèles génératifs. Notre stratégie optimise une fonction de

coût en minimisant l'écart entre la position et l'échelle normalisée de la posture 2D obtenue à l'aide d'*openpose*, et la projection 2D virtuelle du modèle cinématique du sujet humain.

Pour l'interaction homme-robot en temps réel, nous avons développé un système distribué asynchrone afin d'associer notre module de détection de gestes statiques à une librairie consacrée à l'interaction physique homme-robot *OpenPHRI*. Nous validons la performance de notre framework grâce à une expérimentation de type « apprentissage par démonstration » avec un bras robotique.

Title

Vision-based Human Gestures Recognition for Human-Robot Interaction

Abstract

In the light of factories of the future, to ensure productive, safe and effective interaction between robot and human coworkers, it is imperative that the robot extracts the essential information of the coworker. To address this, deep learning solutions are explored and a reliable human gesture detection framework is developed in this work. Our framework is able to robustly detect static hand gestures plus upper-body dynamic gestures.

For static hand gestures detection, *openpose* is integrated with Kinect V2 to obtain a pseudo-3D human skeleton. With the help of 10 volunteers, we recorded an image dataset *opensign*, that contains Kinect V2 RGB and depth images of 10 alpha-numeric static hand gestures taken from the American Sign Language. "Inception V3" neural network is adapted and trained to detect static hand gestures in real-time.

Subsequently, we extend our gesture detection framework to recognize upper-body dynamic gestures. A spatial attention based dynamic gestures detection strategy is proposed that employs multi-modal "Convolutional Neural Network - Long Short-Term Memory" deep network to extract spatio-temporal dependencies in pure RGB video sequences. The exploited convolutional neural network blocks are pre-trained on our static hand gestures dataset *opensign*, which allow efficient extraction of hand features. Our spatial attention module focuses on large-scale movements of upper limbs plus on hand images for subtle hand/fingers movements, to efficiently distinguish gestures classes. This module additionally exploits 2D upper-body pose to estimate distance of user from the sensor for scale-normalization plus determine the parameters of hands bounding boxes without a need of depth sensor. The information typically extracted from a depth camera in similar strategies is learned from *opensign* dataset. Thus the proposed gestures recognition strategy can be implemented on any system with a monocular camera.

Afterwards, we explore 3D human pose estimation strategies for monocular cameras. To estimate 3D human pose, a hybrid strategy is proposed which combines the merits of discriminative 2D pose estimators with that of model based generative approaches. Our method optimizes an objective function, that minimizes the discrepancy between position & scale-normalized 2D pose obtained from *openpose*, and a virtual 2D projection of a kinematic human model.

For real-time human-robot interaction, an asynchronous distributed system is developed to integrate our static hand gestures detector module with an open-source physical human-

robot interaction library *OpenPHRI*. We validate performance of the proposed framework through a teach by demonstration experiment with a robotic manipulator.

Contents

List of Figures **xv**

List of Tables **xxi**

Nomenclature **xxiii**

1 Introduction **1**

2 Background and State of the Art **5**

2.1 Safety in Collaborative Robotics 5

2.2 Gestures Detection in Human-Robot Interaction 7

2.3 Sign Language Detection 9

2.4 Dynamic Gestures Detection 10

2.4.1 Traditional Strategies 10

2.4.2 3D Convolutional Neural Networks 11

2.4.3 Multi-Modal Approaches 11

2.4.4 Optical-Flow Based Deep Methods 12

2.4.5 CNN-LSTM Strategies 12

2.4.6 Attention Based Methods 13

2.4.7 Pose-Driven Attention Mechanisms 13

2.4.8 Multi-Label Video Classification 14

2.4.9 Chalearn 2016 Gesture Recognition Challenge Strategies 15

2.5 3D Human Pose Estimation 16

2.5.1 Discriminative Approaches 16

2.5.2 Generative Approaches 17

2.5.3 Hybrid Approaches 17

2.6 Conclusion 18

3 Static Hand Gestures Detection **21**
3.1 Our Contributions . . . 21
3.2 Skeleton Extraction and Hand Localization . . . 22
3.2.1 Skeleton Extraction Module . . . 23
3.2.2 Image Acquisition and Hand Localization Module . . . 23
3.2.3 Asynchronous Integration of the Modules . . . 24
3.3 Convolutional Neural Network for Hand Gestures Detection . . . 26
3.3.1 Preparation of Dataset/Dataset recordings . . . 28
3.3.2 Background substitution and Preprocessing of the Hand Images . . . 29
3.3.3 Adapting Inception V3 to Gesture Recognition . . . 32
3.3.4 Quantification of the Trained CNN . . . 33
3.4 OpenPHRI Integration . . . 34
3.5 Example Industrial Application of the Proposed Framework . . . 37
3.6 Conclusion . . . 40

4 Dynamic Gestures Detection **43**
4.1 Our Contributions . . . 44
4.2 Datasets Description . . . 44
4.2.1 Chalearn 2016 Isolated Gesture Recognition Dataset . . . 44
4.2.2 Praxis Cognitive Assessment Dataset . . . 46
4.3 Spatial Attention Module . . . 46
4.3.1 Pose Extraction Module . . . 46
4.3.2 Focus on Hands Module . . . 52
4.4 Video Data Processing . . . 54
4.4.1 Features Extraction . . . 55
4.4.2 Label-wise Sorting and Zero-Padding . . . 55
4.4.3 Train-Ready Data Formulation . . . 56
4.5 CNN-LSTM Model . . . 56
4.5.1 Convolutional Neural Networks . . . 56
4.5.2 Long Short-Term Memory Networks . . . 57
4.5.3 CNN-LSTM for Gesture Recognition . . . 58
4.6 Training . . . 59
4.6.1 Multi-Stage Training . . . 59
4.6.2 Training Chalearn Dataset . . . 60
4.6.3 Training Praxis Dataset . . . 61
4.7 Results . . . 61
4.8 Conclusion . . . 64

5 3D Human Pose Estimation **67**
5.1 Our Contributions 68
5.2 Human Kinematic Model 68
5.3 2D Projection of Human Model 69
5.4 Scale and Position Normalization of the Skeleton 69
5.5 Formulation of Objective Function 71
5.6 Experimental Setup 71
5.7 Results 71
5.8 Conclusion 74

6 Discussion and Conclusion **75**

Bibliography **79**

Published Papers **95**

List of Figures

1.1 Illustration of an example human-robot interaction scenario where a user communicates with the robot through hand gestures 3

3.1 The overall vision pipeline of our static hand gestures detector. Grey lines represent one-time pathway for dataset preparation and convolutional neural network training. The route represented by blue lines is complied by our static hand gestures detector execution program. 22

3.2 Localization of hand through *openpose* is illustrated. The bounding box is titled with an angle that the forearm makes with horizontal, while the size of bounding box is determined by the mean depth value of the wrist joint. The mean depth value is computed by averaging the depth pixel values of a 6 $\times$ 6 matrix centered at the wrist joint. 24

3.3 The overall pipeline of our asynchronous distributed network for pHRI using hand gestures 25

3.4 Samples of the gestures considered for training. The labels represent the letters and the numbers taken from American Sign Language. The last gesture is one of the several **None** gestures included in the training set. . . . 27

3.5 A volunteer recording '7' gesture in the laboratory 28

3.6 The process of background substitution. 29

3.7 Samples of hand gesture images with original (labeled images) and substituted backgrounds (below originals). Note the remnants of the original backgrounds. This phenomenon is due to dilation of the binary masks. While it could be avoided by using techniques like chroma key, we do not intend to use a uniform background, to avoid bringing any extra apparatus in operation. In the experimental results (Sect. 3.3.4), we show that despite these remnants, gesture detection is highly accurate. 30

3.8 Image processing operations of histogram equalization, introduction of Gaussian and salt and pepper noise are performed on the training images. First row in each sub-image shows unprocessed image while the processed images are shown in the second rows. 31
3.9 Image processing operations applied to the training images include color-shift, zoom, shear, rotation, axes flip and position shift processes. 32
3.10 Plot of validation accuracy (top) and validation loss (bottom) 33
3.11 Normalized Confusion Matrix Quantified on Test-Set 34
3.12 Safe Physical Human Robot Interaction Setup 35
3.13 The FSM used for the experiment. A plus sign indicates an addition to the controller (a new constraint or new input) while a minus indicates a removal. 36
3.14 Screenshots from the robotic experiment by operators Op1 and Op2 (a) Op1 manually guiding the robot to a waypoint in the workspace. (b) Op1 records the way-points using Record gesture. (c) Op1 replay the taught waypoints by Replay gesture. (d) Op2 stands far from the robot so it moves with full speed. (e) Op2 stops the robot by applying external force (or accidental touch). (f) Op2 stands near the robot, so it moves slowly ensuring operator's safety. (g) Op2 gives Reteach command to the robot. (h) Op2 sets the new waypoints manually. (i) Op2 gives Record command. (j) Op2 stops the robot by Stop gesture. (k) Op2 resumes the robot operation by Resume gesture. (l) Op1 ends the robot operation by giving End command. 38
3.15 Experimental results. From top to bottom: hand gesture detection (dashed lines correspond to detection instants and plain line to the activation signals), control point translational velocity, external force at the end-effector, distance between the camera and the closest human body part and velocity scaling factor computed by OpenPHRI to slow down the motion. 39

4.1 Illustration of our overall proposed strategy for dynamic gestures detection. We employ *openpose* to extract raw 2D upper-body pose plus hands key-points of the person from monocular images. Then, raw hand key-points, original RGB input frames and our learning-based hands depth estimators f_l and f_r are passed to Focus on Hands Module which performs filtering of hand key-points and subsequently crops hands images of the person performing gestures. Pose Extraction Module performs skeleton filtering by interpolating missing skeletal joints coordinates as well as Gaussian smoothing of the pose. It also perfoms "Scale and Position Normalization" exploiting our learning-based skeleton depth estimator f_n. For visualization purpose, offset values are added to the normalized skeleton in the display image. Then, normalized skeleton is passed to Features Extraction block which performs pose augmentation. Cropped left/right hands images and *augmented pose vector* are then fed into our proposed CNN-LSTM that outputs framewise static hand gestures labels and dynamic gesture label for the input video. . . 45

4.2 Illustration of the proposed coordinate replacement strategy. All variables are assumed non-zero unless stated otherwise. (a) If a joint coordinate, say p_{2K} in the new frame K is zero, while $(p_{2k})_{k=1}^{k=K-1} \neq 0$, then p_{2K} is replaced by the immediate previous non-zero value i.e., p_{2K-1} and coordinate replacement counter for this joint i_2 is incremented by 1. This process may be repeated consecutively if same conditions persist until i_2 equals to K. If a non-zero value of p_{2K} reappears before i_2 equals to K, i_2 is reset. (b) If same conditions continue as in a, while the value of i_2 reach limit K, all values corresponding to considered joint i.e., $(p_{2k})_{k=1}^{k=K}$ in this case, are replaced with 0 and coordinate replacement counter is reset. (c) If all previous values corresponding to a joint e.g., $(p_{2k})_{k=1}^{k=K-1}$ are zero, while $p_{2K} \neq 0$, then all values in the window corresponding to the considered joint are replaced by non-zero p_{2K} in this case. 48

4.3 An illustration of proposed skeleton filtering in function. The frames are arranged from left to right with respect to their appearance in time. It can be observed that the output (frame with blue boundary) of our filtering strategy copes with the missing joint coordinates in the window. Gaussian smoothing is also applied on the joints which absorbs jitter in the raw output skeleton from *openpose*. The opted size of window is 7 frames, while only 5 frames are shown for clarity. The bounding boxes on the hands are extracted as a part of proposed *spatial attention module* detailed in Section 4.3.2 49

4.4 An illustration of scale ambiguity in monocular images; (a) Exhibits a scene where two humans of the identical physical form are standing at different distances from the camera. (b) Represents an image taken from the monocular camera. It is evident that no depth information can be perceived from such an image. To emphasize this ambiguity, we place a virtual table beneath the feet of apparently smaller human which fakes him as a child who stands besides an adult at the same distance from the camera. (c) Highlights our problem more evidently, that the scale ambiguity in monocular images makes it hard to determine the region-of-interest, driven by d_1 and d_2 variables, around hands of the persons for gestures detection. 50

4.5 Manual features extraction/pose augmentation from upper body skeleton for scale-normalization is presented. In the left image, we show 8 upper-body joint coordinates (red circles), vectors joining these joints (black lines) and angles between these vectors (shown in green). From all non-zero upper-body coordinates, we compute a *line of best fit* (in blue) which can clearly be seen in the left image. In the right, we show all the augmented vectors (in purple) between unique pairs of all upper-body joints. The angles between augmented vectors and the line of best fit are also computed which are not visible in this image. The reference neck depth coordinate (blue circle), obtained through Kinect V2 depth map, is also shown in the right figure against which the 97 components augmented pose vector is mapped to estimate approximate depth of the person. 51

4.6 Illustration of augmented pose vector features for hands bounding boxes size estimation. 54

4.7 Illustration of the proposed CNN-LSTM network for dynamic gestures recognition 59

4.8 Illustration of the training curves for 47 top gestures from Chalearn 2016 isolated gestures recognition dataset. The obtained model is utilized for weights initialization of the network for classification of all 249 gestures. . 60

4.9 Training curves of Praxis gesture dataset. 61

4.10 Training curves of the proposed CNN-LSTM network for all 249 gestures of Chalearn 2016 isolated gestures recognition dataset. The network is trained in four phases which can be distinguished by the vertical lines in the plot. . 62

4.11 Illustration of the confusion matrix/heat-map of the proposed model evaluated on test set of Chalearn 2016 isolated gestures recognition dataset. It is evident that most samples in the test set are recognized with high accuracy for all 249 gestures (diagonal entries, 86.75% overall). 63
4.12 Illustration of the confusion matrix for Praxis gestures dataset evaluated on 501 (original and mirrored) test samples. The diagonal values represents number/contribution of videos of the corresponding class in the dataset. Accuracy values for individual gesture labels are displayed on the bottom row and the last column. 65

5.1 An illustration of our proposed 3D pose estimation strategy in operation. . . 67
5.2 Illustration of position & scale-normalization procedure. 70
5.3 Illustrations of 3D pose estimations with movements mainly on upper-limbs. The skeletons in the second row with green lines (red dots) are 2D projections of the kinematic human model, while yellow dots in the same images are position and scale normalized 2D pose obtained from *openpose*. The optimization problem solves to minimize the discrepancy between these two skeletons to estimate 3D human pose. 72
5.4 3D pose estimations with full-body articulations. It can be seen that proposed strategy fails to solve depth ambiguity for (lower) limbs movements in *saggital* plane. Also the optimization falls into local minima in some cases as shown. 73

List of Tables

4.1 Comparison of the reported results with ours on Chalearn 2016 isolated gestures recognition dataset. The challenge results are published in [1]. Order of the entries is set with respect to the test results. 64
4.2 Comparison of reported dynamic gestures detection results on Praxis gestures dataset. The author in [2] also achieved best results with a CNN-LSTM network. 64

5.1 Joint limits in radians along *X, Y* and *Z axes*. Longer joint names are abbreviated such as *LS* for *Left Shoulder*. Same applies for *Elbow, Hip* and *Knee* joints. 68

Nomenclature

Acronyms / Abbreviations

API Application Programming Interface

ASL American Sign Language

BoW Bag of Words

CIFAR Canadian Institute For Advanced Research

CNN Convolutional Neural Network

CNN-LSTM Convolutional Neural Network - Long Short-Term Memory

COCO Common Objects in Context

FC Fully Connected

FOA Focus on Hands

FRI Fast Research Interface

FSM Finite-State Machine

GPU Graphics Processing Unit

GRU Gated Recurrent Unit

HOF Histogram of Optical Flow

HOG Histogram of Oriented Gradients

ILSVRC ImageNet Large Scale Visual Recognition Challenge

ISO International Standards Organization

ISO/TS International Standards Organization/Technical Specifications

LRCN Long-Term Recurrent Convolutional Network

LSTM Long Short-Term Memory

MBH Motion Boundary Histogram

MCI Mild Cognitive Impairment

MLP Mult-Layered Perceptron

MNIST Modified National Institute of Standards and Technology

MoCap Motion Capture

MSR Microsoft Research

NTU Nanyang Technological University

NUI Natural User Interface

OpenNI Open Natural Interaction

OpenPHRI Open Physical Human Robot Interaction

R-CNN Region-Convolutional Neural Network

ReLU Rectified Linear Unit

ResC3D Residual Convolutional 3D

ResNet Residual Networks

RGB-D Red Green Blue - Depth

RNN Recurrent Neural Network

SCI Severe Cognitive Impairment

SDK Software Development Kit

SIFT Scale Invariant Feature Transform

SVM Support Vector Machines

TUI Tangible User Interface

VGG Visual Geometry Group

V-REP Virtual Robot Experimentation Platform

YOLO You Only Look Once

Chapter 1

Introduction

The advent of Industry 4.0, as a modern trend of automation and data exchange in the manufacturing industry, has proposed the concept of smart factories of the future [3]. This evolving industry demands a more effective and involved collaboration between humans and robots, where each partner can constructively utilize the strengths of the others to increase productivity and work quality [4]. Safety of the human coworkers and an efficacious interaction between humans and robots are key factors of success in such an industrial setting [5, 6]. To ensure safety, the ability of the robot to detect an external force, differentiate between intended and accidental forces and to adapt to the rapidity of the human coworker is essential [7]. Nevertheless, the sense of vision is also imperative for modern collaborative robots to monitor the behavior and actions of their human coworkers for communicating or preventing accidents [8].

Generally, robots are designed and programmed to perform specialized tasks. Hence, it is difficult for an unskilled worker to reprogram the robot for a new task [9]. The traditional robot teaching methods are tedious, non-intuitive and time consuming. Multi-modal interfaces that include vision-based gesture detection frameworks, constitute instances of natural and tangible user interfaces (NUIs and TUIs). NUIs exploit the user's pre-existing knowledge and actions – related to daily practices – to offer natural and realistic interactions. This allows humans to directly interact with robots through voice, gestures, touch and motion tracking rather than instructing them the same by typing commands [10].

In many industrial settings, it is not convenient to communicate through speech because of interference produced by machines operations. The conventional use of teach pendants is itself too complicated for new users to learn. Portable devices are always required to be charged almost on daily-basis and may also have complex menu trees or networking problems in the interaction software. A well known study [11] shows that 93% of the human communication is non-verbal and 55% of this is accounted for elements like body posture

and facial expressions etc. In this perspective, capabilities like gesture recognition and human behavior understanding may be extremely useful for a robotic system in physical human-robot interaction scenarios [12]. To unburden the human coworker from carrying any extra device while s/he manoeuvres the robot, in physical human-robot interactions like in teach-by-demonstration applications, gestures are considered to be natural and intuitive ways to communicate/interact with the robot [13].

Gestures transmit key information and complement natural human conversation. Human gestures have been classified into different categories like manipulation, semaphores, deictic, gesticulation and sign languages [14]. Manipulative gestures are performed to mimic the movements that control/manipulate objects or the entities being manipulated. Semaphores define signals communicated through flags, lights or arms. Deictics are pointing gestures while gesticulations describe the free-form arbitrary and inexplicable movements that are accompanied by conversational speech. Sign languages are characterized by complete grammatical and lexical specification. In between gesticulations and sign languages, there are language-like gestures that include pantomimes and emblems. Pantomimes replace the speech with iconic and motion replica of the referent while emblems are codified gesture representations not driven by any formal grammar.

Gestures can also be classified into two types; static and dynamic gestures [15]. In the perspective of human hands, static gestures are defined by hand and finger posture at a certain moment in time [16] while dynamic gestures (in video sequences) additionally involve movement of body parts e.g., waving of hand. The temporal dimension in dynamic gestures causes gesture recognition to be a challenging problem due to its high-dimensional and rich input space plus model complexity [17, 18]. Typically, local frame-level (motion) features are aggregated into mid-level video representations or temporal sequence modeling is performed through either traditional methods or lately with the help of deep networks. Recently, attention-based methods [19], inspired by human perception, are proposed that only focus selectively on parts of the scene for information acquisition at specific places and time. The consumer depth cameras have been quite popular among the computer-vision and robotic researchers, providing complementary depth information which helps in tasks which were considered harder earlier [20].

The aim of this research is to develop vision-based robust gestures detection strategies suitable for human-robot interaction tasks. In the earlier parts of this thesis, Microsoft Kinect V2 is employed as the main vision sensor, while we steer the attention of this work towards building strategies that can be exploited with a monocular camera in later sections of this thesis. Figure 1.1 shows an example scenario where a user is interacting with a robot through gestures.

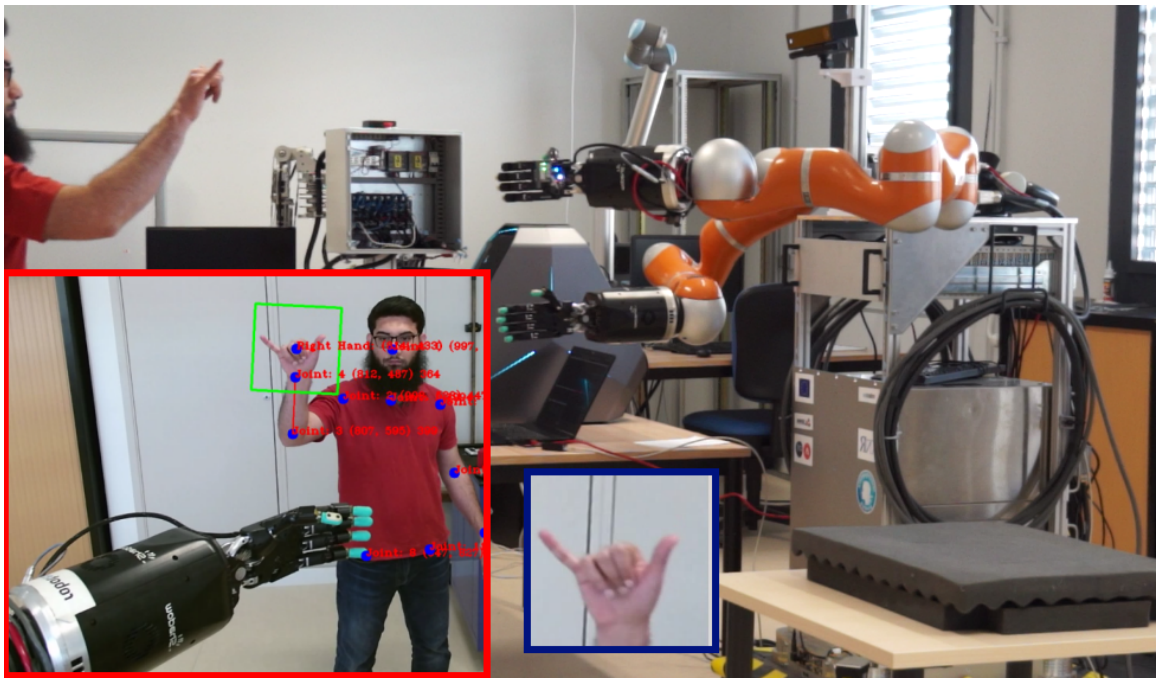


Figure 1.1 Illustration of an example human-robot interaction scenario where a user communicates with the robot through hand gestures

In the beginning, we propose a real-time robust and background independent static hand gesture detection strategy based on transfer learning [21] with convolutional neural networks which has already been published in [22, 23]. The intuitiveness of our system comes from the fact that the human does not need to wear any specific suits (Motion capture suits or inertial sensors) neither to carry a specialized remote control nor to learn complicated teach pendant commands. Such additional burdens would make the interaction unnatural [24]. The proposed static hand gestures detection system is integrated with a physical human-robot interaction library *OpenPHRI* [25] for robot control. Primarily, it provides a natural means for robot programming and reprogramming through hand gestures, while at the same time ensures safety of the human coworker by complementing the standard collaborative modes in *OpenPHRI*.

Subsequently, we propose a multi-stream spatial-attention based dynamic gesture recognition network which is a combination of convolutional neural network and long short-term memory network (CNN-LSTM). The proposed strategy is designed to work with RGB images/videos from conventional/monocular camera. In the perspective of human-robot interaction, we only assume gestures performed by a single person and no multi-person interactions are considered. The upper-body 2D human pose is exploited as one of the modalities/streams in the presented gesture recognition system. Our spatial-attention module,

localizes human hands in the scene, determines the size of bounding boxes and focus on the hand images which are exploited as another modality in our network from pure RGB images/videos. We utilize our background invariant static hand gestures detection convolutional network [23] to extract 1024 elements deep features from the cropped hand images. The obtained hand representations and the "augmented upper-body pose" are concatenated in the intermediate layers of the proposed network architecture. Our network is trained on *Chalearn 2016 Looking at People isolated gesture recognition* dataset [26] and on *Praxis cognitive assessment* dataset [2] demonstrating the state-of-the-art results on pure RGB input.

Thereupon, the task of 3D pose estimation with a monocular camera is explored. The proposed strategy is a hybrid approach that combines the merits of a discriminative 2D pose extractor with that of a model based generative pose estimator for physically plausible estimations.

The main contributions of this thesis are:

- A robust **static hand gestures detector** with Kinect V2 RGB-D vision sensor, trained on our *opensign* sign language dataset with background substitution method.
- An asynchronous distributed system which allows real-time static hand gestures detection demonstrated through a **human-robot interaction** experiment.
- A spatial-attention based multi-modal **dynamic gestures recognition** strategy employing CNN-LSTM network with the state-of-the-art performance on Chalearn 2016 isolated gesture recognition dataset [1] on pure RGB input.
- A hybrid **3D human pose extraction** strategy for a monocular camera as an optimization approach, which is validated through online human pose estimation experiments.

The rest of this thesis is organized as follows. Background and state of the art is presented in Chapter 2. Our work on static hand gestures detection is detailed in Chapter 3. Dynamic gestures recognition strategy is explained in Chapter 4. 3D human pose estimation is described in Chapter 5 while we conclude in Chapter 6.

Chapter 2

Background and State of the Art

The emerging concept of cyber-physical system is a mechanism that employ extensive automation and manage self-organization of machines and component parts in complex manufacturing scenarios, using different sensor modalities [3]. The primary role of human workers in such a setting will be to dictate a production strategy and to supervise its implementation by the robotic systems. We hereby survey the state of art in the context of vision-based gesture recognition for safe human-robot interaction in cyber-physical systems.

We will review the literature on safety in collaborative robotics (Section 2.1), gesture detection in human-robot interaction (Section 2.2) and sign language detection (Section 2.3) relevant to our research. We thereafter go through the state of the art in activity/dynamic-gestures recognition (Section 2.4) followed by literature on 3D human pose estimation with monocular camera (Section 2.5).

2.1 Safety in Collaborative Robotics

A recent survey on human-robot collaboration in industrial settings is presented in [10]. The authors talk about human safety citing several ISO standards, discuss intuitive interfaces for robot programming/collaboration and explore different industrial applications of human-robot collaboration. With regards to safety, they recall the four collaborative modes from ISO 10218-1/2 and ISO/TS 15066 [27–29]: “Safety-rated monitored stop”, “Hand guiding”, “Speed and separation monitoring” and “Power and force limitation”. Since in this work we addressed the first and third, let us now focus on works related to these modes.

In [30], the authors present a tire workshop assistant robot. SICK S300 laser sensors are utilized for navigation, obstacle avoidance and human detection. The authors define three areas surrounding the robot namely “Safe area”, “Collaboration area” and “Forbidden

area". The main disadvantage of this technology[31] is that several thousands of reflective landmarks are required for reliable navigation of the robot in a cluttered environment.

The authors of [32] thoroughly discuss several aspects of *speed and separation monitoring* in collaborative robot workcells. They analyze laser-based human tracking systems. The human coworkers are detected through centroid estimation of the detected objects and as the authors state, this varies based on the motion of legs, shifting of clothes, and sensor noise. The authors emphasize the technological advancement of safety-rated cameras and on-robot sensing hardware for enabling speed and separation monitoring in unstructured environments. Moreover, the importance of human-specific identification and localization methods is discussed for reliable physical human robot collaboration.

In [33], the authors present the preliminary results of their research on sensor-less radio human localization to enable *speed and separation monitoring*. A wireless device-free radio localization method is adopted with several nodes connected in mesh configuration, non-regularly spread over a large indoor area, so that the human-operator being localized does not need to carry an active wireless device. The concept of user tracking in wireless sensor networks is extended in [34]. This study considers the availability of the source attached to the human coworker's body in the industrial scenario.

The idea of trajectory dependent safety distances is proposed in [35] to attain dynamic *speed and separation monitoring*. This method avoids fixed extra safety clearances and is optimized with respect to the functional task at hand. Alternative sensing modalities for *speed and separation monitoring* include motion capture systems [36] and vision based depth cameras [37, 38]. In this regard, [31] compares structured light depth cameras and stereo-vision cameras for mobile robot localization in the industry.

As all these works highlight, an advantage of vision over other sensors is that it does not require structuring the environment and/or operator. Furthermore, it is generally more rich, portable (a fundamental feature for mobile robots) and low-cost, even when depth is also measured by the sensor (as with Microsoft Kinect). While at present Kinect is far from being certifiable for safety, we are confident that in the near future similar RGB-D sensors will. For all these reasons, we decided to use RGB-D vision for addressing *safety-rated monitored stops* and *speed and separation monitoring*. As in [32], we adopt the idea of human-specific localization to effectively identify the presence of humans in cluttered environments. Our contributions on safety will be detailed in the subsequent section after reviewing literature on gesture detection in human-robot interaction.

2.2 Gestures Detection in Human-Robot Interaction

Once safety is guaranteed, collaboration is possible. To this end, researchers have proposed to use body gestures for communicating with a robot. The literature on gesture detection in human-robot interaction scenarios is enormous. Here, we focus on works that rely mainly on RGB-D sensing.

A task-oriented intuitive programming procedure is presented in [39] to demonstrate human-like behavior of a dual-arm robot. The authors decompose complex activities in simpler tasks that are performed through task-oriented programming where the focus is given to "what to be done, rather than how to do it". Moreover, through the development of intuitive human interfaces, high level commands are transferred to a sequence of robot motion and actions. For human-robot interaction, the authors use Kinect V1 [40] to extract human skeletal coordinates for gesture detection, and the built-in microphone array of Kinect V1 to detect the oral commands. Whole body gestures (extended arms) are used to achieve robot motion in a dashboard assembly case. Although the idea of task decomposition and controlling the robot through human gestures is beneficial, the considered gestures, as in [41], are counterintuitive and tiring.

The authors of [42] present methods for obtaining human worker posture in a human-robot collaboration task of abrasive blasting. They compare the performance of three depth cameras, namely Microsoft Kinect V1, Microsoft Kinect V2 [43] and Intel RealSense R200 [44]. Kinect V1 uses a structured light approach to estimate the depth map, Kinect V2 is a time-of-flight sensor, while RealSense R200 has a stereoscopic infra-red setting to produce depth. In the blasting process, the abrasives are suspended in the air or fill the surrounding environment, and significantly decrease scene visibility. The use of image-based methods to extract human worker pose is challenging in such environments. The experimental observations suggest that Kinect V1 performs best in the real blasting environment, although no concrete reason could explain this. They also present a novel camera rig with an array of four Kinect V1 to cover a 180° horizontal field of view. The use of Kinect V1, to extract human pose for a marker-less robot control method is also presented in [45].

In [46], the authors present an online robot teaching method that fuses speech and gesture information using text. Kinect V2 localizes the position of hands in the scene, while their orientation is measured by an inertial measurement unit. The gesture and speech data are first converted into a description text, then a text understanding method converts the text to robot commands. The proposed method is validated by performing a peg-in-hole experiment, placing wire-cut shapes, and an irregular trajectory following task.

To ensure safe interaction, [47, 48] proposes a virtual reality training system for human-robot collaboration. A virtual game simulation is developed for real-time collaboration

between industrial robotic manipulators and humans. A realistic virtual human body, including a simple first person shooter view, simulates the user's vision. A head mount display and a Kinect V1 track the human head and skeleton pose respectively. Several interaction tasks are accomplished. These include selection of objects, manipulation, navigation and robot control. This technique is useful to establish the acceptability of a collaborative robot among humans in a shared workspace, as well as to tackle mental safety issues.

In [9], the authors present a strategy to use speech and a Wii controller to program a Motoman HP6 industrial robot. This helps workers, with no knowledge of typical programming languages, in teaching different activities to the robot. A neural network is trained to recognize hand gestures using features extracted from the accelerometer output of the Wii-controller. In [49], the authors train artificial neural networks to classify 25 static and 10 dynamic gestures to control an industrial 5 degrees-of-freedom robotic arm. A data glove, CyberGlove II, and a magnetic tracker, Polhemus Liberty, are used to extract a total of 26 degrees-of-freedom.

The authors of [7] present a study for measuring trust of human coworkers in fence-less human-robot collaboration for industrial robotic applications. To ensure safety of the human coworkers, it is essential to equip the robot with vision sensors, so that it can understand the environment and adapt to worker's behavior. The paper also discusses the use of RGB-D cameras to detect pointing gestures and proximity monitoring for safety using the depth information. In [12], authors use human gesticulations to navigate a wheeled robot through pointing gestures directed on floor. The interaction scheme also includes detection of facial gestures which often fails, as stated by the authors, because the untrained users make those gestures subtly.

In [50], the authors propose object recognition through 3D gestures using Kinect V1. They exploit the depth information from Kinect V1 to subtract the object background. This strategy often fails if predefined environmental assumptions are not met. Moreover, a histogram matching algorithm is used to recognize the objects placed on a white color table. Such techniques have recently been outperformed by modern deep learning ones like convolutional neural networks [51]. The authors of [52] propose a human-robot interaction system for the navigation of a mobile robot using Kinect V1. The point cloud acquired from Kinect V1 is fit on a skeleton topology with multiple nodes, to extract the human operator pose. This technique is not reliable to obtain the skeletal pose unless the human body non-linear anatomical constraints are modeled in the design of the skeleton topology.

According to [53], sign language is among the most structured set of gestures. Hence, in our work, we proposed the use of American Sign Language (ASL) for communicating with the robot. In the following section, we discuss previous works on sign language detection.

2.3 Sign Language Detection

Hand gesture detection techniques can be mainly divided into two categories: electronic/glove-based systems and vision-based methods. Some researchers prefer the use of wearable sensors to deal with occlusions or varying light conditions [54]. These sensors are expensive, counterintuitive and limit the operator's dexterity in his/her routine tasks. The vision-based methods can be further divided into two categories; a) methods that use markers and b) pure vision-based methods [55]. Since pure vision-based methods do not require the users to wear any data-gloves or markers, they offer ease-of-use for the operators to interact with the robots/machines. Furthermore, in Sect. 2.1 we have highlighted the advantage of using vision for safety monitoring. In this thesis, we therefore opt for a pure vision-based method and review only works on vision-based sign language detection.

Early research on purely vision-based methods for ASL recognition dates back to 1992 [56]. In this work, the authors use motion detection to capture start/stop instances of the sign/gesture, hand location tracking to record trajectory of the gesture, trajectory shape (using curve eccentricity) and detection of hand shapes at each stop position. The hand shapes are classified using the Hough Transform method described in [57]. The authors in [55] utilize a similar method as in [56]. It consists of a Canny filter that detects the hand edges in the scene, followed by a Hough Transform that extracts the feature vector of size 200. A neural network is then devised to classify hand gestures. The dataset used to train the neural network is extremely small and it is assumed that the image background is uniform. The authors do not mention the hands' localization in the scene during the recognition phase. Thus, it is assumed that the system only works if the hand appears in a specific region of the image.

One of the initial works in detecting ASL gestures through Hidden Markov models is discussed in [53]. The authors propose two settings in this research i.e., the second person view (desk based recognizer) and the first person view (wearable based recognizer). The proposed system recognizes sentences of the form "personal pronoun, verb, noun and adjective, pronoun" generated through 40 randomly chosen words. In both systems, videos were recorded and analyzed offline for ASL translation. An a priori model of the skin color is used to segment hands in the scene, while the absolute positions of the detected blobs in the image are used to distinguish left and right hand. The use of absolute positions of the hands in addition to a cumbersome wearable camera and computer system on the head significantly constrain the movement of the "signer" in the scene.

Recently in [58], researchers proposed an ASL translation system using binary hand images by keeping the edge information in the image intact. They use an image cross-correlation method to identify the signs by comparison with gesture images in a database. A similar hand gesture detector based on binary images is proposed in [59]. The author proposes

a color-independent (using preprocessed binary hand images) hand gesture detector that relies on a convolutional neural network (CNN), inspired by LeNet [60]. The classification accuracy of such system depends largely on the preprocessing steps of image segmentation performed with color or intensity thresholding, while CNNs are inherently able of learning color features robustly, as shown in [61]. Such systems also normally require a plain or white background for hand segmentation, which is hard to obtain in realistic human-robot interaction scenarios.

The use of depth cameras is becoming increasingly popular in applications like hand gestures detection or sign language translation. A thorough survey on 3D hand gesture recognition is presented in [62]. The depth information from such sensors can be used to segment the hands in cluttered backgrounds, by setting a depth threshold, while the normalized depth image of the hand adds the information for correct classification of the hand gestures [63]. The accuracy of such techniques depends on range and resolution of the depth sensors. Nevertheless, the use of depth sensors is beneficial, since it aids the detection of fine-grained hand gestures [64]. Latest works in deep learning have allowed the extraction of 2D hand skeletons from conventional RGB images [65, 66]. This can be used as a basis to fit a 3D kinematic hand model through an appropriate optimization technique as described in [67], thus eliminating the need of depth sensors for this purpose.

2.4 Dynamic Gestures Detection

2.4.1 Traditional Strategies

Traditional activity/dynamic gestures detection approaches aggregate local spatio-temporal information using hand crafted features. These visual representations include Harris3D detector [68], the Cuboid detector [69], dense sampling of video blocks [70], dense trajectories [71] and improved trajectories [72]. Visual representations obtained through optical flow like Histogram of Oriented Gradients (HOG), Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) have shown excellent results for video classification on a variety of datasets [70, 73]. In these approaches, global descriptors of the videos are obtained through encoding the hand crafted features using bag of words (BoW) and Fischer vector encodings [74], which typically assigns descriptors to one or several nearest elements in a vocabulary [75], while classification is performed through support vector machines (SVM).

2.4.2 3D Convolutional Neural Networks

The considerable success of deep neural networks on image classification tasks [76–78] instigated exploitation of the same in activity detection domain. The strategy to adapt CNNs for 3D volumes (3D-CNNs) obtained through stacking a few video frames to learn spatio-temporal features presented in [79] for action recognition is among the pioneering works in its category. The authors in [80] proposed a significant approach of learning the evolution of temporal information through Long Short-Term Memory (LSTM) recurrent neural networks [81] from features extracted through 3D-CNNs applied on short video clips of approximately 9 successive frames. The authors in [82] presented several CNN architectures operating on individual frames as well as with stacked-frames input. It was found that the stacked-frames architectures performed similar to the single-frame model. This demonstrated that the learned spatio-temporal features from short video clips were local thus unable to extract motion information well in the full video sequences. Thus, stacking frames i.e., 3D-CNN approach per se was unable to extract motion information well in the full video sequences.

2.4.3 Multi-Modal Approaches

The authors in [83] proposed a multi-modal gesture recognition strategy utilizing information from raw video data, articulated pose and audio stream. The proposed model function at two spatial scales to represent large-scale upper-body motions plus subtle hand articulation, and at two temporal scales to describe short momentary motions and longer sequences (gestures) as well. The hand images are cropped from the depth frames keeping the size of hands approximately constant through the estimation of bounding boxes size which is normalized by distance between the sensor and hands. No differentiation among the left and right hands is performed, instead the left hand images are mirrored horizontally to eliminate the differences in hand orientation. A four-layered CNN architecture is proposed for each hand to encode short spatio-temporal blocks. The first pair of layers perform 3D-convolutions followed by spatio-temporal max-pooling operations while the second pair of layers execute 2D-convolutions and spatial max-pooling. This is followed by a multi-layered perceptron (MLP) with softmax activation to classify $N+1$ gestures. The upper-body skeleton data is augmented to form a 139-dimensional pose descriptor. The pose descriptors are then stacked to represent a feature vector for the corresponding momentary motion. This is then fed into a fully-connected MLP with sigmoid units to classify $N+1$ gestures with softmax units in the last layer. Similarly, the output of an automated speech recognition module for the audio stream is arranged in the form of bag-of-words, formulating the appearance frequencies of

$N+1$ classes. This multi-modal output is then fused through a RNN which is trained in two stages to compensate for vanishing gradient problems associated with RNNs.

The extension of this work is presented in [84] with the proposal of a multi-modal multi-scale detection of *dynamic poses* of varying temporal scales. The employed modalities include intensity and depth videos, plus articulated pose information obtained through depth map. The authors termed a 3D volume of stacked video frames, synchronized across modalities, at a given temporal scale/step s as a *dynamic pose*. Three different values of s (2, 3 and 4) are chosen to capture multi-scale temporal evolution of the information to compensate for different tempos and styles of articulation by the users. A progressive learning method is proposed that includes pre-training of individual classifiers on separate channels and iterative fusion of all modalities on shared hidden and output layers. Moreover, a binary classifier is trained for gesture localization. This approach won the *Chalearn 2014 Looking at People Challenge (track 3)* [85] that involved recognizing 20 categories from Italian conversational gestures performed by different people and recorded with a RGB-D sensor.

2.4.4 Optical-Flow Based Deep Methods

The authors in [86], proposed a convolutional network based activity detection scheme along the same lines of [82]. The authors presented the idea of decoupling spatial and temporal nets. The proposed architecture in [86] is related to two-stream hypothesis of the human visual cortex [87]. The spatial nets corresponds to the ventral stream, which performs object recognition and the temporal nets are homologous to the dorsal stream, which detects motion. The spatial stream in this work operates on individual video frames while input to the temporal stream is formed by stacking optical flow displacement fields between multiple consecutive frames. In [88], authors presented improved results in action recognition task by employing trajectory-pooled two-stream CNN inspired by [86]. The authors exploited the concept of improved trajectories [72] as low level trajectory extractor. This allows characterization of the background motion in two consecutive frames through the estimation of homography matrix by taking camera motion into account.

2.4.5 CNN-LSTM Strategies

The authors in [89] proposed aggregation of frame-level CNN activations through 1) Feature-pooling method and 2) LSTM neural network for longer sequences. The authors argued that predictions on individual frames of video sequences or on shorter clips as in [82] may only contain local information of the video description and may confuse classes if there are

fine-grained distinctions. Explicit motion information in terms of optical-flow images is also incorporated in this method to compensate for 1 fps processing rate.

The authors in [90] proposed a Long-term Recurrent Convolutional Network (LRCN) for three situations i.e., 1) Sequential input and static output for cases like activity recognition 2) Static input and sequential output for applications like image captioning and 3) Sequential input and output for video description purposes. The visual features from RGB images are extracted through a deep CNN, which are then fed into stacked LSTM in distinctive configurations corresponding to the task at hand. The parameters are learned in an “end-to-end" fashion, such that the visual features which are relevant to the sequential classification problem are extracted.

The authors in [91] proposed a CNN-LSTM for skeleton-based human action recognition. Instead of applying convolutional operation on raw images, the input is arranged in a three-dimensional volume of skeletal information i.e., (*x, y, z*) of each joint, for a fixed number of consecutive frames. The CNN is first attached to a two-layer MLP and pre-trained which then is replaced by a LSTM network for learning temporal features for activity recognition.

2.4.6 Attention Based Methods

The application of convolutional operations on entire input images tends to be computationally complex and expensive. The authors in [92] discussed the idea of *visual representation* which implies that humans does not form detailed depiction of all objects in the scene instead, the human perception is focused selectively on the objects needed immediately. This additionally is supported by the concept of *visual attention* presented in [93] which is later improvised for deep methods as in [19]. The authors proposed an idea of a *glimpse sensor* which extracts a retina-like representation with high-resolution region at a certain location progressively surrounded by lower dimensional description of the original image. This model is built around a recurrent neural network which aggregates the information over time to predict a new sensor focus location. The parameters of *glimpse sensor* are learned as a reinforcement learning problem.

2.4.7 Pose-Driven Attention Mechanisms

A spatio-temporal attention mechanism conditioned on pose is proposed in [94]. A topological ordering of skeletal joints is defined as a non-Hamiltonion connected cyclic path. This property of the path ensures that the neighborhood relationships are preserved. The articulated pose data is encoded into 3D tensors over time by concatenating pose vectors. A convolutional network then learns pose feature representations from these 3D tensors

as it extract relationships along its depth between joint coordinates, neighboring joints, features which are further away from the human body and between poses corresponding to two different people. A spatial-attention mechanism inspired by [19] is also proposed except that it is pose-conditioned over 4 attention points i.e., hand locations of two people in the scene. A spatial attention distribution is learned conjointly through the hidden state of the LSTM network which is responsible of learning the temporal evolution of information, and the learned pose feature representations. This spatial attention distribution determines dynamically, through learned weights, which hand crop deserves more attention for gesture recognition. The same spatial attention distribution is stacked for a sub-sequence combined with learned feature representation from the hidden states of LSTM, and is exploited to formulate an adaptive temporal pooling mechanism. This operation assigns higher weights to the more discriminative hidden states, provided that it has seen full sequence before the prediction is offered. The proposed architecture is trained on NTU RGB+D dataset [95] and the knowledge transfer is demonstrated to MSR Daily Activity 3D dataset [96].

The authors in [17] present a slightly different approach to their previous work in [94]. Instead of learning spatial attention distribution through the hidden states of a LSTM network and pose feature representations, it is learned through an *augmented pose vector*, which is defined by the concatenation of current pose, velocity and accelerations for each joint over time. The LSTM network exploited in the previous work is replaced by a *Gated Recurrent Unit* (GRU), originally introduced in [97], as a recurrent function. The authors propose a statistic named *augmented motion*, which is obtained by the sum of absolute velocity and accelerations of all body joints at the current time step to perform temporal pooling on hidden states at the end of a sequence.

2.4.8 Multi-Label Video Classification

A multi-label action recognition scheme is presented in [98]. The authors extended the labels of THUMOS action recognition dataset [99] to fine-grained dense multi-label annotations and named it Multi-THUMOS. A novel Multi-LSTM network is proposed to tackle multiple inputs and outputs. The authors fine-tuned VGG-16 CNN which is already trained on ImageNet [76], on Multi-THUMOS dataset on an individual frame level. A fixed length window of 4096-dimensional fc7 features of the fine-tuned VGG-16, is passed as input to the LSTM through an attention mechanism that weights the contribution of individual frames in the window. The final class(es) labels are obtained with a weighted average of multiple outputs through learned weights. The network is also trained with shifted labels to predict actions in the videos.

2.4.9 Chalearn 2016 Gesture Recognition Challenge Strategies

In this thesis, we present the state-of-the-art results on *Chalearn 2016 Looking at People isolated gestures recognition* dataset, which was released for "large-scale" learning and "user independent" gesture recognition from RGB or RGB-D videos. The details of this dataset will be discussed in Section 4.2.1. Here we briefly explore the literature on strategies employed in reported state of the art for this dataset.

In [100] authors proposed a multi-modal large-scale gesture recognition scheme on *Chalearn 2016 Looking at People isolated gestures recognition* dataset [26]. ResC3D network [101] is exploited for feature extraction, while late fusion strategy is opted for combining features from multi-modal inputs in terms of canonical correlation analysis. The authors used linear SVM to classify final gestures. ResC3D network allows to exploit spatial features extraction strength of ResNet [77] with temporal features extraction capability of 3D-CNNs. A *key frame attention mechanism* is also proposed for weighted frame unification which exploits movement intensity in the form of optical flow, as an indicator for frame selection. Retinex filter [102] is applied on RGB images to normalize illumination changes while median filter is applied on depth images for noise reduction. This team (ASU) obtained first place in the second phase of the challenge.

The team (SYSU-ISEE) [1] that obtained second place, learned discriminative motion features through RGB-D videos, optical flow sequences and skeleton. The skeleton is obtained via Regional Multi-person Pose Estimation algorithm [103]. The team employed LSTM network to learn the temporal evolution of skeleton information. Rank pooling algorithm [104] is applied on optical flow and depth frames to extract static cues, which are then passed through VGG-16 network independently. The output of VGG-16 and LSTM is combined through late-fusion strategy. The team (Lostoy) [1] that obtained third place proposed a masked 3D-CNN on portions of RGB-D images occupied by hand bounding boxes obtained through a pose-estimation method.

Lately authors in [105] present the state-of-the-art results on *Chalearn 2016 Looking at People isolated gestures recognition* dataset. A novel multi-channel architecture namely FOANet, built upon spatial *focus of attention (FOA)* concept is proposed. Inspired by [106], the authors crop region of interest occupied by hands in the RGB and depth images through region proposal network and Faster R-CNN method as described in [107]. To distinguish left and right hands, *openpose* skeleton extractor presented in [66, 65] is utilized. The proposed architecture comprises of 12 channels in total with 1 global (full-sized image) and 2 focus (left and right hand crops) channels for 4 modalities (RGB, depth and optical flow fields extracted from RGB and depth images) each. The softmax scores of each modality is fused through the proposed sparse fusion network.

In this thesis, we propose a spatial attention-based CNN-LSTM architecture which models the evolution of spatial and temporal information in video sequences through augmented pose and hand images. The proposed strategy is described in Chapter 4 while we present our detailed contributions in Section 4.1.

2.5 3D Human Pose Estimation

Current 3D human pose estimation approaches can be roughly categorized into buttom-up discriminative, top-down generative and hybrid methods [108–110]. Discriminative strategies [111–115] directly regress 3D pose from image data. Generative ones [116–122] search for a plausible body configuration in the pose space that matches the image data, while hybrid methods [110, 123–125] make the most out of both approaches.

3D human pose estimation methods can also be categorized into two groups; one-stage approaches [114, 126, 127] which directly regress 3D pose from images and two-stage methods [122, 128–130] that first estimate 2D pose in the form of joint location confidence maps and then lift this 2D prediction to 3D pose either by a constraint deep regression strategy [131, 132] or by matching the predictions with 2D projections of existing 3D poses from a database [129] or by fitting a 3D model on this 2D prediction [122, 128]. With the availability of larger datasets [126], the state-of-the-art approaches rely on deep networks [115, 133]. However, the basic work-flow as mentioned here, largely remains unchanged.

2.5.1 Discriminative Approaches

As mentioned above, discriminative methods tend to predict 3D pose directly from image data, which can either be monocular images [113, 134, 114, 135, 127, 136], depth images [112, 125, 137] or short image sequences [115]. Earlier works on human pose estimation from a single image relied on discriminatively trained models to learn a direct mapping from image features such as silhouettes, HOG, or SIFT, to 3D human poses [111, 138–143]. Recent methods [114, 136] tend to exploit deep architectures for 3D pose estimation relying on pre-trained reliable 2D pose extractors. Lately in [144], the authors exploit a convolutional neural network to yield 2D and 3D joint locations simultaneously and fit a model-based kinematic skeleton against these predictions. The authors in [145] propose an end-to-end architecture for direct regression of multi-person joint 2D and 3D pose in RGB images. The ground truth full-body 3D poses are estimated through a data driven retrieval method from a motion capture dataset given 2D pose as input.

Discriminative approaches often include time consuming offline training phase relying on large annotated datasets to learn a mapping from image input to 2D/3D human pose output. Nonetheless, these approaches are computationally efficient during run-time, perform single frame pose estimation and do not require initialization [109, 110, 146].

2.5.2 Generative Approaches

Generative methods use a 3D human model and find a 3D pose iteratively by estimating its position, orientation and joint angles that brings appearance of the model or its projection in concordance to the image input [108–110]. In early works, this was achieved through optimization of an energy function through hand-crafted features extracted from the input images such as silhouettes [119, 147, 116, 148, 149], trajectories [150] and manually extracted features [120, 151].

Generative methods often provide physically plausible solutions with high accuracy, and do not require training. However, because of online feature generation and comparison to the input observations, they are computationally expensive during run-time, suffer from drift and track loss plus require initialization [109, 110, 146].

2.5.3 Hybrid Approaches

Estimation of 2D joint locations in an image is easier than directly predicting 3D human pose [108]. Moreover, with the advent of deep networks accompanied by large datasets, reliable 2D pose estimators have been released to the community [65, 66, 152, 153]. Moreover, discriminative 3D pose detectors are often less accurate while the process of 3D pose tracking in the generative methods normally require manual initialization [154]. This approach is also often prone to local minima because it initializes the current pose with the previously detected pose. This lays the foundation of hybrid methods which combines the discriminative 2D pose regressors (or rough frame-by-frame 3D human pose through RGB-D sensors) with refined 3D pose estimation and model tracking approach of generative methods.

The authors in [154] introduce a hybrid motion capture scheme that automatically combines a discriminative 3D pose detector of low accuracy with a generative 3D pose tracker that provides refined 3D pose estimations. The authors in [125] exploit nearest neighbor matching strategy to construct 3D human skeleton models from depth imagery. In [122], the authors utilize 2D joint proposals obtained through a convolutional network and fit a statistical 3D body model to recover the entire shape of the human body. Nevertheless, this approach is not real-time as the optimization process requires 20-60 seconds per image. The authors in [155] adopt a non-parametric approach and utilize the predicted 2D pose to look

up the nearest 3D pose in a motion capture dataset. In [110], the authors employ openpose [65, 66] and depth information from a RGB-D sensor to extract a pseudo-3D human skeleton, as also performed by [22], and propose an energy optimization framework based on *particle swarm optimization* [156] with 2D and 3D joint hypothesis for accurate 3D human pose estimation.

2.6 Conclusion

Lately, the tremendous success of deep neural networks on image classification tasks [76, 77, 61, 78] instigated exploitation of the same in the domain of activity recognition. Hence, in our work on static hand gestures detection detailed in Chapter 3, we chose to exploit convolutional neural networks for gestures classification. Contrary to previous methods which employ CNN on binary or hard-coded background subtracted images for hand gestures recognition like [59, 50], we propose the strategy of color hand images background substitution with random pattern and indoor architecture images to increase data variance, which allows the network to learn robust hand features independent of the backgrounds. Thus, our CNN has learned to concentrate on image pixels occupied exclusively by hands which enables it to distinguish subtle hand movements accurately. Thus permit real-time hand gestures detection by avoiding time-consuming rigid image processing methods during the recognition phase. This work is detailed in Chapter 3 of the thesis.

Subsequently, in the context of dynamic gestures recognition which we present in Chapter 4, the work of Karpathy *et al.* [82] established that the stacked-frames architectures performed similar to the single-frame model. This demonstrated that the learned spatio-temporal features from short video clips were local thus unable to extract motion information well in the full video sequences. Thus, stacking frames i.e., 3D-CNN approach per se was unable to extract motion information well in the full video sequences. Moreover, the proposed strategies in [83, 84] are similar in function to [82] except that they included depth images, speech and pose as additional modalities. However, it lacked a dedicated equipment to learn evolution of temporal information and may fail in cases where understanding of long-term dependencies for the performed gestures may be required. Also, the performance of these strategies is untested on large-scale gestures datasets. Optical flow-based methods including the *key frame attention mechanism* proposed in [100] may help to emphasize on frames where motion is detected, they are unable to differentiate between motion caused by the objects in the background or irrelevant objects in the scene. Thus, in our work, we developed a novel strategy to model human-centered spatio-temporal dependencies for the recognition of dynamic gestures. We employ CNN to extract spatial features from the input

frames while LSTM is utilized to learn the evolution of information in time. Our *spatial attention mechanism* localizes and crops hand images of the person which are subsequently passed as inputs to our CNN networks unlike previous techniques [90, 89] where entire image frames are exploited as input. Contrary to [98], where a pre-trained state-of-the-art network is fine-tuned on entire image frames of gestures dataset, we fine-tune *Inception V3* on background substituted hand gestures dataset to be used as our CNN block as already mentioned above. We extract image-embeddings from the last fully connected (FC) layer of our fine-tuned *Inception V3* of size 1024 elements which is exploited as an input modality in our dynamic gestures detector. Moreover, previous strategies for dynamic gestures recognition/video analysis [17, 94, 83, 84] employed 3D human skeleton to learn large-scale body motion thus required specialized sensor modalities, we on the contrary, only utilize 2D upper-body skeleton as an additional modality to RGB hand images in our algorithm. Nevertheless, scale information about the subjects is lost in monocular images. Thus, we also propose novel *learning-based depth estimators* which determine the approximate depth of the person from the camera and region-of-interest around his/her hands from upper-body 2D skeleton coordinates only. To reiterate, the inputs to our dynamic gestures detector are limited only to color hand images and an augmented pose vector obtained from 8 upper-body 2D skeleton coordinates, unlike other existing approaches like [105], which includes full-frame images in addition to hand images, depth frames and even optical flow frames altogether in their algorithms. Thus our proposed strategy is generic and straightforward to implement in mobile systems.

Finally in this thesis, we present a novel strategy to estimate 3D human pose from only monocular images which is detailed in Chapter 5. The proposed strategy is actually a hybrid method for 3D human skeleton estimation as an optimization problem which minimizes the discrepancy between 2D skeletal joint coordinates obtained from a discriminative 2D pose extractor, and virtual 2D camera projection of a 3D kinematic model of a human-body.

Chapter 3

Static Hand Gestures Detection

We developed an elementary framework for static hand gestures in [22] which presented a tool handover task between robot and human coworker through static hand gestures. A convolutional neural network, inspired mainly by LeNet [60] was developed, to classify four hand gestures. However, the dataset exploited in [22] was small, and the hand images were recorded only by one individual. This could not guarantee correct detection of hand gestures made by other individuals and with backgrounds having rich textures.

In this chapter, the extension of [22] is detailed. A hand gesture detector is trained on ten gestures instead of only four utilized earlier. Moreover, the backgrounds are replaced with random pattern/indoor-architecture images to make the detection robust and background invariant. The vision pipeline is then integrated with *OpenPHRI* [25] to complement the library with two collaborative modes of the ISO/TS 15066 safety standards. This integration is detailed in Section 3.4 of this chapter. We propose an interaction setting where a human coworker can safely instruct commands to the robot via gestures.

3.1 Our Contributions

- Development of a real-time hand gesture detection framework which localizes hands through asynchronous integration with a discriminative 2D skeleton detector and classifies hand gestures at frame-rate of approximately 20fps.
- Integration of Kinect V2 depth map with the obtained 2D skeleton to get a pseudo 3D skeleton, which is used for *speed and separation monitoring* to ensure the safety of the human coworker.
- Training a background-invariant hand gestures detection system through transfer learning from Inception V3 convolutional neural network.

- On-line release of hand gestures database of Kinect V2 recordings for benchmarking and comparison.
- Integration of the developed hand gesture detection module with safe physical human robot interaction framework, namely *OpenPHRI*.
- Validation of the proposed framework for robot teaching and control of Kuka LWR 4+ arm with the detected hand-gestures.

The overall vision pipeline of the proposed framework is illustrated in Fig. 3.1. Each named box is a cyclic process and dotted arrows represent asynchronous communications between these processes. Each process is described in the following sections in detail.

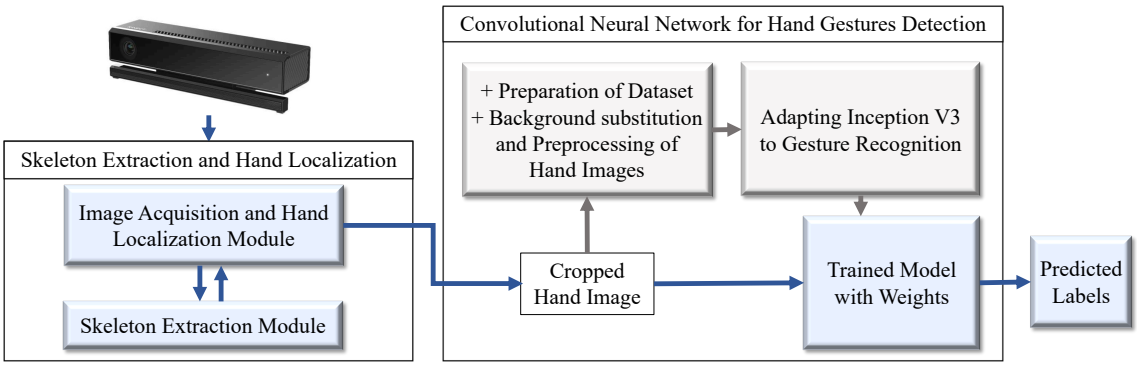


Figure 3.1 The overall vision pipeline of our static hand gestures detector. Grey lines represent one-time pathway for dataset preparation and convolutional neural network training. The route represented by blue lines is complied by our static hand gestures detector execution program.

3.2 Skeleton Extraction and Hand Localization

For safe Human-Robot Interaction, it is essential for the robot to understand its environment, particularly the human coworker. In this research, we opted for Microsoft Kinect V2 as the main sensor to capture the visual information of the human coworker. Kinect V2 is a time-of-flight sensor and provides a larger field-of-view and higher resolution RGB and depth images than its predecessor Kinect V1. This allows the robot to extract functional information from the scene, like human(s) presence or object/obstacle detection, including depth perception.

3.2.1 Skeleton Extraction Module

In this work, we utilize *openpose* [66, 65], to extract skeletal joint coordinates, as in [157, 158]. This library returns 2D skeletal coordinates (x_i, y_i, c_i), for $i = 1, ..., 18$, from a RGB image, using confidence maps and parts affinity fields in a multi-person scene; x_i and y_i are the abscissas and ordinates respectively of 18 COCO body parts [159], while c_i represent their confidence measure. *Openpose* works on the principle of *convolutional pose machines* described in [65].

Openpose is a discriminative 2D pose extractor and is not trained on pre-defined body poses. Hence, it is preferred over libraries like *OpenNI* and *Microsoft SDK* as they are often not accurate in skeleton extraction, require initialization pose and constraint the user to face the sensor. For real-time skeleton extraction, this method requires a multi-GPU hardware with the output frame-rate mainly dependent on the number of persons in image. The average frame rate that is achievable with two *Nvidia GeForce GTX 1080* for Kinect V2 is approximately 14 fps. Since we employ only one GPU in our framework, we obtain 6 fps with 1 person in the scene.

3.2.2 Image Acquisition and Hand Localization Module

The strategy to localize human body and its sub-parts (i.e., hands or face) depends mainly on the output of utilized sensor. In [160] the authors use skin color for hand segmentation using a conventional RGB camera, as in [59]. Human body localization is performed using laser sensors in [161], and its sub-parts are obtained through Kinect with the *OpenNI* library as in [162]. In [52], the authors localize human body, inspired by [163], through merging clusters of the point cloud obtained from the Kinect V1 after voxel filtering and ground plane removal. Lately, infrared based sensors e.g., Leap Motion, are developed to track fingers of a hand in the near proximity (within 25 to 600 millimeters) of the sensor. However this range is too close for our application. In [67], authors adapt a state-of-the-art object detection deep learning technique namely *YOLO V2*, adapted to localize hands and head/face of a person in a scene. The authors have utilized *openpose* to first extract hands and face images from recorded videos with human activity, and then used these images to train *YOLO V2* to detect hands and the face of the person in the scene in real-time. The face is detected to differentiate left hand from the right one. This is an efficient method to detect hands in the scene in real-time but requires a separate training/adaptation of *YOLO V2* for hands and faces.

In our research, since we obtain the skeletal joint coordinates from *openpose*, training a separate hand detector to localize hands is not required. To estimate the hands position, we fit a line between the elbow joint and the wrist joint returned by *openpose* (with added depth

information from Kinect V2) and extend this line to one-third of its original length (empirical value) in the direction of hand to approximately reach the center of hand. A bounding box is then centered at the approximated hand center at an angle which the forearm makes with the horizontal. This makes the hand image acquisition rotation invariant. The size of each bounding box is determined by the mean depth value of a 6×6 matrix centered at the wrist joint, obtained through Kinect V2 depth map as shown in Fig. 3.2. The hand images are cropped with reference to the tilted bounding box, re-scaled to size 224×224 pixels and rotated again such that the cropped image becomes vertical.

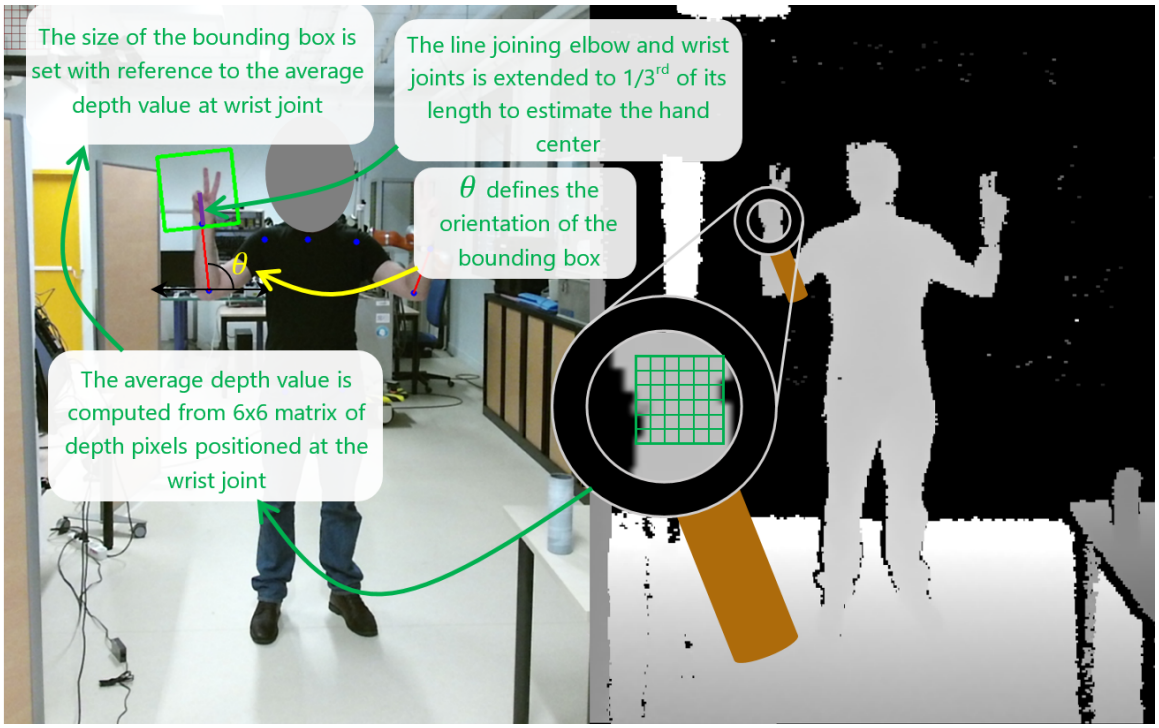


Figure 3.2 Localization of hand through *openpose* is illustrated. The bounding box is titled with an angle that the forearm makes with horizontal, while the size of bounding box is determined by the mean depth value of the wrist joint. The mean depth value is computed by averaging the depth pixel values of a 6×6 matrix centered at the wrist joint.

3.2.3 Asynchronous Integration of the Modules

In our work published in [22], we integrated *openpose* with gesture recognition sequentially to obtain an overall temporal resolution of approximately 4 fps. Here we propose an inter-process distributed system, designed through *nanomsg* socket library[1] which has drastically

[1]https://nanomsg.org

increased the frame rate of our vision pipeline. The afore-mentioned inter-process distributed system works using a “request-reply" communication pattern, known as scalability protocol.

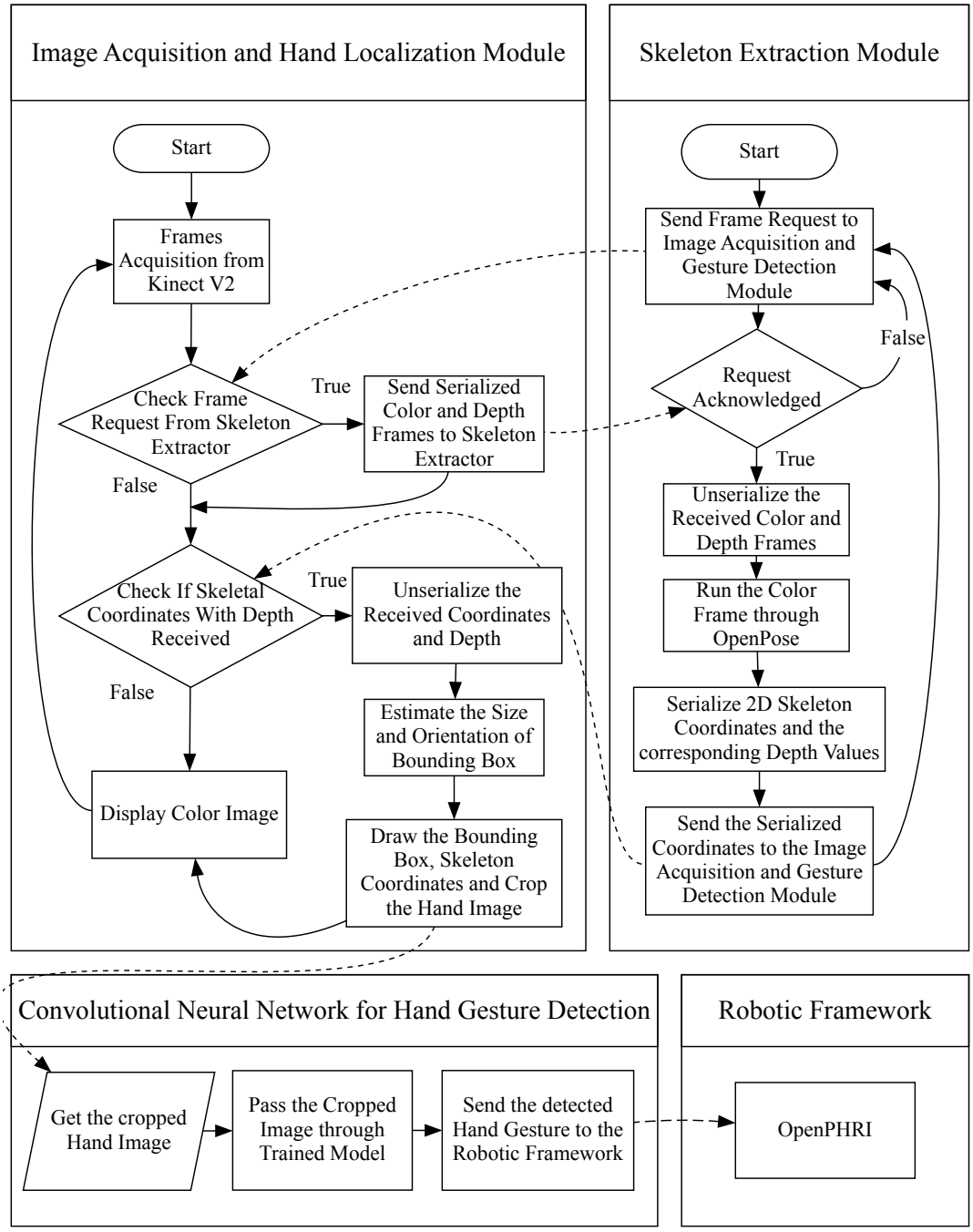


Figure 3.3 The overall pipeline of our asynchronous distributed network for pHRI using hand gestures

Furthermore, it ensures that no frames are lost during communication. Figure 3.3 illustrates this asynchronous communication between the proposed framework modules via dotted lines. The image acquisition and hand localization module retrieves the image stream from Kinect V2 and checks if a frame request has arrived from the skeleton extraction module. When a frame request is received, the current RGB and depth image are first serialized through flatbuffers[2] and then passed to the skeleton extraction module. The skeleton extraction module unserializes the received frames with flatbuffers and then pass the RGB image through the forward-pass of *openpose* which returns a vector of 2D skeleton coordinates (x_i, y_i, c_i). The calculated mean depth values, as described in the previous section, are concatenated with the 2D skeleton coordinates and this 3D vector (x_i, y_i, d_i) is then sent to the image acquisition and hand localization module. The integration of Kinect V2 depth map with the 2D skeleton coordinates from *openpose* however does not represent the actual 3D coordinates of the joints and represents only the surface depth value of the joints. There is a possibility that a joint is occluded in the scene by an object or the body itself. To prevent false detection of depth hence preventing potential accident, we use the confidence measure for each joint returned by *openpose*. The depth value of each joint is only updated if $c_i > 0.5$ (this is an empirical value), otherwise the previous depth value is kept. The image acquisition and hand localization module expects to receive coordinates from skeleton extraction module in each execution cycle. Once the coordinates are received, the hand is segmented and cropped image (described in Section 3.2.2) runs through the forward-pass of trained convolutional neural network for hand gesture detection. The detected hand gesture label is sent to the robot controller running *OpenPHRI* to pilot the experiment. The overall frame rate of our gesture detection pipeline is approximately 20 fps while the skeleton is extracted and the hand location is updated at around 5 fps. This significantly improves the execution performance of the vision system as compared to that in [22], which finally leads to a system which better reacts to human commands.

3.3 Convolutional Neural Network for Hand Gestures Detection

In [22], we designed the CNN architecture for hand images with relatively plain backgrounds, while the number of gestures were set to 4 and the gestures were recorded by a single person. Here we employ 10 static hand gestures recorded by 10 volunteers (8 males and 2 females) of age 22 to 35 and the backgrounds of hand images are substituted with random pattern

[2]https://google.github.io/flatbuffers.

and indoor architecture images (explained in Section 3.3.2). This makes the recognition problem more complex as compared to the one presented in [22], where only 1 volunteer and 4 gestures had been considered. Therefore we opted for transfer learning for gesture recognition, exploiting state-of-the-art CNNs pre-trained on large image data from the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [164]. In particular, Inception V3 [165] which is state-of-the-art in image classification for 1000 classes, is adapted for our background-independent hand-gesture recognition task. Inception V3 is available in Keras python library [166] with pre-trained weights. Figure. 3.4 shows samples of the static gestures we trained our framework on. The gestures include 9 letters/numbers taken from ASL [167] and a **None** gesture that is not among these nine. The letters/numbers are chosen such that they resemble with each other (like F, 7 and W; A, L and Y) so as to challenge the training and ensuring robustness of the CNN. The **None** gesture is important to determine if the person does not intend to interact with the robot. Our system also generates this label when the line joining the elbow and wrist joint (forearm) of the user is too low (in the lower two quadrants of the axes centered at the elbow joint). In this case, the robot controller ignores this command and does not initiate any action (which is likely undesired, since the user's hand is low). We could have excluded the **None** gesture from the trained network, by applying a predefined threshold to the nine (one per class) network scores. However, since the relative scores of the ten classes vary a lot according to the operating conditions, it is not possible to fix a priori such threshold.

Figure 3.4 Samples of the gestures considered for training. The labels represent the letters and the numbers taken from American Sign Language. The last gesture is one of the several **None** gestures included in the training set.

3.3.1 Preparation of Dataset/Dataset recordings

To create a dataset for gesture recognition and off-line development, RGB and depth image streams from Kinect V2 are saved in the local workstation. The frames are saved with an approximate frame rate of 20 fps. Each gesture is recorded by each volunteer for around 12 seconds with both hands (see Fig. 3.5), at three distances of 5, 3 and 1.5 meters away from Kinect V2. The depth information near Kinect V2 is rich and accurate, thus the images recorded at the distance of 1.5 meters are used for the fine-tuning of Inception V3 (discussed in Section 3.3.3). However, since the network is trained only on RGB images, the hand gestures can also be recognized at other distances. We are releasing our dataset *opensign* [3] online. *Opensign* contains RGB and depth (registered) frames of volunteers recording 10 gestures. The RGB images are saved in *png* format, while the float data of the depth images are saved in *bin* files. The total number of samples in our dataset is 20950. These include 8646 original images, and 12304 synthetic images obtained by substituting backgrounds with the technique that we will explain in Sect. 3.3.2.

Figure 3.5 A volunteer recording '7' gesture in the laboratory

We divide the dataset of 20950 images with a ratio of 3:1:1, i.e., 12570 train images and mutually exclusive 4190 samples for cross validation and test sets. Train images go through extensive pre-processing (explained in Section 3.3.2), while only selective pre-processing operations are applied to cross-validation images so as to keep them near those obtained during recognition in the robotic interaction experiments.

[3] http://bit.do/OpenSign

3.3.2 Background substitution and Preprocessing of the Hand Images

Background substitution is performed so the network is trained to detect hand gestures independently from the background. We use nearly 1100 images of random pattern and indoor architectures which are freely available on the internet[4]. The background substitution process is illustrated in Fig. 3.6. A binary mask for background substitution is created using the depth information from Kinect V2. All the pixels that lie at distance within ± 18 % (empirical value) of the mean depth value computed at the wrist joint (obtained through *openpose*) are set to 1, while the rest are zeroed. This binary mask is broadcasted into three channels and then multiplied by the cropped RGB hand image to get a background subtracted hand. An inverted mask is also created by simply applying a “NOT" operation on the mask originally computed. The background pattern images are cropped to multiple 224×224 sized images (as it is the set size of hand images) which are subsequently multiplied by the inverted hand mask. The hand image with subtracted background and the pattern images multiplied with the inverted binary mask are then added in the final step of background substitution.

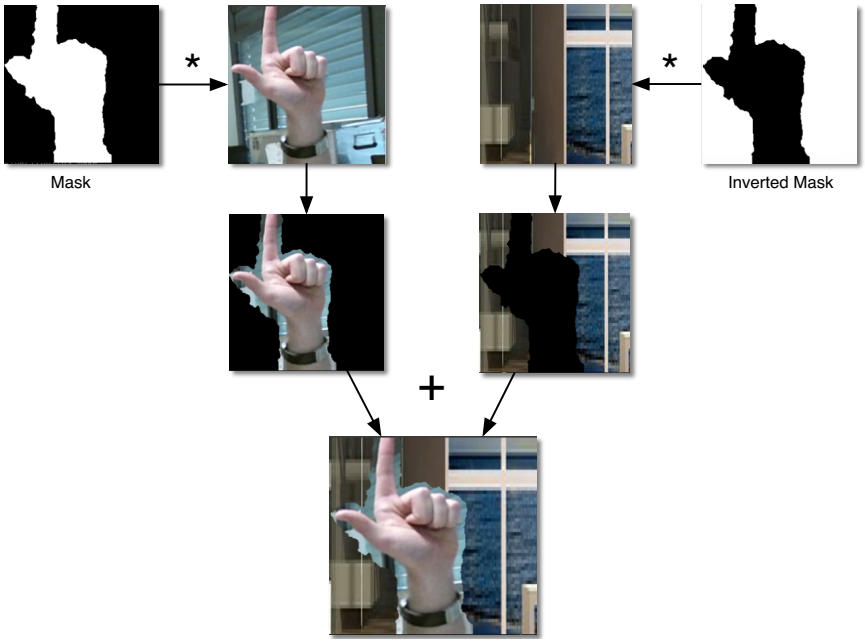


Figure 3.6 The process of background substitution.

Figure 3.7 shows the samples of gestures with original and substituted backgrounds. As discussed in Section 3.3.1, all the training images (images with substituted and original backgrounds) go through several pre-processing steps. Image processing operations of histogram equalization and introduction of Gaussian and salt and pepper noise are applied

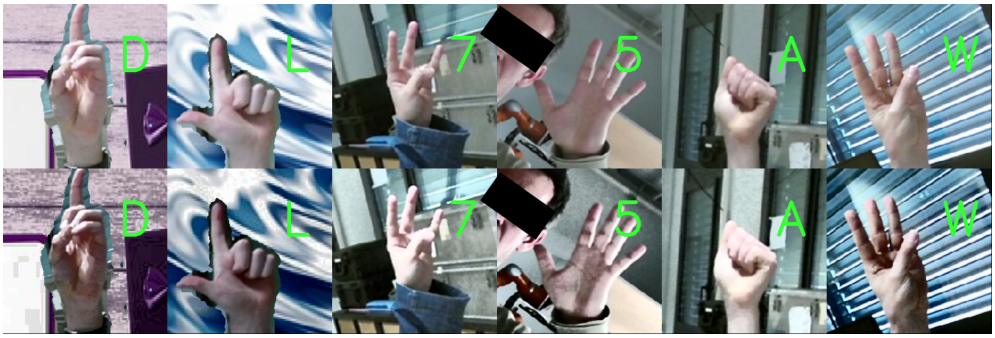
[4]https://pixabay.com/

on 30% of training images each while the remaining 10% are left unprocessed. Figure 3.8 shows random samples of original and processed images after the addition of Gaussian noise and histogram equalization.

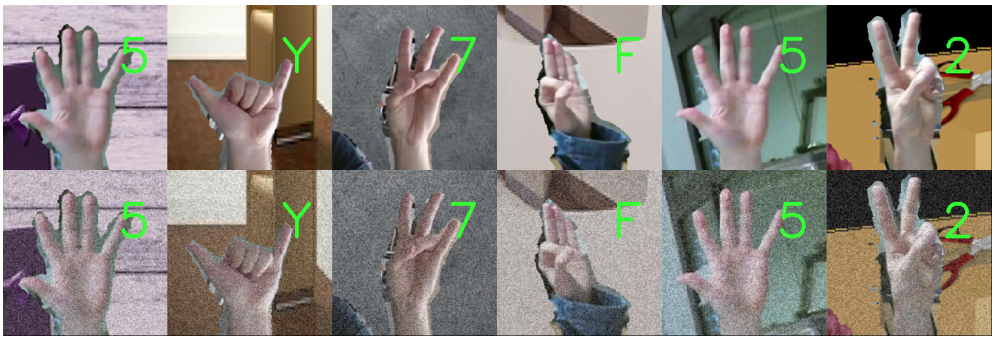
Figure 3.7 Samples of hand gesture images with original (labeled images) and substituted backgrounds (below originals). Note the remnants of the original backgrounds. This phenomenon is due to dilation of the binary masks. While it could be avoided by using techniques like chroma key, we do not intend to use a uniform background, to avoid bringing any extra apparatus in operation. In the experimental results (Sect. 3.3.4), we show that despite these remnants, gesture detection is highly accurate.

For robust gesture detection, we also use the real-time data augmentation feature of Keras library. Keras real-time data augmentation is designed to be iterated by the model fitting

process, creating augmented image data in defined batch size during training. This reduces the memory overhead of the computer but adds additional time cost during model training. The image processing operations that are applied on all training images (after the addition of noise and histogram equalization as discussed above) using the Keras library include channel shift, zoom, shearing, rotation, axes flip and position shift.



(a) Samples of training images after histogram equalization



(b) Samples of training images after the introduction of Gaussian noise

(c) Samples of training images after the introduction of salt and pepper noise.

Figure 3.8 Image processing operations of histogram equalization, introduction of Gaussian and salt and pepper noise are performed on the training images. First row in each sub-image shows unprocessed image while the processed images are shown in the second rows.

The batch size for model fitting is set to 100 samples. These transformations are applied in real-time during model training. So the number of train images remains the same while

each batch for training is applied with selected – yet randomly chosen – transforms. In Fig. 3.9, we show samples of processed training images with Keras being passed to the CNN.

3.3.3 Adapting Inception V3 to Gesture Recognition

In image classification problems, the input data i.e., an image, is formed by low-level edges, curves and color combinations irrespective of the type of object that the image represents. It is therefore assumed that the early layers in the pre-trained state-of-the-art networks have learned to efficiently extract those features from the images thus they need to be preserved.

Figure 3.9 Image processing operations applied to the training images include color-shift, zoom, shear, rotation, axes flip and position shift processes.

Inception V3 is trained to recognize 1000 classes of objects as explained in Section 3.3. To adapt Inception V3 to classify only 10 gestures, the last softmax activation layer of this network with 1000 neurons should be replaced with a new layer of 10 neurons. As implemented in Keras, the Inception V3 has 10 trainable inception blocks. We perform training in three phases. In the first phase all the layers (hence inception blocks) in the network are frozen with the exception of the new layer added and the CNN is trained for 10 epochs only. This fine-tune the weights of the new layer exploiting the knowledge of all pre-trained inception blocks. Then we unfreeze last two inception blocks and trained the CNN for 10 epochs, and then we trained top four inception blocks so the network is fine-tuned properly on our dataset. This gradual unfreezing of inception blocks prevents damaging the pre-trained weights and thus avert over-fitting.

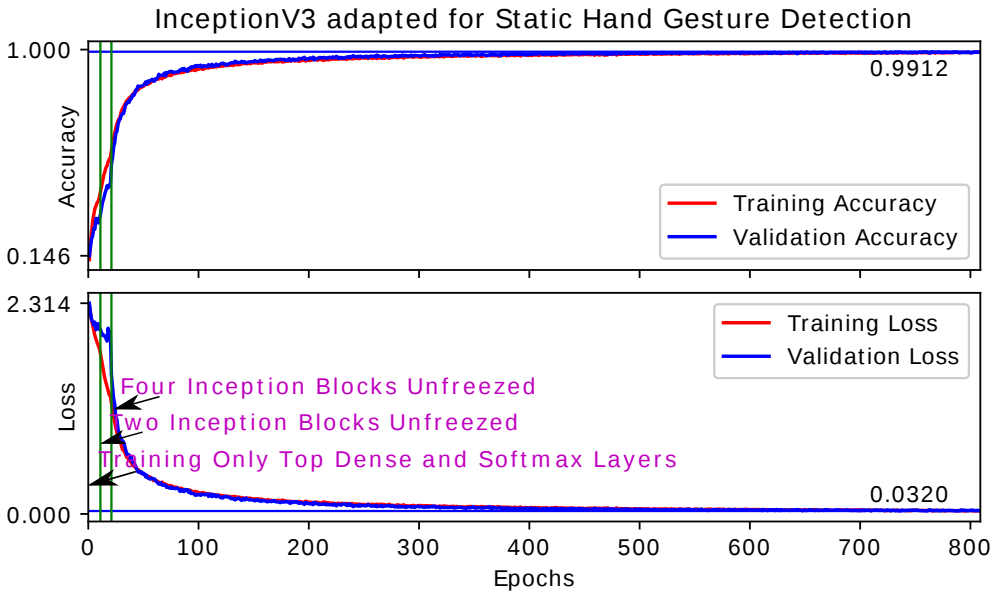


Figure 3.10 Plot of validation accuracy (top) and validation loss (bottom)

The validation set is used to chose the best performing weights and then the network is tested on the unseen test set to quantify/estimate accuracy of the selected weights. Figure 3.10 illustrates the training curve of validation accuracy and loss of our dataset. Each epoch took approximately 130 seconds to pass and the network was able to achieve validation accuracy of 99.12% at 745^{th} epoch taking around 27 hours of training.

3.3.4 Quantification of the Trained CNN

To validate and quantify the results even further, the accuracy of the trained CNN is tested with a test set of 4190 new images. The overall test accuracy of the trained CNN is found to be 98.9% on test set. The normalized confusion matrix in Fig. 3.11 shows the accuracy of each gestures and misinterpretation of one gesture against the others. It can be observed that despite 94.3% accuracy of the **None** gesture, it was misinterpreted the most among all. The reason for this lower accuracy is that the **None** gesture defines all gestures that do not appear like the other 9 as well as all transitional gestures.

It is difficult to include all the transitional gesture possible to be classified as **None** gesture. Moreover, it can be observed from a close inspection of the test results that the CNN is very accurate in identifying a gesture as **None** when a person is holding an object in his hand. It is inferred that if the CNN is additionally trained on a gesture like "an object in hand", this class label will be easily distinguished. Meanwhile, this misinterpretation can be

dealt by adding a software constraint, as explained in Section 3.3, of not invoking gesture detector until the arm is in the upper two quadrants of the axes centered at the elbow joint of the person, as we did in [22].

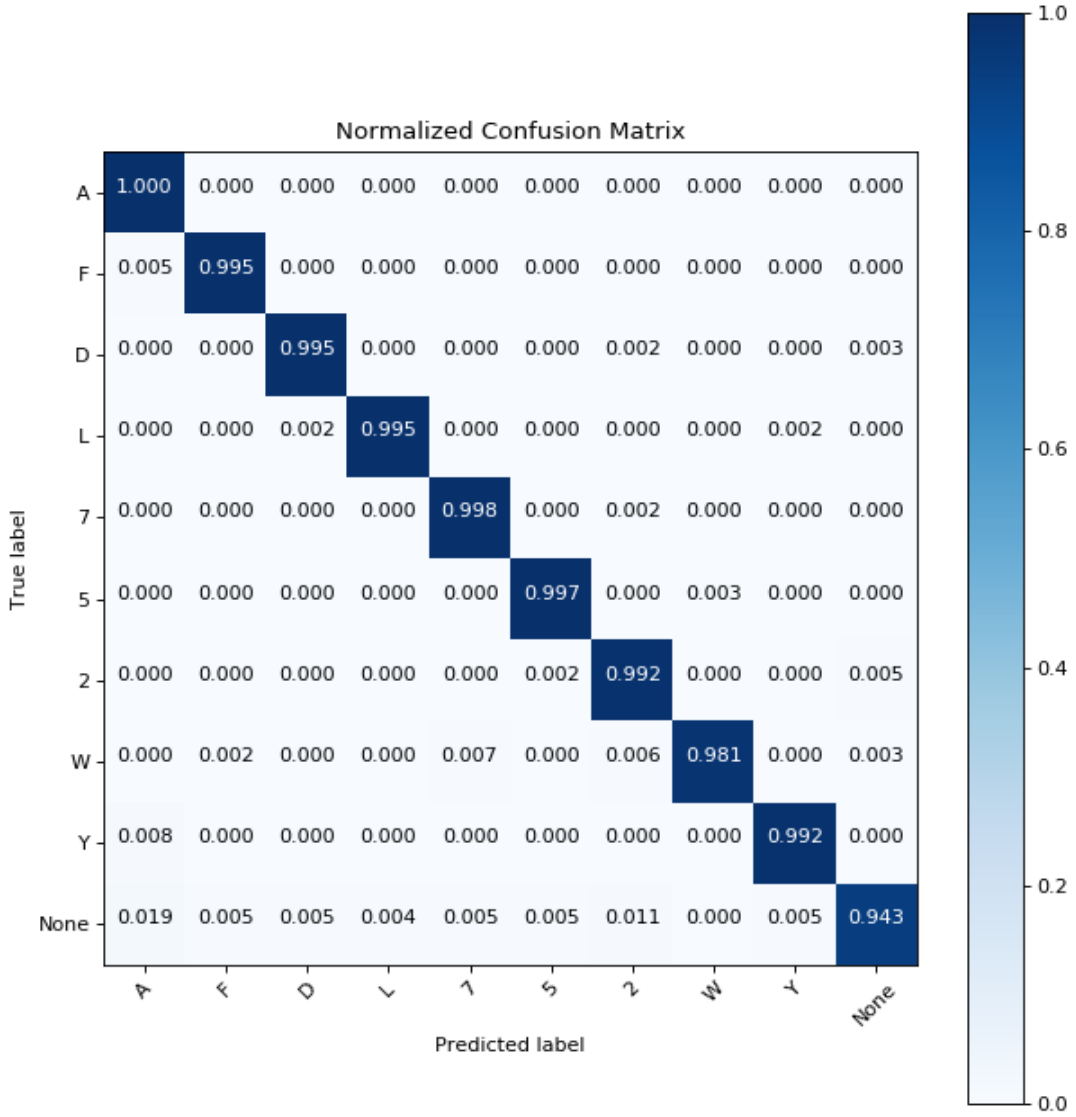


Figure 3.11 Normalized Confusion Matrix Quantified on Test-Set

3.4 OpenPHRI Integration

To control the robot and to remain safe during human-robot collaboration, we have used *OpenPHRI* open-source control library. This library allows to describe the task to perform

using force and velocity inputs in both the joint and task spaces while enforcing safety constraints such as velocity limitations, speed and separation monitoring or safety-rated monitored stops.

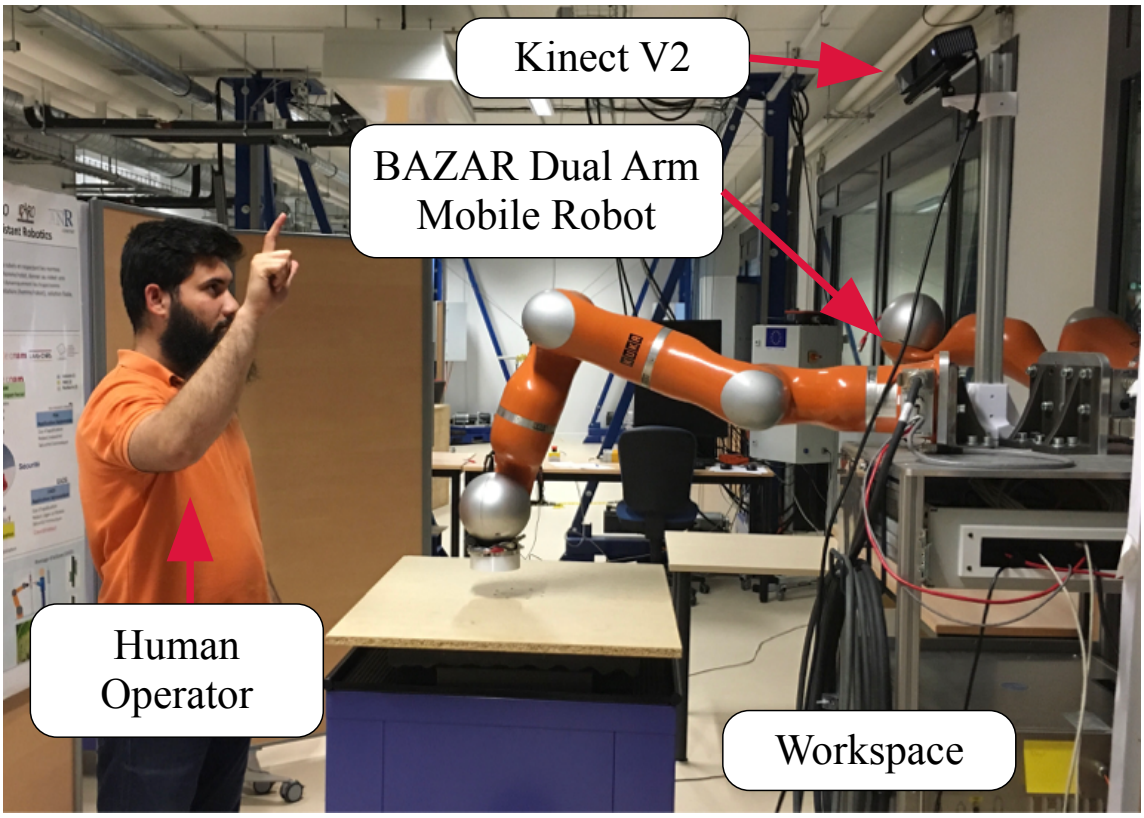


Figure 3.12 Safe Physical Human Robot Interaction Setup

As discussed in Section 2.1, ISO 10218-1/2 and ISO/TS 15066 have imposed safety requirements for industrial robot systems. Moreover, these ISO standards have identified four collaborative modes which are briefly explained as follows:

- *Safety-rated monitored stop* - This states that the human and robot can operate in a shared space but not at the same time. As soon as the human operator occupies the shared space, the robot must stop until the human exits the shared space.
- *Hand guiding* - In this mode, the human coworker can teach the robot positions/waypoints by physically moving the robot without any means of an intermediate interface.
- *Speed and separation monitoring* - This defines three zones of the shared space say red, yellow and green. The operation of robot depends on the presence of human in each zone. If human coworker is in the green zone the robot operates at its full speed, at reduced speed in yellow zone and it should stop in the red zone.

- *Power and force limiting* - This mode prescribes the limitation of power and force to allow humans to work side-by-side with the robot. The robot should be able to handle collisions with the human to prevent any harmful consequences.

OpenPHRI inherently is able to adopt all four collaborative modes efficiently. The first and the third modes however, require safety-rated monitoring sensors. As described in the previous sections, our proposed framework obtains a pseudo 3D human skeleton, which is used to determine the distance of the closest body part of the human coworker to the robot. This is integrated with *OpenPHRI* to complement the two collaborative modes.

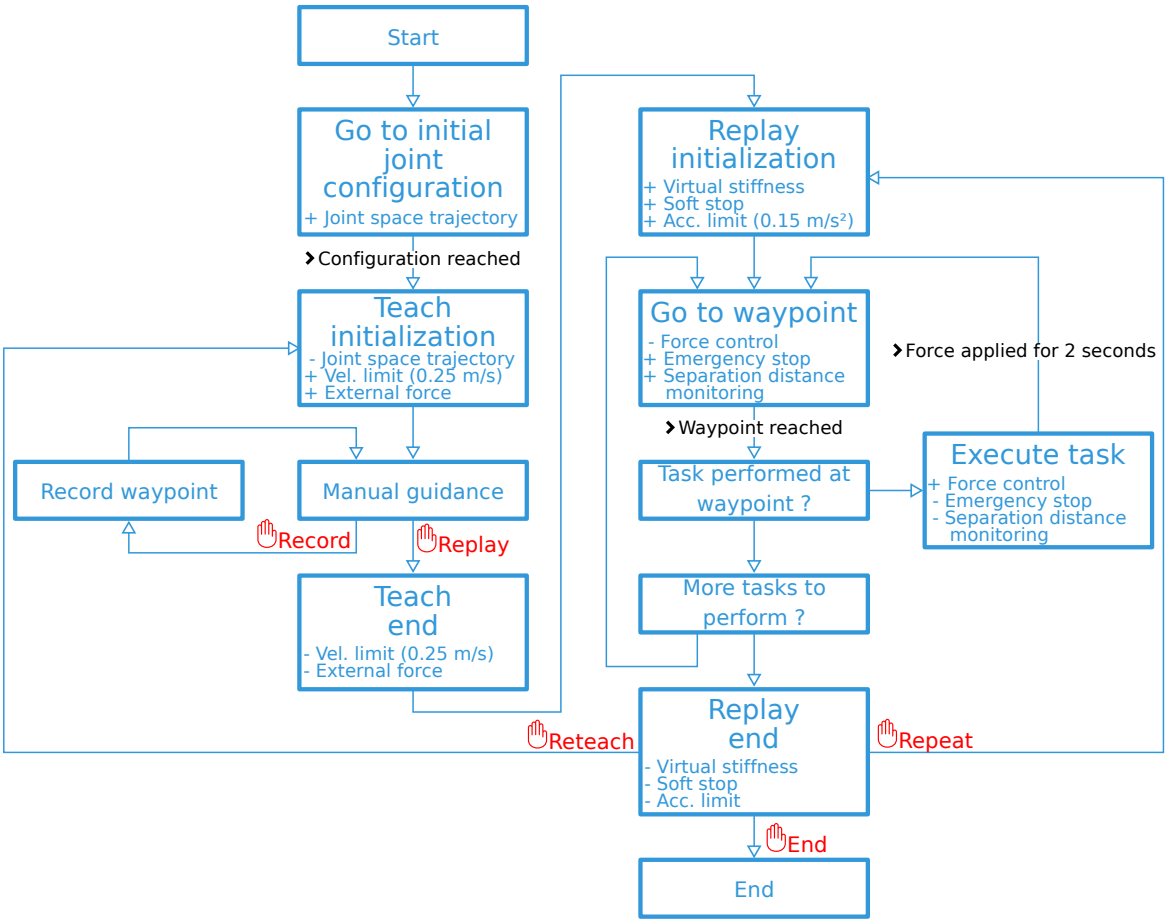


Figure 3.13 The FSM used for the experiment. A plus sign indicates an addition to the controller (a new constraint or new input) while a minus indicates a removal.

3.5 Example Industrial Application of the Proposed Framework

To demonstrate the effectiveness of the proposed approached, we set up an industrial-like experiment where multiple operators can safely interact sequentially with the robot using both hand gestures and physical contact. The experiment is decomposed into two phases: 1) a teaching by demonstration phase, where the user manually guides the robot to a set of waypoints and 2) a replay phase, where the robot autonomously goes to every recorded waypoint to perform a given task, here force control.

BAZAR robot used for the experiments is composed of two Kuka LWR 4+ arms with two Shadow Dexterous Hands attached at the end-effectors and a Kinect V2 mounted on top of it [168]. The arms are attached to a Neobotix MP700 omnidirectional mobile platform. In our scenario, shown in Figure 3.12 the mobile base is kept fixed and only the left arm, without the hand, is used. The communication with the embedded arm controller is done using the FRI library [5]. The external force applied to the arm's end-effector is estimated by the embedded controller (based on joint torque sensing and on knowledge of the robot's dynamic model) and retrieved by FRI. The control rate is set to 5ms. To orchestrate the experiment, we have designed a finite state machine (FSM), depicted in Figure 3.13. The transitions between the states are either automatic (no text), depending on sensory information (arrow with text) or triggered by gestures (hand sign with text).

A video of the experiment is available online[6] and snapshots are given in Figure 3.14. The experiment goes as follow. First, the robot goes to a predefined initial joint configuration before initializing the *Teach* phase. Once this initialization is performed, the robot is ready to be manually guided and taught the waypoints where the tasks have to be performed during the *Replay* phase. Each time a **Record** gesture (L letter sign) is detected, the current end-effector pose is recorded. When a **Replay** gesture (A letter sign) comes in, the *Teach* phase is ended and the *Replay* phase is initialized. Then, the robot goes to the first recorded waypoint while limiting its velocity thus ensuring safety of the human worker (speed and separation monitoring in the FSM) according to the distance of the closest detected body part. This distance corresponds to the depth value given by Kinect V2 at the joint image coordinates obtained from *openpose* as explained in Section 3.2.3. If the closest body part is occluded by the robotic arm, the depth value (that will then correspond to the depth value of the robot itself) is discarded while the next closest body part visible in the scene is considered a reference for depth.

[5] https://cs.stanford.edu/people/tkr/fri/html/
[6] https://www.youtube.com/watch?v=lB5vXc8LMnk

Figure 3.14 Screenshots from the robotic experiment by operators Op1 and Op2 (a) Op1 manually guiding the robot to a waypoint in the workspace. (b) Op1 records the way-points using Record gesture. (c) Op1 replay the taught waypoints by Replay gesture. (d) Op2 stands far from the robot so it moves with full speed. (e) Op2 stops the robot by applying external force (or accidental touch). (f) Op2 stands near the robot, so it moves slowly ensuring operator's safety. (g) Op2 gives Reteach command to the robot. (h) Op2 sets the new waypoints manually. (i) Op2 gives Record command. (j) Op2 stops the robot by Stop gesture. (k) Op2 resumes the robot operation by Resume gesture. (l) Op1 ends the robot operation by giving End command.

This estimation of body parts distance is not available with the default output of OpenPose but it is possible, thanks to our integration, of Kinect V2 depth map. This amplifies the usefulness of OpenPose skeleton extraction while assuring a safe interaction of a human coworker with the robot. While in autonomous motion, the robot can be stopped at any time (Soft Stop constraint in the FSM) using a **Stop** gesture (number 5 sign). Making this gesture will slow down the robot until a full stop is reached. This is useful if an operator must enter the robot workspace without fearing any injury. The **Resume** gesture (Y letter sign) can be made to resume normal operation. When the robot reaches the waypoint, it switches to the task execution. In this scenario the task is to apply a 30N force for 2s along the vertical axis. Once the task has been executed, the robot goes back to its waypoint and moves to the next

ones to repeat the same operations. If the task has been performed at all the waypoints, the *Replay* phase ends and the next action is determined by the operator. A **Reteach** gesture (number 7 sign) will move the FSM to the *Teach* phase while a **Repeat** gesture (F letter sign) will repeat all the tasks at the recorded waypoints. If no other operation is needed, an **End** gesture (number 2 sign) will end the experiment.

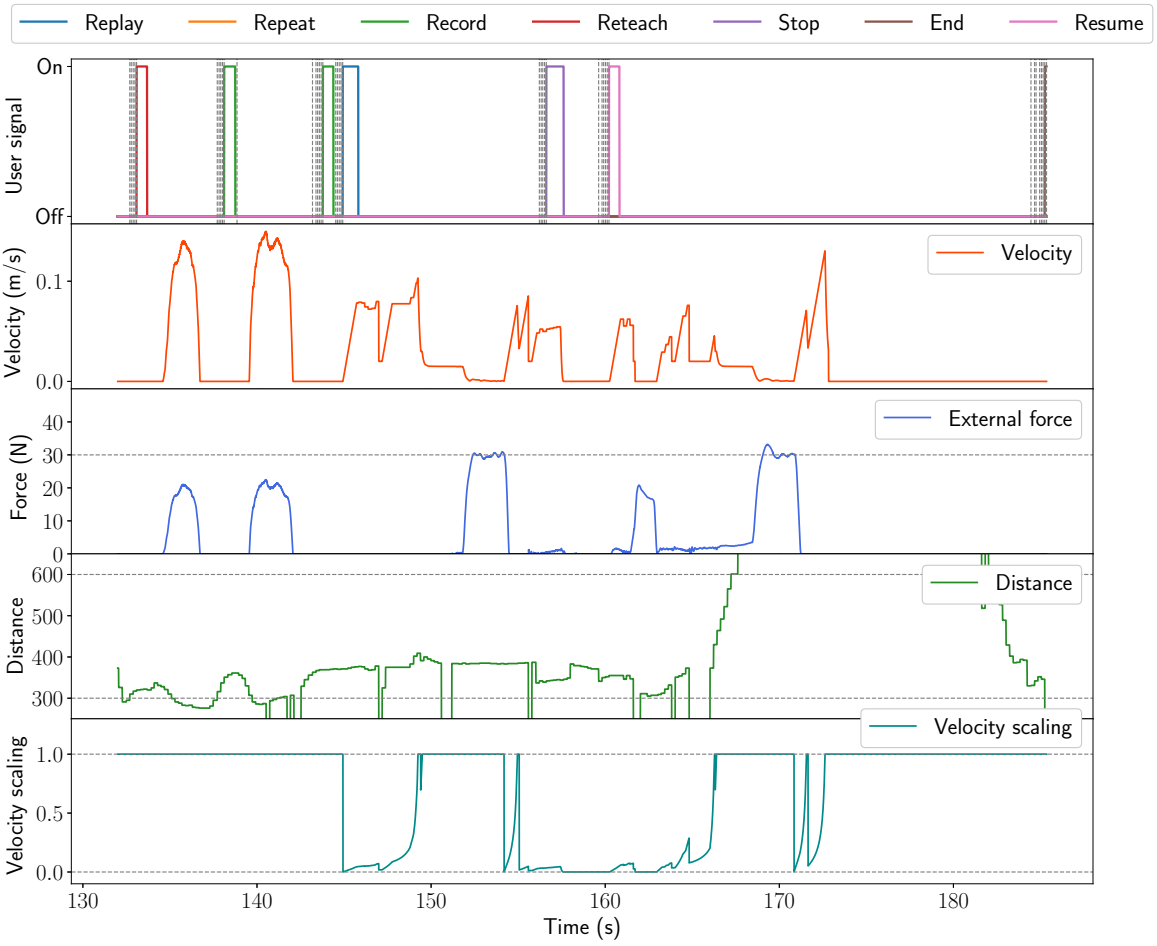


Figure 3.15 Experimental results. From top to bottom: hand gesture detection (dashed lines correspond to detection instants and plain line to the activation signals), control point translational velocity, external force at the end-effector, distance between the camera and the closest human body part and velocity scaling factor computed by OpenPHRI to slow down the motion.

Experimental results are show in Fig. 3.15. The time axis has been limited to the 132-185s range for better readability. The top graph displays the result of the hand gesture detection where each vertical dashed line corresponds to the detection of a gesture. To filter out false positives, a gesture is considered valid if it appears in five consecutive frames. Considering the hand-gesture detection frame rate of 20Hz, this gives a 250ms delay between the making

of the gesture and its detection. This delay should not impact human-robot interaction since the average human reaction time usually lies within the 200-250ms range[7]. Once the same gesture has been detected five times in a row, the corresponding signal is activated. False positives can be observed, e.g. at t=139 s when the first record signal ends, but thanks to the filtering systems no incorrect signal activation is made. The two following graphs in Fig. 3.15 show the end-effector translational velocity and force.

It can be seen that through the *Teach* phase, i.e. until t=135 s, the velocity simply follows the force applied to the robot. Then, the *Replay* phase starts and the end-effector velocity is now the result of the motion made to reach the waypoints and also by the force regulation applied at these locations. Between the two task executions (t=153 s and t=170s), one can observe some force applied to the robot at t=162 s. A safety feature is programmed to prevent accidents due to unexpected contact between the operator and the robot, leading to a monitored stop. In this situation, the robot stays still until the contact disappear and then resumes its motion to the second waypoints. The fourth graph displays the distance to the closest body part. The values are the raw ones provided by the Kinect V2 and are unitless. As mentioned previously, this distance is used to adapt the velocity limitation so that the robot can move quickly when nobody is around but slows down when an operator is approaching. The velocity limit is at a minimum of $0.02m/s$ at a distance of 300 and at a maximum of $0.3m/s$ at a distance of 600. The effect of this limitation can be observed multiple times, including after the beginning of the *Replay* phase where the distance suddenly drops below 300, enforcing a very slow motion of the robot. The last graph shows the evolution of the scaling factor computed by *OpenPHRI*. A value equals to one means that no velocity reduction has to be performed to comply with the constraints (velocity and acceleration limits, speed and separation monitoring and safety-rated monitored or soft stop). When at least one constraint would not be respected considering the current inputs, the scaling factor decreases below one to make sure that all constraints are satisfied. When the value reaches zero, the robot is at a complete stop. Using this technique allows to easily slow down the robot only when it is necessary.

3.6 Conclusion

In the perspective of smart factories – also known as factories of the future – we have introduced a real-time human-robot interaction framework for robot teaching using hand gestures. The proposed framework relies on our novel rotation and background invariant robust hand gesture detector. This is achieved by adapting a pre-trained state-of-the-art

[7]http://humanbenchmark.com/tests/reactiontime

convolutional neural network, namely Inception V3, to the classification of 10 hand gestures. The CNN is trained on an image dataset of 10 hand gestures, recorded with the help of 10 volunteers. The dataset *opensign*, is open and available to the computer vision community for benchmarking. We also release the source code of our hand gesture detector[8].

The accuracy of the trained CNN is validated with a set of test images and is found to be 98.9%. To reaffirm the quality of the hand gesture detector and to validate it on a mock-up example industrial scenario, we perform a robotic experiment. Safety and effectiveness of the experiment are guaranteed by our physical human-robot interaction library, *OpenPHRI*. Besides, real-time operation is established by asynchronous integration of the different modules present in our framework. The experiment proves the efficiency of the proposed framework, that ensures a natural means for robot programming. The robot is also aware of its distance from the human worker thanks to the integration of Kinect V2 and *openpose*. To guarantee the safety of the human coworker in close vicinity, the robot slows down using the velocity scaling feature of *OpenPHRI*.

The presented approach requires the user to know the gestures the robot can perceive. However, once s/he has memorized these gestures, it will be more natural for her/him to communicate with the robot. Integrating face identification algorithms in this framework, could also be a security feature. It will allow only selected people to interact with the robot without entering any passwords or fingerprints scanning which might require the users to come in close proximity to the robot.

Despite the quantified accuracy and experimental results, the capabilities of our system are limited by the depth range of the vision sensor. Moreover, the system is trained and tested in indoor settings and may fail in bright light due to the resulting contrast in RGB images. Backgrounds with intense texture may also compromise detection. To handle this, distinct background images should be substituted in the hand images to train the proposed network. Nevertheless, we believe that the results presented in this chapter are a very promising step towards the development of vision-based robot programming framework.

[8]https://github.com/OsamaMazhar/openhandgesture

Chapter 4

Dynamic Gestures Detection

Activity recognition or dynamic gestures detection is a problem that has been widely studied for mainly two objectives; to develop an alternative to traditional input devices in human-computer/machine interfaces like mouse, keyboard, teach pendants and even touch interfaces [169], plus to analyze video content to deal with the recent explosion of data and information on internet [170] and for applications such as automatic video surveillance [171]. In the context of human-computer interaction, gestures driven applications include interactive games [172, 173], sign language recognition [174, 84] and robot control [175, 176, 22].

Inspired by the long-term recurrent convolutional network (LRCN) model proposed in [90], we develop our CNN-LSTM network to model spatio-temporal dependencies to recognize dynamic gestures. We adopt the idea of fine-tuning CNN as proposed in [98], to model frame-level spatial appearance of input image(s). Our CNN model is an Inception-V3 network, which is fine-tuned on our background substituted static hand gesture dataset, thus able to extract subtle hand movements efficiently. We are also convinced by the idea of *visual attention* presented in [92] which is inspired by the human perception of selective focus. In particular, we implement the conception of pose-driven spatial attention mechanism as proposed in [94], in a manner that the hand images are cropped from full RGB frames guided by the hand position and pose obtained through the 2D skeleton extractor. We also use augmented pose as an additional modality to the hand images in our work inspired by [84]. The augmented pose allows our network to learn large-scale upper-body motions, while subtle hand movements that distinguish several inter-class dependencies are learned by the spatial-attention module. Our overall strategy to detect dynamic gestures from monocular cameras is illustrated in Figure 4.1.

4.1 Our Contributions

- We propose a dynamic gestures classification strategy based on CNN-LSTM network with the state-of-the-art performance on Chalearn 2016 isolated gesture recognition dataset [26].
- Gesture recognition is performed on pure RGB images without the need of any specialized sensor.
- We present the idea of learning-based depth estimators to predict the distance of the person and his/her hands from the camera exploiting only the upper-body 2D skeleton coordinates.
- Hand deep features are extracted through a pre-trained CNN on background augmented static hand gestures dataset for effective feature extraction.
- A multi-stage learning pipeline is proposed for training large-scale video dataset on machines with less computational power.
- We also present the state-of-the-art performance on Praxis cognitive assessment dataset [2] on correctly performed gestures.

4.2 Datasets Description

The proposed work exploits two dynamic gestures datasets i.e., large-scale *Chalearn 2016 Looking at People isolated gesture recognition* dataset [26] which is actually created from *Chalearn 2011 gestures* dataset [177], and *Praxis cognitive assessment* dataset [2].

4.2.1 Chalearn 2016 Isolated Gesture Recognition Dataset

The gesture vocabulary in this dataset is taken mainly from nine categories corresponding to different application domains including body language gestures (like scratching head or crossing arms), gesticulations, illustrators (like Italian gestures), emblems (like Indian Mudras), sign language, semaphores (like referee signals, guiding machinery or a vehicle), pantomimes, actions/activities (like drinking or writing) and dance postures. The gestures recordings are performed through Microsoft Kinect [178, 179], which was fixed at approximately 4 feet away from the head of a volunteer. The volunteers were requested to stand approximately 1 foot away from the wall to get depth contrast against the background. The

frame size of images/videos is 320×240 pixels while the camera is set such that the upper body including head, shoulders and waists are visible in most videos.

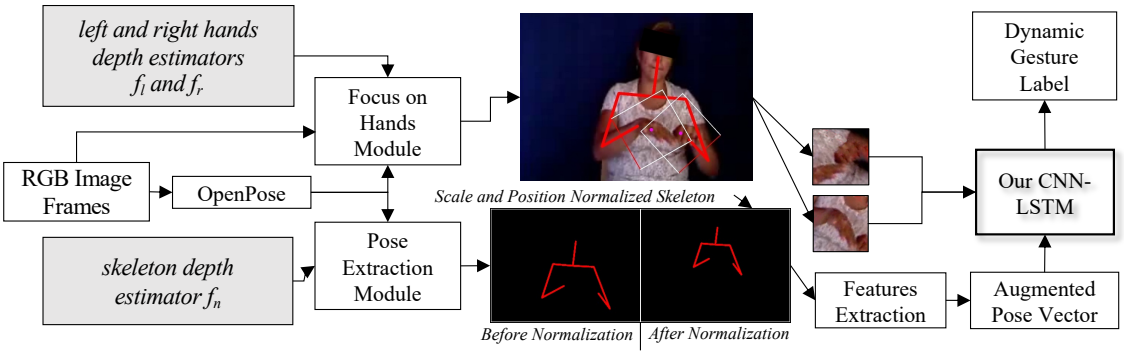


Figure 4.1 Illustration of our overall proposed strategy for dynamic gestures detection. We employ *openpose* to extract raw 2D upper-body pose plus hands key-points of the person from monocular images. Then, raw hand key-points, original RGB input frames and our learning-based hands depth estimators f_l and f_r are passed to Focus on Hands Module which performs filtering of hand key-points and subsequently crops hands images of the person performing gestures. Pose Extraction Module performs skeleton filtering by interpolating missing skeletal joints coordinates as well as Gaussian smoothing of the pose. It also perfoms "Scale and Position Normalization" exploiting our learning-based skeleton depth estimator f_n. For visualization purpose, offset values are added to the normalized skeleton in the display image. Then, normalized skeleton is passed to Features Extraction block which performs pose augmentation. Cropped left/right hands images and *augmented pose vector* are then fed into our proposed CNN-LSTM that outputs framewise static hand gestures labels and dynamic gesture label for the input video.

The dataset also contains videos with full-body frames as well as those with volunteers sitting while performing gestures. This dataset has 47,930 RGB and depth videos in total while each video represents only one gesture. Total number of gestures are 249 which are performed by 21 different individuals. The dataset has been divided into three mutually exclusive subsets namely training, validation and test sets. Training set has 35,876 videos with gestures performed by 17 volunteers. Validation and test sets have 5,783 and 6,271 videos respectively performed by 2 volunteers each.

In our work, we first arranged the gestures with respect to the number of videos available in the training dataset. We realized that videos are not uniformly distributed among all gestures. Some labels contain up to 851 videos while others only have around 64 with mean number of videos equal to 144 and 36 average frames in each video for all 249 gestures. This arrangement is beneficial if a network is to be trained on selective gestures. In that case, the gestures with higher number of videos can be the preferred choice to start training the network with. This distribution of train, valid and test data in our work is slightly different

than the proposed approach in the challenge. We combine and shuffle the provided train, valid and test sets together which brings the total number of videos to 47,930. The network is trained on 35,930 videos while hyper-parameters are optimized on the validation data of 6,000 videos. The trained model is evaluated and a confusion matrix/heat-map is generated on the test data of 6,000 videos.

4.2.2 Praxis Cognitive Assessment Dataset

Praxis gesture dataset [2] is designed to diagnose *apraxia*, which is a motor disorder caused by brain damage. *Apraxia* is a neurodegenerative disorder in which the patient has a difficulty to plan and perform motor tasks despite his/her willingness and the fact that request to execute the task has been fully understood [180]. This dataset contains RGB (960×540 resolution) and depth (512×424 resolution) images recorded by 60 subjects plus 4 clinicians with Kinect V2. From the volunteers, 29 were elderly with normal cognitive functionality while others had medical conditions which include amnestic mild cognitive impairment (MCI), unspecified MCI, mixed dementia, Alzheimer's disease, posterior cortical atrophy, corticobasal degeneration and severe cognitive impairment (SCI). For this dataset, 29 gestures were performed by the volunteers in total (15 static and 14 dynamic gestures). These gestures are divided into three categories: abstract, symbolic and pantomimes. Gestures image sequences in the dataset are additionally labeled as "correct" or "incorrect" depending on the execution of gestures by the volunteers which is based on the clinicians opinion.

In our work, only dynamic gestures i.e., 14 classes are considered while their pathological aspect is not taken into account i.e., only gestures labeled "correct" are selected. Thus, the total number of considered videos in this dataset is 1247 with mean length of all samples equal to 54 frames.

4.3 Spatial Attention Module

We can divide spatial attention section into two parts: pose extraction module and focus on hands module.

4.3.1 Pose Extraction Module

The strategies found in the literature that employ skeleton information, normally exploit 3D-pose already provided in the datasets, which is obtained through sensors like Microsoft Kinect as in the case of *Chalearn 2014 Looking at People Challenge*. This makes these strategies device dependent, even if they operate on RGB images in case of multi-modal methods like

[94], and will not be applicable to the systems which lack these (or similar depth) sensors until a reliable 3D-pose estimator from only RGB images is developed. Lately, effective 2D skeleton extractors based on deep methods like *openpose* [66, 65] have been released to the open-source community. For this reason, we opted to devise our algorithm such that it operates only in RGB image domain, thus can be implemented on applications/robots that lack depth sensors. For skeleton extraction, as mentioned above, we employ *openpose* to extract 2D full body skeleton plus hands skeleton (optionally, depending on the availability of computational power) to localize hands in the scene accurately. Any other skeleton extractor like [152] can be employed in place of *openpose*. We first resize the dataset videos to $1080 \times C$ pixels where C is corresponding value of resized image columns obtained with respect to new row value i.e., 1080, while maintaining the aspect ratio of the original image; 1440 in this case. Then all the resized videos are fed into *openpose* skeleton extractor, one at a time, with hand key-point extraction flag raised and the output raw skeleton coordinates are saved on the disk as *.json* files for offline processing.

Filtering Skeleton

Openpose is a discriminative 2D pose estimation approach which extracts N skeleton coordinates frame-by-frame and doesn't employ pose tracking as already mentioned in Chapter 3. The occasional jitter in the skeleton output or absence of joint coordinates within successive frames may cause problems in gesture learning/modeling. Thus we develop a pose filtering strategy that rectify random disappearance of the joint(s) coordinates plus smooths the skeleton output. We work on a window of frames with adjustable size K of odd numbers (selected value is 7) and first perform coordinates replacement for the missing joints. This procedure is driven by the following two equations:

$$p_{nK} = p_{nK-1}, \text{ if } (p_{nk})_{k=1}^{K-1} \neq 0 \wedge p_{nK} = 0 \wedge \sum_{i_n=0}^{K+1} 1 \leq K \tag{4.1}$$

$$(p_{nk})_{k=1}^{K} = \begin{cases} 0, & \text{if } p_{nK} = 0 \wedge \sum\limits_{i_n=0}^{K+1} 1 > K \\ p_{nK}, & \text{if } p_{nK} \neq 0 \wedge (p_{nk})_{k=1}^{K-1} = 0 \end{cases} \tag{4.2}$$

where p_{nk} are coordinate values of the n_{th} joint in the output pose vector of *openpose* in frame k of the selected window, p_{nK} are the coordinate values of the same joint in the last frame of the window and i_n is coordinate replacement counter for the skeleton joint in question.

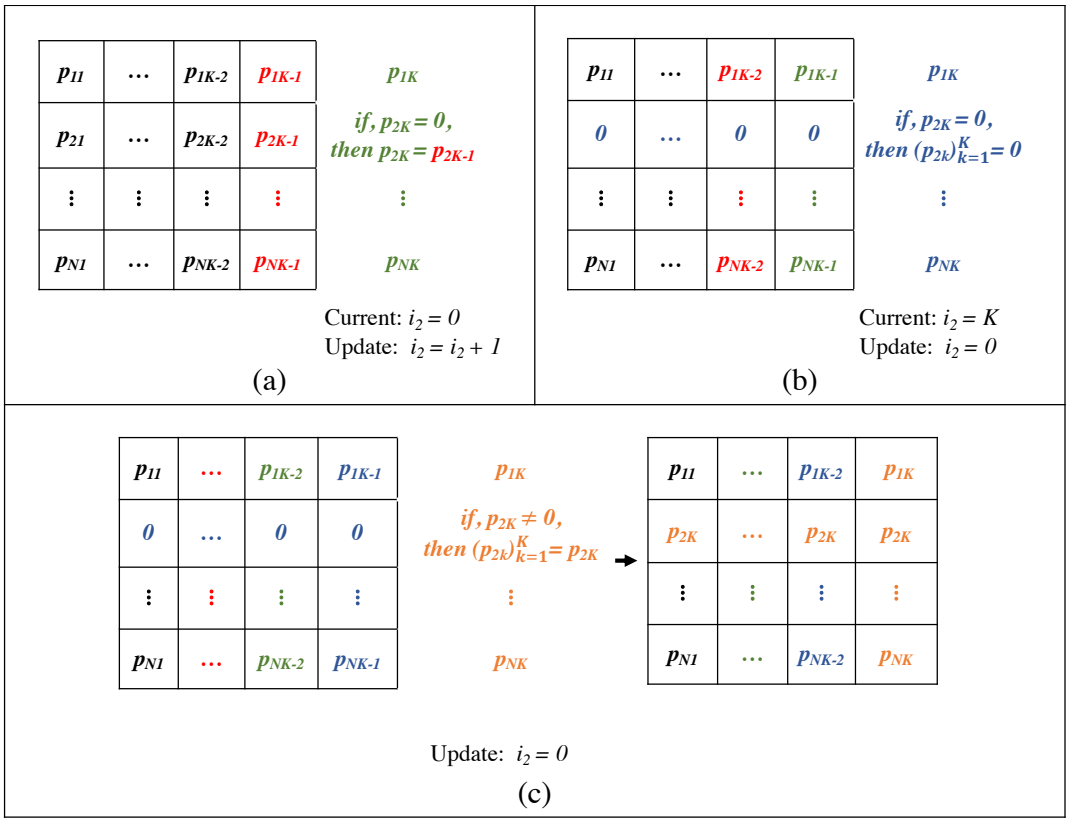


Figure 4.2 Illustration of the proposed coordinate replacement strategy. All variables are assumed non-zero unless stated otherwise. (a) If a joint coordinate, say p_{2K} in the new frame K is zero, while $(p_{2k})_{k=1}^{k=K-1} \neq 0$, then p_{2K} is replaced by the immediate previous non-zero value i.e., p_{2K-1} and coordinate replacement counter for this joint i_2 is incremented by 1. This process may be repeated consecutively if same conditions persist until i_2 equals to K. If a non-zero value of p_{2K} reappears before i_2 equals to K, i_2 is reset. (b) If same conditions continue as in a, while the value of i_2 reach limit K, all values corresponding to considered joint i.e., $(p_{2k})_{k=1}^{k=K}$ in this case, are replaced with 0 and coordinate replacement counter is reset. (c) If all previous values corresponding to a joint e.g., $(p_{2k})_{k=1}^{k=K-1}$ are zero, while $p_{2K} \neq 0$, then all values in the window corresponding to the considered joint are replaced by non-zero p_{2K} in this case.

The missing joint coordinate values are replaced with the immediate previous i.e., p_{nK-1} (non-zero) values in the pose vector. Only K consecutive replacements are allowed for each joint and this is monitored by the coordinate replacement counter i_n. If a joint was absent in the entire window of K frames, and non-zero coordinate values appears in new frame of the window, all the corresponding joint values in the pose vector are replaced with the new non-zero value. This procedure is explained with an example in Figure 4.2.

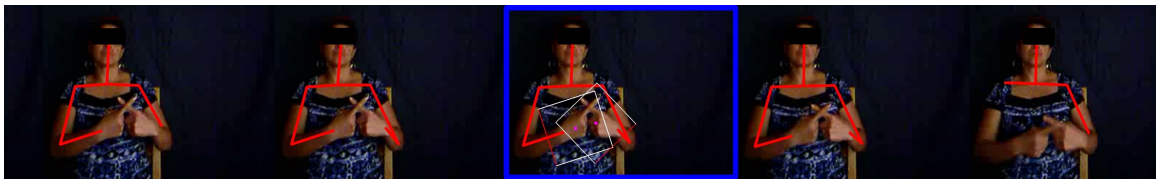


Figure 4.3 An illustration of proposed skeleton filtering in function. The frames are arranged from left to right with respect to their appearance in time. It can be observed that the output (frame with blue boundary) of our filtering strategy copes with the missing joint coordinates in the window. Gaussian smoothing is also applied on the joints which absorbs jitter in the raw output skeleton from *openpose*. The opted size of window is 7 frames, while only 5 frames are shown for clarity. The bounding boxes on the hands are extracted as a part of proposed *spatial attention module* detailed in Section 4.3.2

.

In the second step of skeleton filtering, we apply Gaussian smoothing to the individual joint pose vectors. The application of this filter remarkably solves the jitter problem in the skeleton pose and smooths out the joint movements in the frame positioned at the center of selected images window. The output of proposed filtering strategy is shown in Figure 4.3.

Scale and position Normalization of the pose

Monocular (2D or even 3D) pose estimations, as obtained through *openpose*, are scale ambiguous [144, 146]. The information about depth/distance of the user from camera, or pose output in true metric space, can help to determine the size of hand bounding box, or to eliminate undesired influence of variable user distance from our gesture detection algorithm. This phenomenon is illustrated in Figure 4.4. One approach to obtain approximate 2D pose in metric space is to map/calibrate the height of predicted skeleton from image pixels domain to distance of the person from camera provided height of the user in metric space is known. Another approach to acquire a calibrated skeleton output (specifically 3D), given height of the user in true metric space, is through design of a predictor (e.g., a deep network) that returns height-normalized skeletons, as presented in [144]. We present a novel learning-based strategy to approximate user distance from a monocular camera and to estimate the scale-factor in 2D pose predictions obtained through *openpose*, without an obligation of known user height.

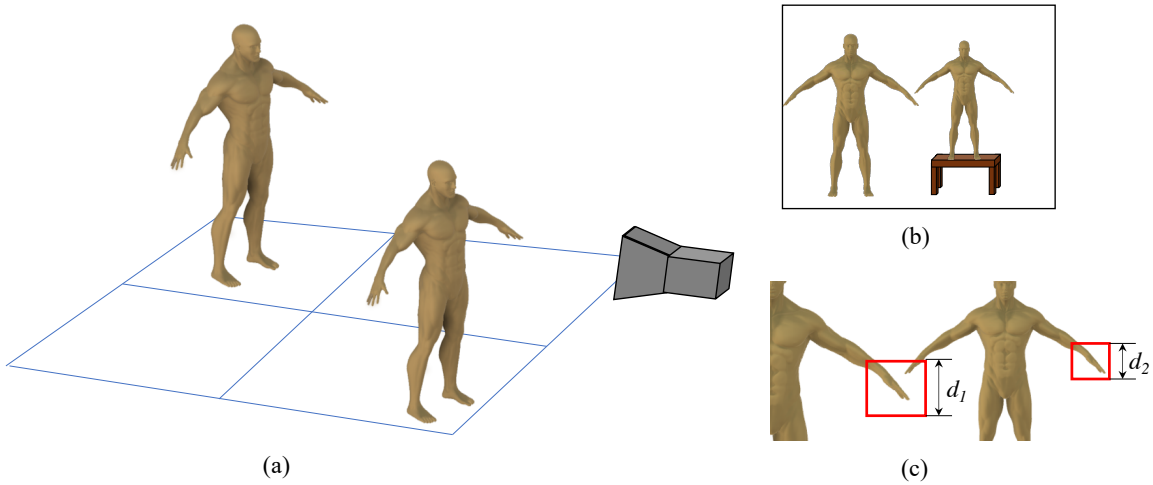


Figure 4.4 An illustration of scale ambiguity in monocular images; (a) Exhibits a scene where two humans of the identical physical form are standing at different distances from the camera. (b) Represents an image taken from the monocular camera. It is evident that no depth information can be perceived from such an image. To emphasize this ambiguity, we place a virtual table beneath the feet of apparently smaller human which fakes him as a child who stands besides an adult at the same distance from the camera. (c) Highlights our problem more evidently, that the scale ambiguity in monocular images makes it hard to determine the region-of-interest, driven by d_1 and d_2 variables, around hands of the persons for gestures detection.

Our gestures detection work is based on 8 upper-body joint 2D coordinates as shown in Fig. 4.5, denoted as $\mathbf{p}^{(i)}(x^i, y^i)$, $i = 0 \ldots 7$ ($i = 0$ corresponds to the *Neck* joint, which is considered as a root node) where x^i and y^i are image coordinates of skeletal joint i. The proposed concept of estimating user depth from the selected upper-body coordinates is to exploit a RGB-D sensor dataset, devise a neural network and learn its parameters which maps information from 8 upper-body skeleton coordinates to the ground-truth depth of *Neck* joint. Inspired by the work in [84], which demonstrated that augmenting pose coordinates may improve performance, we develop a 97 components pose vector x_n from 8 upper-body joint coordinates.

To eliminate the influence of user position in image, the coordinates of *Neck* joint are subtracted from rest of the vectors $\mathbf{p}^{(i)}$. Non-zero coordinates of the joints are exploited to obtain a *line of best fit* through least square method. In addition to 7 vectors from anatomically connected joints, 21 vectors between unique pairs of all upper-body coordinates are also obtained while their abscissas and ordinates are saved. The lengths of individual augmented vectors are also computed. Then 6 angles formed by all triplets of anatomically connected joints (in image plane) are calculated. 28 more angles are estimated between 28 (anatomically connected plus augmented) vectors and the previously obtained *line of best fit*. The resultant

97 components augmented pose vector include 42 elements from abscissas and ordinates of the augmented vectors, their 21 estimated lengths and 34 computed angles concatenated together.

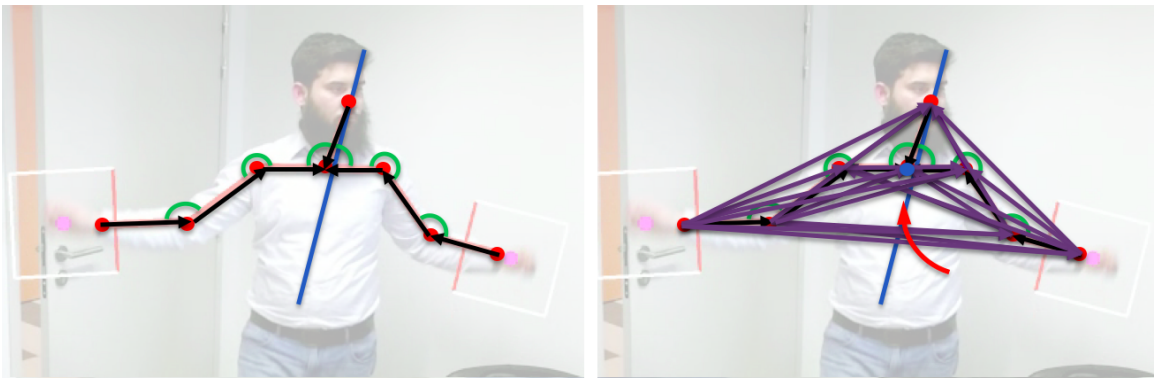


Figure 4.5 Manual features extraction/pose augmentation from upper body skeleton for scale-normalization is presented. In the left image, we show 8 upper-body joint coordinates (red circles), vectors joining these joints (black lines) and angles between these vectors (shown in green). From all non-zero upper-body coordinates, we compute a *line of best fit* (in blue) which can clearly be seen in the left image. In the right, we show all the augmented vectors (in purple) between unique pairs of all upper-body joints. The angles between augmented vectors and the line of best fit are also computed which are not visible in this image. The reference neck depth coordinate (blue circle), obtained through Kinect V2 depth map, is also shown in the right figure against which the 97 components augmented pose vector is mapped to estimate approximate depth of the person.

To obtain the ground-truth depth/distance information of *Neck* joint, we utilize *opensign* static hand gestures dataset [181]. This dataset include recordings of RGB and depth image streams from Kinect V2 acquired at approximate frame rate of 20 fps for 10 static hand gestures. Each gesture is recorded for around 12 seconds by 10 volunteers each, with both hands at three distances of 5, 3, and 1.5 meters away from the sensor. We execute our augmented pose extractor on all video files in the dataset and save corresponding raw distances of *Neck* joint against each 97 component feature vector. A 9 layers neural network f_n is then realized that optimizes parameters $\theta_{\mathbf{n}}$ given an augmented pose vector $\mathbf{x_n}$ and ground-truth d_n to regress approximate distance value $\widetilde{d_n}$ with mean squared error of 8.34×10^{-4}. This relationship can be formalized by the following equation:

$$\widetilde{d_n} = f_n(\mathbf{x_n}, d_n; \theta_{\mathbf{n}}) \tag{4.3}$$

Thus the estimated 2D pose from *openpose* is scale-normalized by multiplying with a computed scale-factor which is obtained by dividing the predicted distance $\widetilde{d_n}$, given the

augmented pose $\mathbf{x_n}$ for each frame, with an empirical integer value without a need of depth sensor or known user height.

Our scale normalization strategy may not be able to estimate actual distance of a person in true metric space from camera in the absence of known user height specifically in extreme cases (e.g., an extra-ordinary tall user or a child). However, the intended operation of our proposed strategy in this work i.e., to scale-normalize 2D skeleton for gesture detection, functions properly for all users irrespective of their heights.

4.3.2 Focus on Hands Module

We focus the attention of our dynamic gesture detector on hands in two steps. First step is to localize hands in the scene and second step is to determine size of the bounding box of detected hands and cropping the hand images.

Hand Localization

One possibility to localize hands in the image is to utilize object detectors like [182–184], trained on hand images as presented in [185]. Exploitation of such strategies also brings along problems of distinguishing left and right hands as their detection is performed in a local manner thus lack contextual information. This problem has also been addressed in [185] by training the detector with two classes "hands" and "head". This extra information about the location of head in the scene aids in distinguishing left and right hands. However, such a solution can also fail to solve this ambiguity, without extra information known, if the arms are crossed.

To avoid these complexities, we chose to employ *openpose* for hand localization in the scene. We perform this operation in two ways. The preferred way is to exploit the hand key-points detection of *openpose* for hand localization. The library outputs 21 key-points for a single hand image. From our observations, the extraction of hand key-points from *openpose* is more susceptible to the problems of jitter and misdetections of the keypoints than in the skeleton extraction, specifically on low resolution images/videos as in *Chalearn 2016 Looking at People isolated gesture recognition* dataset. Consequently, we apply the same filtering operations driven by equations 4.1 and 4.2 described in Section 4.3.1 on the raw hand key-points obtained from *openpose*. Once the filtered hand key-points are obtained, we estimate the mean of all non-zero coordinates as presented in the following equation:

$$p_c(x,y) = \frac{1}{N^*} \sum_{i=1}^{N^*} p_i(x,y) \tag{4.4}$$

where $p_c(x,y)$ is the obtained hand center coordinates in the image, N^* is the number of non-zero hand key-points, while $p_i(x,y)$ are found (thus non-zero) hand key-points. This hand localization method is precise and does not posses ambiguity in distinguishing left and right hands. If, *openpose* fails to detect hand key-points at all, hand localization module switch to the second method we have developed, similar to the one presented in [23]. The second approach to localize hands operates by fitting a line between elbow and wrist joints 2D image coordinates similar to what has been presented in Section 3.2.2. This line is extended by one-third of its length (empirical value) along the same direction towards hands. The end point of the extended line is considered hand center coordinates. The inclination angle of this line is also utilized to determine rotation of the bounding box, which we will discuss in next section.

It has to be noted that opting to extract hand key-points through *openpose* is a computationally expensive approach. We optionally can skip this methodology in its entirety and may adopt the second approach of exploiting only elbow and wrist joints (obtained by extracting full body pose) if available computational power is limited (e.g., in laptops). In our case, we run openpose on 1080×1440 sized videos with two Nvidia's GeForce GTX 1080 desktop GPUs and obtain around 10 frames per second with hand key-point extraction enabled. On a laptop with Nvidia's GeForce GTX 1060 Max-Q GPU, we are able to obtain only about 3 frames per second with the same *openpose* configuration and input videos.

Bounding-box Size Estimation

Once the hands are located in the image, it needs to be cropped at parts held by hands. To crop input image pixels occupied by hands, we need to determine the size of hand(s) bounding box(es). Since our gestures detection system relies only on RGB images in run-time, we develop two additional neural networks which learn to estimate the size of hands bounding boxes analogous to that described in Section 4.3.1, Equation 4.3. Following the approach detailed for scale-normalization of the pose, we formulate 54 components augmented pose vector separately for each hand as shown in Figure 4.6.

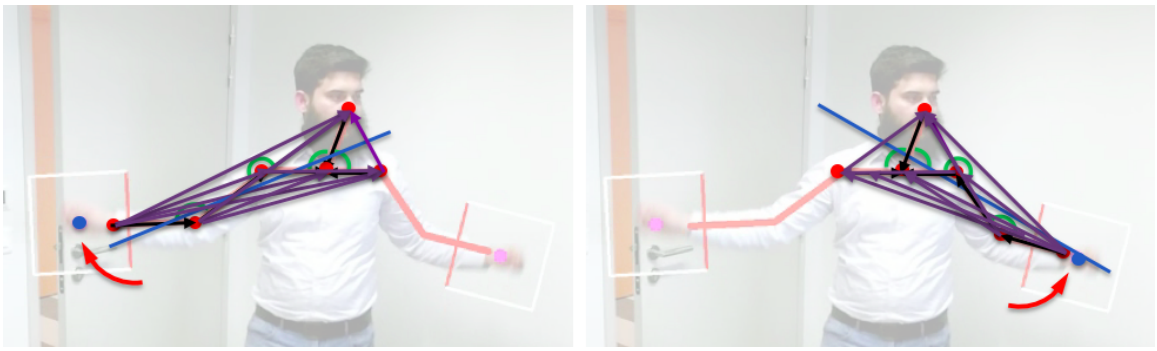


Figure 4.6 Illustration of augmented pose vector features for hands bounding boxes size estimation.

These augmented pose vectors are mapped against ground-truth hands raw depth values obtained from *opensign* dataset, through two separate neural networks described by following two representations:

$$\widetilde{d_l} = f_l(\mathbf{x_l}, d_l; \theta_\mathbf{l}) \tag{4.5}$$

$$\widetilde{d_r} = f_r(\mathbf{x_r}, d_r; \theta_\mathbf{r}) \tag{4.6}$$

where f_l and f_r represent 9 layers neural networks that optimize parameters $\theta_\mathbf{l}$ and $\theta_\mathbf{r}$ given augmented poses $\mathbf{x_l}$ and $\mathbf{x_r}$ for left and right hands respectively while ground-truth raw depth values of left and right hands d_l and d_r are obtained from Kinect V2 to estimate approximate depths $\widetilde{d_l}$ and $\widetilde{d_r}$. Mean squared error for f_l and f_r are 4.50×10^{-4} and 6.83×10^{-4} respectively. Thus our proposed system can detect size of hand bounding boxes formulated from the estimated depths only from pure RGB images.

4.4 Video Data Processing

Our proposed spatial attention module conceptually allow end-to-end training of the gestures. However, we train our network in multiple stages to speed-up training process, the details of which will be discussed in Section 4.6. This however, requires videos to be processed step-by-step beforehand. This procedure is executed in four steps i.e, (1) 2D pose-estimation, (2) features extraction, (3) label-wise sorting and zero-padding and (4) train-ready data formulation. Prior 2D-pose estimation may be considered a compulsory step even if the network is trained in an end-to-end fashion while other steps can be integrated into the algorithm. 2D pose-estimation, skeleton filtering and scale-normalization has already been explained in Section 4.3.1 while we explain rest of the steps in the following subsections.

4.4.1 Features Extraction

As described in Section 4.3, our main features of interest for gestures detection are pose and hand images. The concept of augmented pose for scale-normalization has been detailed in Section 4.3.1. For dynamic gestures detection, velocity and acceleration vectors from 8 upper-body joints, which contain information about the dynamics of motion, are also appended to the pose vector $\mathbf{x_n}$ to form a new 129 components augmented pose $\mathbf{x_{dyn}}$. Inspired by [84], joint velocities and accelerations are computed as first and second derivatives of scale-normalized joint positions respectively:

$$\delta \mathbf{p}^{(i)}(t) \approx \mathbf{p}^{(i)}(t+1) - \mathbf{p}^{(i)}(t-1) \tag{4.7}$$

$$\delta^2 \mathbf{p}^{(i)}(t) \approx \mathbf{p}^{(i)}(t+2) + \mathbf{p}^{(i)}(t-2) - 2\mathbf{p}^{(i)}(t) \tag{4.8}$$

The obtained velocity and acceleration values are normalized by the frame-rate at which videos are recorded to scale the values with respect to time before being appended in the augmented pose vector.

For every center frame of the selected window, scale-normalized augmented pose vectors $\mathbf{x_{dyn}}$ (as explained in 4.3.1) plus extracted left $\mathbf{i_l}$ and right $\mathbf{i_r}$ hands cropped image vectors respectively (as explained in Section 4.3.2) are appended in three individual arrays. Once features from all frames of a video are extracted, these arrays are written in a *.h5* file on the disk.

4.4.2 Label-wise Sorting and Zero-Padding

The videos in *Chalearn 2016 Looking at People isolated gesture recognition* are randomly distributed. Once the features of interest are extracted and saved in *.h5* files, we sort them with respect to their labels. Meanwhile, it is natural to expect videos (frames sequences, now features arrays) in the dataset to be of discrete frame lengths. Thus we propose to symmetrically pad sequences with zeros, or trim longer ones in the same manner, to limit their size across all gestures depending on their original length. The average video length in this dataset is 32 frames while we fix the length of each sequence to 40 frames in our work. If length of a sequence is less than 40, we pad zeros symmetrically in start and end of the sequence. Alternatively, if the length is greater than 40, symmetric trimming of the sequence is performed. Once the lengths of sequences are rectified (padded or trimmed), we append all corresponding sequences of a gesture label into a single array and save it in another *.h5* file on the disk. Therefore, at the end of this procedure, we will be left only with 249 *.h5*

files corresponding to 249 gestures in *Chalearn 2016 Looking at People isolated gesture recognition* dataset.

Label-wise sorting operation, as presented in this section, is only necessary if we want to train a network on selected gestures (as we will explain in Section 4.6). Otherwise, creating only a ground-truth label array should suffice at this stage.

4.4.3 Train-Ready Data Formulation

To obtain train-ready data, arrays in all gestures *.h5* files are concatenated and normalized either (1) by scaling each feature between zero and one with respect to its range in the dataset or (2) through standardization of each feature to zero mean and unit variance.. Meanwhile, a ground-truth label array is composed by sequentially appending a shared array with the number of elements and gesture class label corresponding to the *.h5* file being processed. Normalization of hand images is accomplished by performing element-wise division of images with maximum pixel intensity value i.e., 255.

In this research, both methods are utilized and the respective use cases will be detailed in Section 4.6. It has to be noted that if features are normalized to zero mean and unit variance, *Rectified Linear Unit (ReLU)* activation function should be avoided in the immediate next layer to the input and *sigmoid* activation should be preferred. Nevertheless, standardization with zero mean and unit variance is a preferred choice as it handles the outliers in the dataset properly.

4.5 CNN-LSTM Model

In this section, we briefly describe the functionality of convolutional neural networks (CNNs) followed by a concise description of long short-term memory (LSTM) networks. Later in this section we present the details of our proposed CNN-LSTM for gesture recognition.

4.5.1 Convolutional Neural Networks

Convolutional neural networks have lately been successful in static image recognition problems such as MNIST, CIFAR and ImageNet large-scale visual recognition challenges (ILSVRC) [82] with the state-of-the-art results even surpassing human-level performance [186]. Contrary to the traditional feed-forward neural networks, convolutional methods are robust against shift, scale and distortions in image classification problems due to their inherent properties of having local receptive fields, shared weights and spatial or temporal sub-sampling [60].

Local receptive fields allow extraction of elementary visual features such as oriented edges, end-points and corners from a small neighborhood of each element in the previous layer through convolutions of fixed sized kernels. Shared weights ensure that a learned elementary feature detector, which is important in a spatial location (x_1, y_1), is used to extract similar features in another location (x_2, y_2) in an image. Units in a layer are grouped in planes within which each of them shares the same set of weights. The set of outputs of the units in such a plane is known as a *feature map*. The relative locations of detected features are more important than their exact positions. Thus the spatial resolution of the feature maps are reduced to curtail the precision with which the position of distinctive features are encoded. This is achieved by sub-sampling the feature maps which is performed by local averaging or max pooling of the layers hereby reducing sensitivity of the output to shifts or distortions. A "bi-pyramid" is formed by successively alternating convolutional and sub-sampling layers such that in each layer, number of feature maps are increased while their spatial resolution is decreased. The feature maps in the later layers are then connected to fully-connected layers followed by an output layer with neurons corresponding to the number of output classes and a squashing activation function like *sigmoid* or *softmax*. The weights are learned through back-propagation strategy in an "end-to-end" fashion.

4.5.2 Long Short-Term Memory Networks

A standard recurrent neural network (RNN) model the evolution of information in a given input sequence $\mathbf{x} = (x_1, x_2, \ldots, x_T)$ by computing hidden vector sequence $\mathbf{h} = (h_1, h_2, \ldots, h_T)$ where $h_t \in \mathbb{R}^N$ is the hidden state with N hidden units, and the output vector sequence $\mathbf{y} = (y_1, y_2, \ldots, y_T)$ through the following recurrent equations [81, 90]:

$$\begin{aligned} h_t &= f(W_{ih}x_t + W_{hh}h_{t-1} + b_h) \\ y_t &= g(W_{ho}h_t + b_o) \end{aligned} \tag{4.9}$$

where the terms W denote the weight matrices (e.g., W_{ih} connects inputs to the current hidden layers, while W_{hh} represents the connections between previous and the current hidden layers). The terms b denote bias vectors (e.g., b_h is the hidden bias vector) while f and g are element-wise non-linear activation functions, such as sigmoid or hyperbolic tangent. Lately, RNNs have demonstrated success in speech recognition [187], language modeling [188] and text generation [189] tasks. However, it can be difficult to train ordinary RNNs for problems that require learning of long-term temporal dynamics likely due to the vanishing and exploding gradient problems [81, 90]. These complications emerge while propagating gradients down through multiple layers of the recurrent network corresponding to the sequence length.

LSTMs [81] on the contrary, exploit memory cells incorporating different gates that enable the network to maintain, forget or update the hidden states given new context information through learned weights. The hidden layer of LSTM is computed as follows [81, 90]:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ g_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{4.10}$$

where σ represents logistic sigmoid function, operator $\odot$ denotes Hadamard product, $i_t \in \mathbb{R}^N$ is input gate, $f_t \in \mathbb{R}^N$ is forget gate, $o_t \in \mathbb{R}^N$ is output gate, $g_t \in \mathbb{R}^N$ is input modulation gate, $c_t \in \mathbb{R}^N$ is memory cell and $h_t \in \mathbb{R}^N$ is a hidden unit. These additional components enables LSTM to learn complex and long-term temporal dependencies in a wide variety of sequence learning tasks. Improved performance has been reported [90] by stacking LSTM blocks such that the hidden state $h_t^{(l-1)}$ of the LSTM in layer $l-1$ is given input to the LSTM in layer l.

4.5.3 CNN-LSTM for Gesture Recognition

As mentioned in Section 4.1, a CNN-LSTM network is devised to model spatio-temporal dependencies inspired by [90], for the recognition/classification of dynamic gestures. An overall representation of the proposed network is illustrated in Figure 4.7. As explained in Section 4.3, our spatial attention module extracts augmented pose and hands of the user. This multi-modal input is fed into the proposed CNN-LSTM network which functions as a *many-to-one* classifier. The weights of time-distributed fully-connected layer are unique for each time-step. The hidden and memory (cell) states of the LSTM blocks are aggregated in time step-by-step until the last input frame is processed. The output of last LSTM block is then sent to a fully-connected dense layer followed by a *softmax* layer to provide gestures class labels probabilities. The hand images are first passed through CNN blocks which actually are Inception V3 networks pre-trained on ImageNet dataset, and fine-tuned on *opensign* static hand gestures dataset [23]. Prior fine-tuning of Inception V3 on augmented backgrounds hand gestures dataset allows efficient and robust extraction of distinctive hands features. Image embeddings of size 1024 elements are provided as output by the CNN-block. Multiple modalities i.e., 129-components standardized augmented pose and image embeddings of 1024 elements for each hand, are fused in intermediate layers of the network.

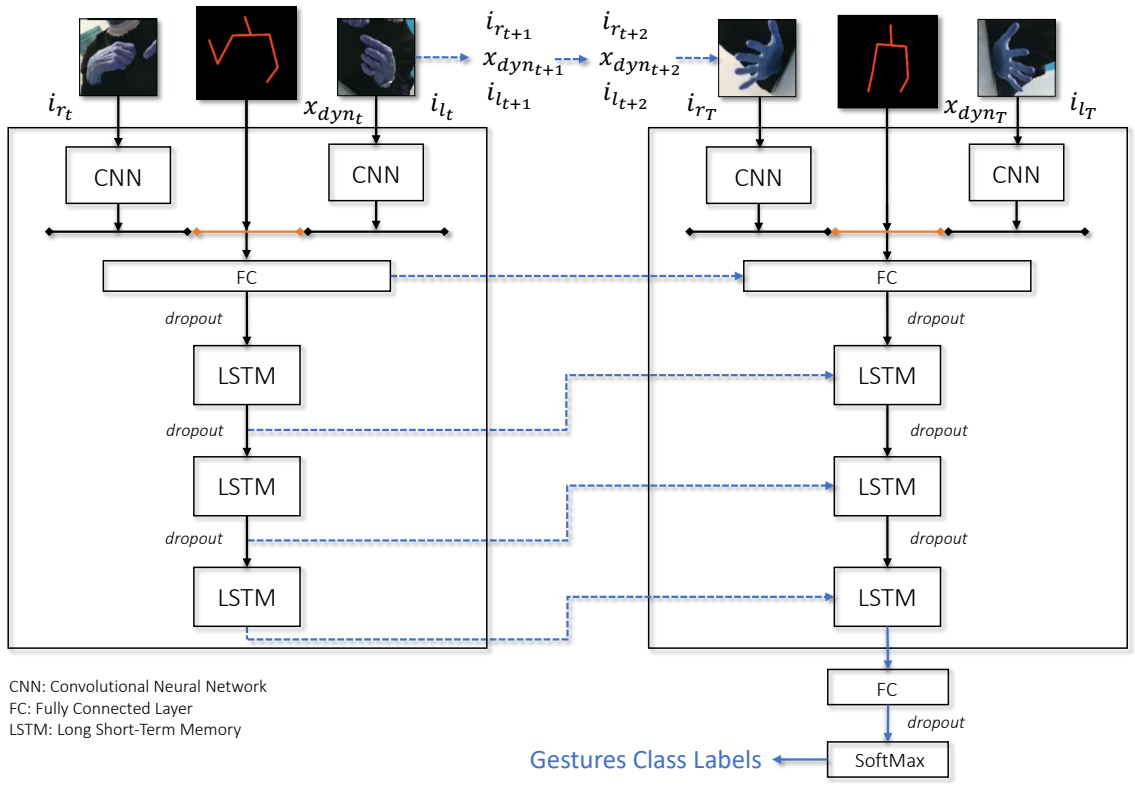


Figure 4.7 Illustration of the proposed CNN-LSTM network for dynamic gestures recognition

Dropout strategy [190] is employed between successive layers to prevent over-fitting and improved generalization of the network. Moreover, batch-normalization technique [191] is exploited that accelerates training of the deep networks.

4.6 Training

The proposed network is trained on a computer with Intel© Core i7-6800K (3.4 GHz) CPU, dual Nvidia GeForce GTX 1080 GPUs and 64 GB system memory. Rest of the details are explained in the following subsections.

4.6.1 Multi-Stage Training

A multi-stage training strategy is proposed that may facilitate to train multi-modal networks faster on large-scale video activity detection datasets on systems with limited GPU memory. With an assumption that the CNN blocks are pre-trained and their weights are no longer expected to adapt to the input frame sequence, the network training can be performed separately in two stages.

First, hands images are passed only through the CNN blocks and embeddings arrays are stored on the disk. Then augmented pose and hand image embeddings are fed into rest of the network, which performs early concatenation of the inputs before passing it to the stacked LSTM blocks. This two stage strategy requires less GPU memory in each step, thus a larger batch-size can be utilized for quicker processing of the data. This strategy however, does not allow end-to-end training of our network.

4.6.2 Training Chalearn Dataset

As mentioned in Section 4.2.1, *Chalearn 2016 isolated gestures recognition* dataset has 35,876 videos in the provided train set while only top 47 gesture labels (arranged with respect to number of samples in descending order) contain 34% of all videos in this set. Average number of samples in top 47 gestures is 260 contrary to 144 for all 249 labels. Thus, 12210 videos of 47 gestures are utilized to pre-train our CNN-LSTM with a validation split of 0.2. The learned parameters of this network are exploited to initialize weights for model training to classify all 249 gestures with 35,930 train samples and mutually exclusive set of 6000 validation videos. *Adam* optimizer [192] is exploited for training our CNN-LSTM network.

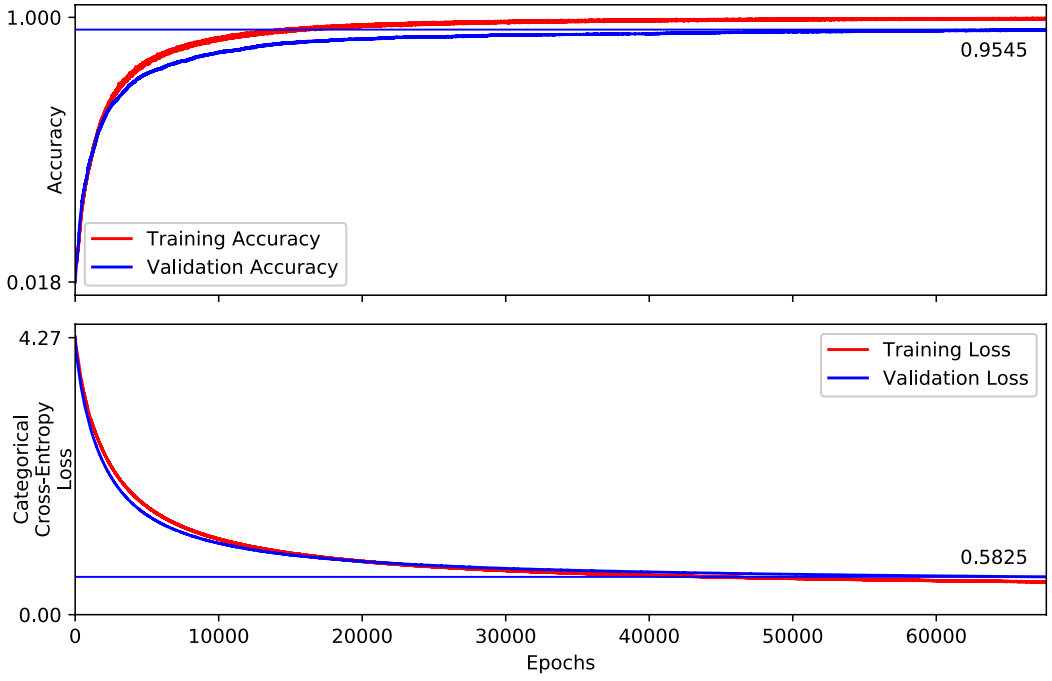


Figure 4.8 Illustration of the training curves for 47 top gestures from Chalearn 2016 isolated gestures recognition dataset. The obtained model is utilized for weights initialization of the network for classification of all 249 gestures.

4.6.3 Training Praxis Dataset

Training our CNN-LSTM network on *Praxis cognitive assessment* dataset is rather a straight forward procedure. As mentioned in Section 4.2.2, this dataset has 1247 videos in total for 14 correctly performed dynamic gestures. The samples are augmented by applying horizontal mirror operation (1) to double the sample size and (2) to subdue the influence of dominant hand ambiguity. 501 videos are randomly extracted as test set while 1993 samples are utilized for training with a validation split of 0.2. Taking the small size of this dataset into consideration, hyper-parameters of the network are adapted to avoid over-fitting.

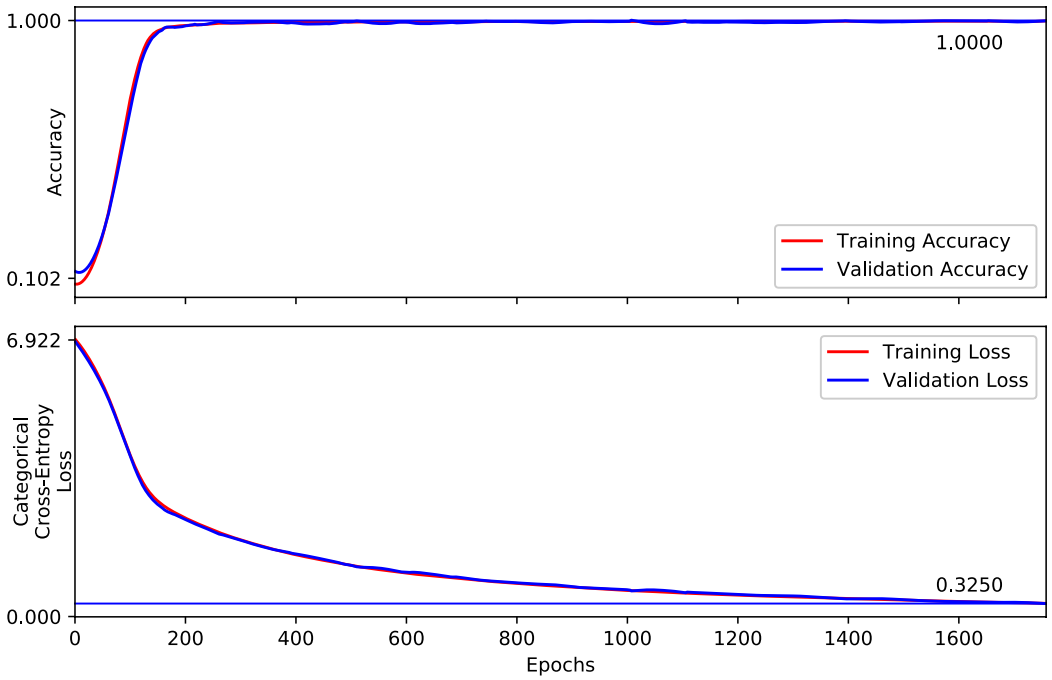


Figure 4.9 Training curves of Praxis gesture dataset.

4.7 Results

For *Chalearn 2016 isolated gestures recognition* dataset, the proposed network is initially trained on 47 gestures with a low learning rate of 10^{-5}. After approximately 66,000 iterations, validation accuracy of 95.45% is obtained as illustrated in the training curves plot in Figure 4.8. The learned parameters for 47 gestures are employed to initialize weights for complete data training with 35,930 train videos and 6000 validation samples for 249 gestures as detailed in Section 4.2.1. The network is trained in four phases. Weights initialization is performed, inspired by *transfer learning* concept of deep networks, by replacing the classification layer (with *softmax* activation function) by the same with output number of

neurons corresponding to the number of class labels in the dataset. In our case, we replace the *softmax* layer in the trained network for 47 gestures plus the *fully-connected (FC)* layer immediately preceding it. The proposed model is trained for 249 gestures classes with a learning rate of 1×10^{-3} and a decay value of 1×10^{-3} by *Adam* optimizer.

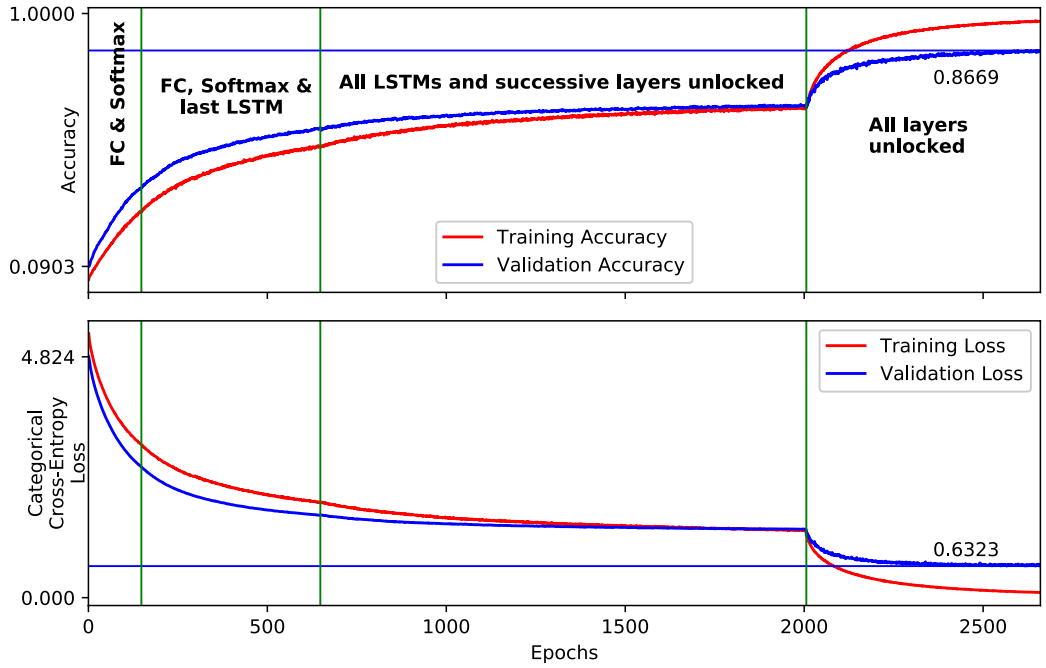


Figure 4.10 Training curves of the proposed CNN-LSTM network for all 249 gestures of Chalearn 2016 isolated gestures recognition dataset. The network is trained in four phases which can be distinguished by the vertical lines in the plot.

With the weights initialized, the early iterations are performed with all layers of the network locked except the newly added *FC* and *softmax* layers. As the number of epochs increases, we successively unlock the network layers from the bottom (deep layers). In the second phase, network layers until the last LSTM block are unlocked. All LSTM blocks and consequently complete model is unlocked in the third and fourth phase of training respectively. By approximately 2700 epochs, our CNN-LSTM achieves 86.69% validation accuracy for all 249 gestures and 86.75% test accuracy on a set of 6000 test samples, which are the state-of-the-art performances on this dataset. The prediction time for each video sample is 57.17 *ms* excluding pre-processing of the video frames, thus continuous online dynamic gesture detection for a human-robot interaction experiment is expected to be real-time. The training curve of the complete model is shown in Figure 4.10 while the confusion matrix/heat-map with evaluations on test set is shown in Figure 4.11. Our results are compared with the reported state-of-the-art in Table 4.1.

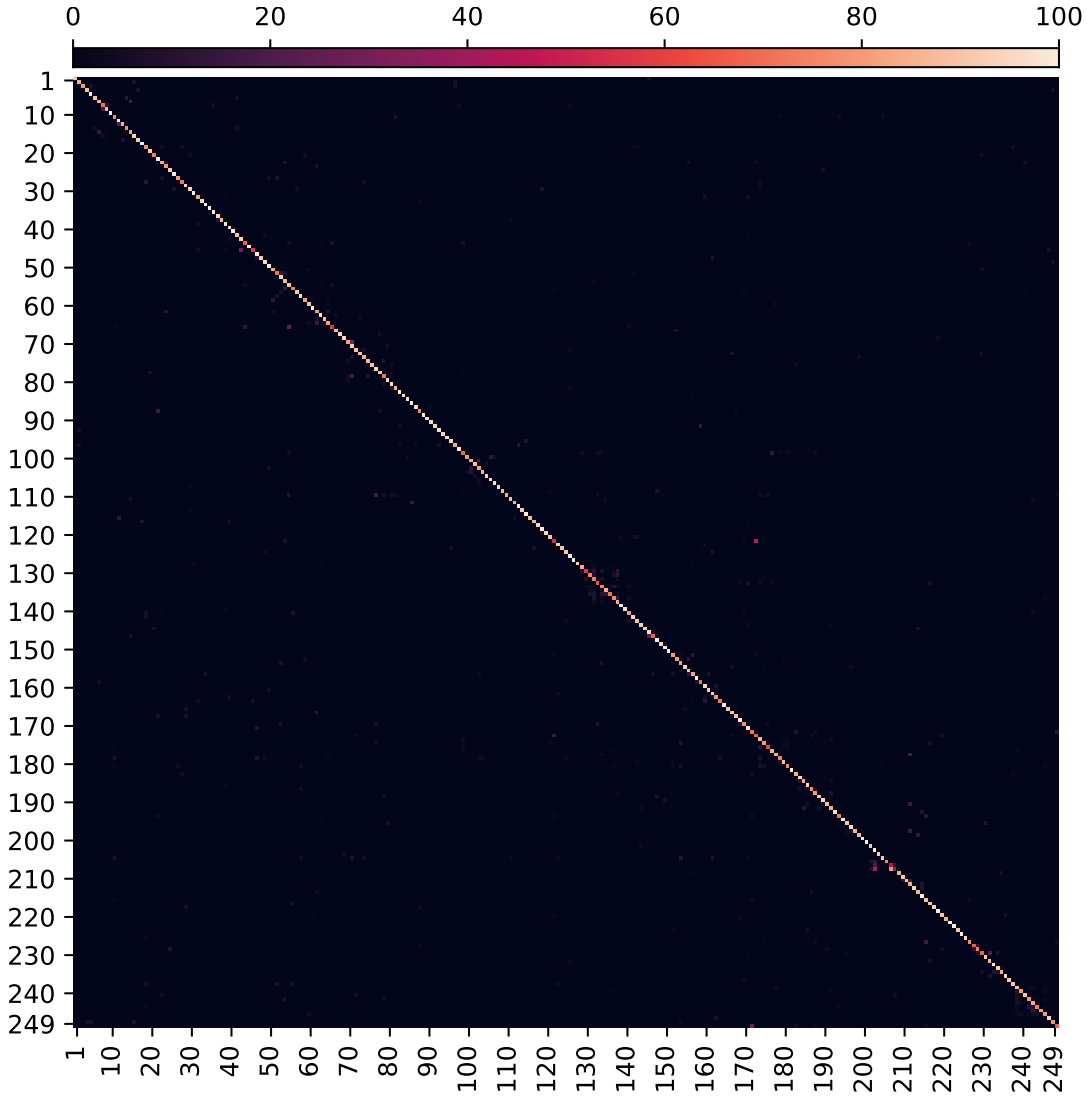


Figure 4.11 Illustration of the confusion matrix/heat-map of the proposed model evaluated on test set of Chalearn 2016 isolated gestures recognition dataset. It is evident that most samples in the test set are recognized with high accuracy for all 249 gestures (diagonal entries, 86.75% overall).

Inspecting the training curves, we observe that the network is progressing towards slight over-fitting in the fourth phase when all network layers are unlocked. Specifically the first *time-distributed FC* layer is considered the culprit for this phenomenon. Although, we already have a dropout layer immediately after this layer with dropout rate equaling 0.85, we skip to further dive deeper to rectify this. However, it is assumed that substitution of this layer with the strategy of *pose-driven temporal attention* presented in [17] or with the *adaptive hidden layer* proposed in [194], may help to reduce this undesirable phenomenon and ultimately may improve results further.

System	Valid %	Test %
Mazhar *et al.* (ours)	**86.69**	**86.75**
FOANet [105]	80.96	82.07
Miao *et al.* [100] (ASU)	64.40	67.71
SYSU_IEEE	59.70	67.02
Lostoy	62.02	65.97
Wang *et al.* [193] (AMRL)	60.81	65.59

Table 4.1 Comparison of the reported results with ours on Chalearn 2016 isolated gestures recognition dataset. The challenge results are published in [1]. Order of the entries is set with respect to the test results.

For *Praxis* dataset, the optimizer and values of learning rate and decay, are same as that for *Chalearn* dataset. The hyper-parameters including number of neurons in *FC* layers plus hidden and cell states of LSTM blocks are (reduced) adapted to avoid over-fitting. The model has obtained 99.6% test accuracy on 501 samples. The training curves of this dataset are presented in Figure 4.9 while the confusion matrix for test data is shown in Figure 4.12. Results comparison on this dataset with the state-of-the-art is shown in Table 4.2.

System	Accuracy % (dynamic gestures)
Mazhar *et al.* (ours)	**99.60**
Negin *et al.* [2]	76.61

Table 4.2 Comparison of reported dynamic gestures detection results on Praxis gestures dataset. The author in [2] also achieved best results with a CNN-LSTM network.

4.8 Conclusion

The proposed strategy to recognize dynamic gestures provides state-of-the-art results on large-scale *Chalearn 2016 isolated gestures recognition* dataset and a small *Praxis* gesture dataset. Our spatial attention mechanism, which focuses on upper-body pose for large-scale body movements of the limbs plus on hand images for subtle hand/fingers movements has out-scored the existing approaches on the datasets which utilize full image frames or only single modality like pose or hands. The parameters that estimate the scale-factor of user and the size of bounding boxes of his/her hands are learned from a static hand gestures dataset recorded with Kinect V2. This enables to exploit only RGB images for gestures recognition

tasks thus the proposed strategy can be implemented with systems that only contains single RGB cameras.

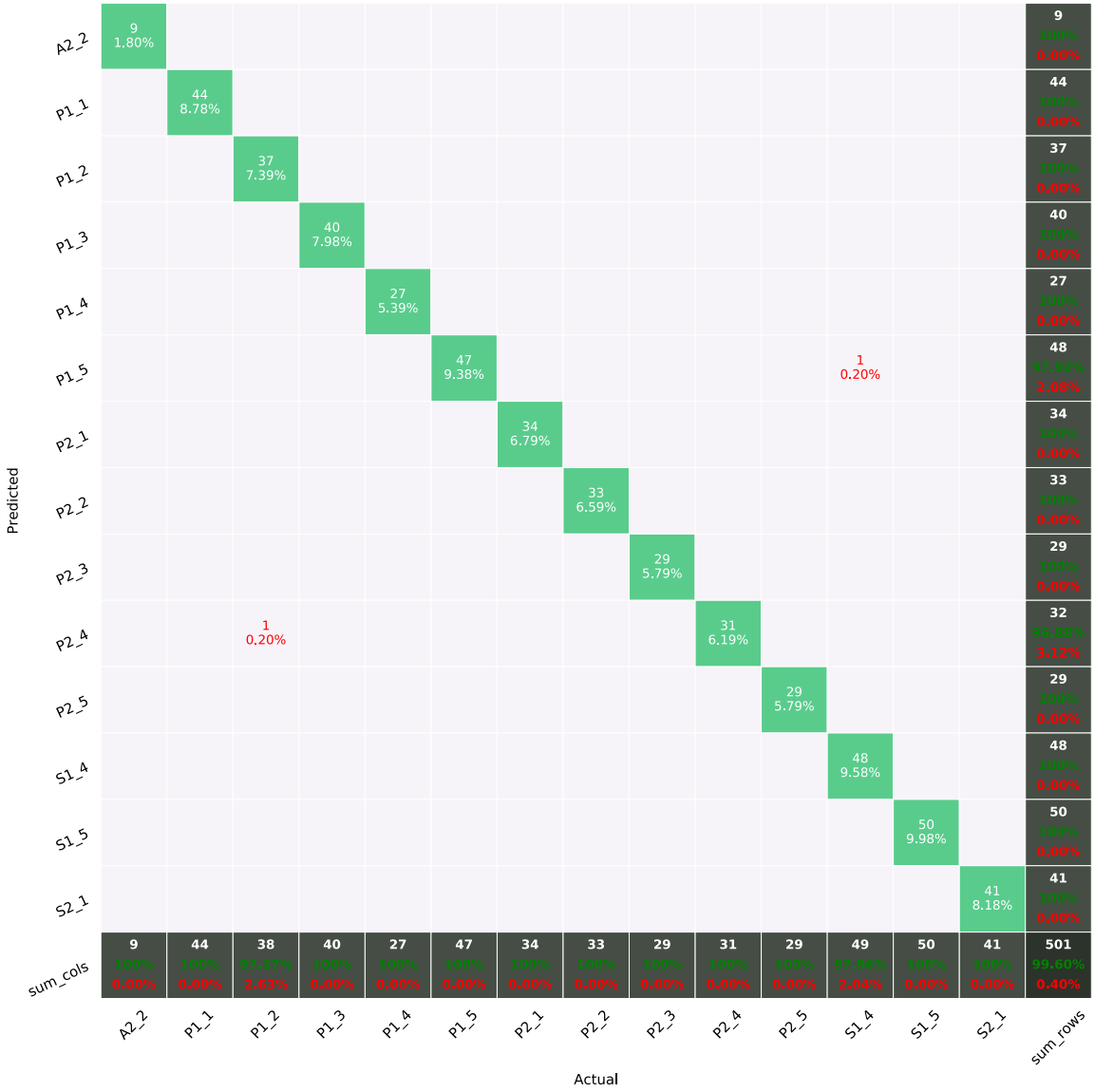


Figure 4.12 Illustration of the confusion matrix for Praxis gestures dataset evaluated on 501 (original and mirrored) test samples. The diagonal values represents number/contribution of videos of the corresponding class in the dataset. Accuracy values for individual gesture labels are displayed on the bottom row and the last column.

The presented weight initialization strategy facilitated parameters optimization for all 249 gestures when the number of samples among the classes varied substantially in the dataset. Class recognition is performed on isolated gestures videos executed by a single individual in the scene. However, we plan to extend this work on continuous gestures recognition to

complement our previous work on human-robot interaction which worked through static hand gestures recognition. This can be achieved in one way by developing a binary motion detector to detect start and end instances of the gestures. Although a multi-stage training strategy is presented, it is desired to develop an end-to-end training approach for potential online learning of new gestures in the system.

Chapter 5

3D Human Pose Estimation

Optical skeleton motion capture has been extremely beneficial in applications such as character animations for movies and games, sports, biomechanics and medicine. Lately, marker-less motion capture methods are explored by the computer-vision community to overcome usability constraints of the commercial systems. Introduction of depth cameras like Microsoft Kinect, brought in novel real-time full-body pose estimation strategies for applications like motion guided game character control, self immersion in virtual reality and human-computer/robot interaction.

In this chapter, a hybrid method for 3D human pose estimation is proposed, which minimizes the distance between joint coordinates obtained from a discriminative 2D pose extractor, and virtual 2D camera projection of a 3D kinematic model of a human-body, through optimization of an objective function.

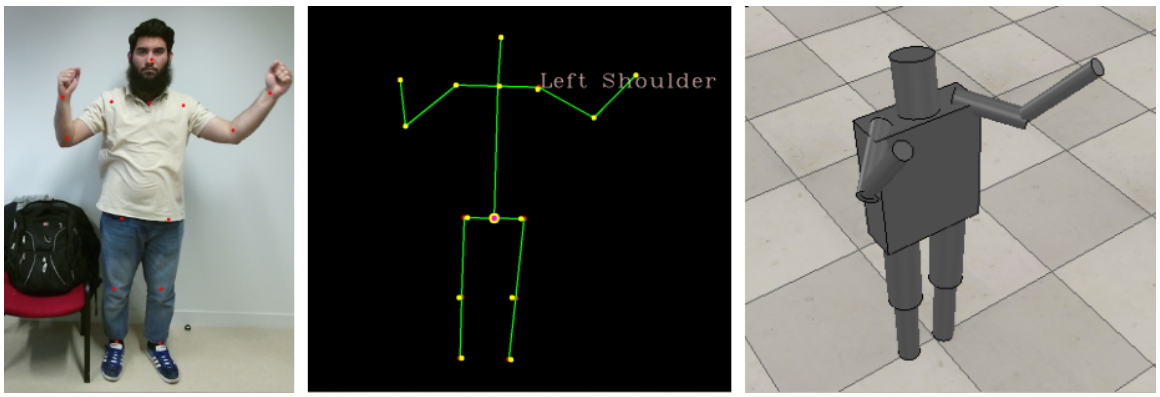


Figure 5.1 An illustration of our proposed 3D pose estimation strategy in operation.

5.1 Our Contributions

- A novel strategy to estimate 3D human pose from a monocular camera is proposed.
- Joint angles of a 3D human kinematic model are optimized such that its 2D projection from a virtual camera matches with joint coordinates on image plane obtained from a discriminative 2D pose estimator i.e., *openpose*.
- Our *pose estimation module* is integrated with *openpose* through *nanomsg* socket library which allows online 3D human pose estimation.
- The proposed system is also integrated with a robot simulator *V-REP* for visualization and interaction experiments.

The proposed strategy to estimate 3D human pose from a monocular camera is described in the following sections.

Joints	X		Y		Z	
	min	*max*	*min*	*max*	*min*	*max*
Trunk	-0.3491	0.3491	-0.1745	0.3491	-1.5708	1.5708
Neck	-1.0472	1.0472	-0.1745	0.7854	-	-
LS	-2.3562	2.3562	0.0	2.3562	-2.3562	0.7854
LE	0.0	1.5708	-	-	-	-
RS	-2.3562	2.3562	0.0	2.3562	-0.7854	2.3562
RE	0.0	1.5708	-	-	-	-
LH	-1.0472	1.0472	0.0	2.7925	-	-
LK	-	-	0.0	2.3562	-	-
RH	-1.0472	1.0472	0.0	1.5708	-	-
RK	-	-	0.0	2.3562	-	-

Table 5.1 Joint limits in radians along *X, Y* and *Z axes*. Longer joint names are abbreviated such as *LS* for *Left Shoulder*. Same applies for *Elbow, Hip* and *Knee* joints.

5.2 Human Kinematic Model

A human kinematic model is formulated which possesses 10 body parts with 14 joints key-points (with distinct degrees of freedom) while the overall posture/orientation of the model is determined by a set of 19 angles vector denoted by $\mathbf{m}$. The names and assumed limits of these joints are presented in Table 5.1. 3D positions of these joints are represented by a vector $\mathbf{p_{3D}}$, which can be deduced through *forward kinematics* computations given lengths of the bones/model-parts and joints angles.

5.3 2D Projection of Human Model

As stated in Section 5.1, the general scheme of the proposed method is to estimate a set of angles $\mathbf{m}$ that represents minimum discrepancy between joint coordinates obtained from *openpose* and 2D projection of the kinematic model from a virtual camera through the proposed objective function.

To obtain $\mathbf{p_{3D}}$, we exploit a rigid-body dynamics library *RBDyn*[1] which performs *forward kinematics* operations given the bone lengths and current state of vector $\mathbf{m}$. The key-points in 3D are then projected onto a virtual image plane by the following camera projection equations:

$$\begin{aligned} \mathbf{p_{2D}} &= \mathbf{C}\mathbf{p_{3D}} \\ \mathbf{C} &= \mathbf{K}[\mathbf{R}|\mathbf{t}] \\ \mathbf{K} &= \begin{bmatrix} f_x & s & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.1}$$

where $\mathbf{C}$ is camera matrix, which can be decomposed into camera intrinsic and extrinsic parameters matrices $\mathbf{K}$ and $[\mathbf{R}|\mathbf{t}]$ respectively and $\mathbf{p_{2D}}$ is a 14 points vector representing 2D projection of human kinematic model 3D key-points ($\mathbf{p_{3D}}$) from a virtual camera on its image plane. The intrinsic parameters from the camera which is used to acquire images of the user can be substituted/replicated in the $\mathbf{K}$ matrix. Extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ are manually tuned such that the 2D obtained projection of human model lies upright and the virtual camera is located at the neck level of the human model. Moreover the distance of the virtual camera is set such that the 2D projection of the model lies well inside the virtual image frame even when all the limbs are stretched to their extremes.

5.4 Scale and Position Normalization of the Skeleton

It is imperative to normalize the obtained 2D skeleton from *openpose* to subdue the influence of scale (distance) and position of the user in the image. Position normalization can be performed by subtracting the coordinates of root joint (*Hip Center in this case*) from rest of the coordinates. *Openpose* provides a 25 joints output with *BODY_25* model. Facial coordinates i.e., ears and eyes plus feet joint coordinates are ignored thus only 15 jointsx 2D coordinates, stored in a vector $\mathbf{p^o}$, are selected in this work. As mentioned earlier, $\mathbf{p_{2D}}$ is a 14 points vector with *Hip Center* point absent. Thus a line is drawn between *Left Hip* and *Right*

[1] https://github.com/jrl-umi3218/RBDyn.git

Hip joints of $\mathbf{p_{2D}}$ and mid-point of this line is extracted, which is assumed as *Hip Center* point (root joint). Accordingly, position normalization of $\mathbf{p^o}$ is performed as stated, and root joint coordinates of the model projection $\mathbf{p_{2D}}$ are added to the normalized $\mathbf{p^o}$ key-points. This super-impose the obtained skeleton over $\mathbf{p_{2D}}$ anchored at their *Hip Centers*.

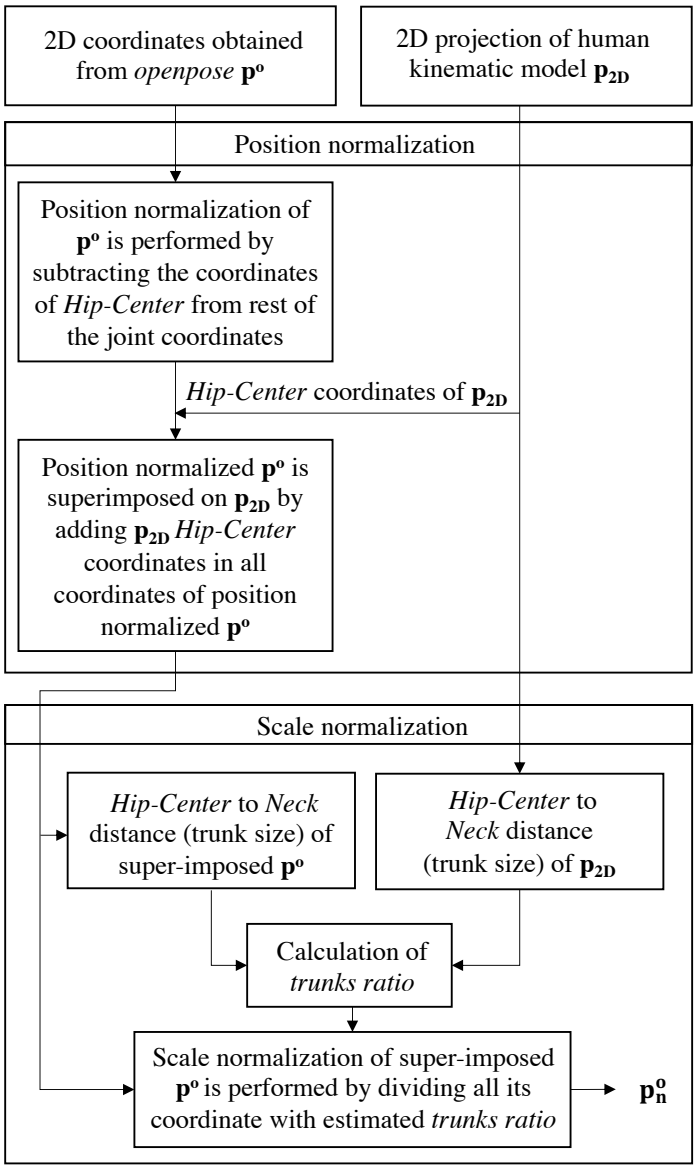


Figure 5.2 Illustration of position & scale-normalization procedure.

For scale-normalization, distance between *Hip Centers* and *Neck* coordinates (approximate trunks sizes) of both skeletons ($\mathbf{p_{2D}}$ and position normalized $\mathbf{p^o}$) are estimated. The coordinates of position normalized $\mathbf{p^o}$ are multiplied with a scale-factor, which is obtained by calculating the ratio of estimated trunks sizes to get position and scale normalized skeleton

$\mathbf{p_n^o}$. This however enforces a strong assumption that the user always stays upright while his pose is being estimated which nevertheless can be true for many human-robot interaction tasks. The overall scale and position normalization procedure is shown in Figure 5.2.

5.5 Formulation of Objective Function

The objective function that our optimization algorithm minimizes is given as follows:

$$\begin{aligned} \mathbf{m}^* &:= \arg\min_{\mathbf{m}} E(\mathbf{m}) \\ E(\mathbf{m}) &= \left\| \mathbf{C} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} FK(\mathbf{m}) - (\mathbf{u}, \mathbf{v})^T \right\|_2 \end{aligned} \tag{5.2}$$

where operator *FK* represents *forward kinematics* operation which returns rotation and position output of the body links. However, only 3D position coordinates in the return values are of our interest, so the rotation parameters are zeroed. $(\mathbf{u}, \mathbf{v})$ represents position and scale normalized 2D joint coordinates $\mathbf{p_n^o}$ as described in the previous section. Our optimization function searches for optimal model parameters $\mathbf{m}^*$ that minimizes the discrepancy $E(\mathbf{m})$ between 2D projections of the human kinematic model $\mathbf{p_{2D}}$, and normalized 2D joint coordinates obtained from *openpose* $\mathbf{p_n^o}$.

5.6 Experimental Setup

The proposed *pose-estimation module* is integrated with *openpose* in a configuration of inter-process distributed network through a socket library *nanomsg*. This allows online frame-by-frame pose-estimation which can be extended to work in real-time for human-robot interaction tasks. Our experimental setup is also connected to a robotic simulator i.e., *V-REP* via its *remote API*. Thus the 3D model can be visualized as it acquires new poses in real-time from optimized parameters.

5.7 Results

As soon as *openpose* extracts 2D human pose $\mathbf{p^o}$, it transfers pose array to the proposed *pose estimation module*. Each optimization cycle that estimates model parameters $\mathbf{m}^*$ takes approximately 14 *ms* of computational time to converge for each input $\mathbf{p_n^o}$ on Intel© Core

i7-7700HQ CPU. Qualitative analysis of the output reveals that the proposed pose estimation approach delivers promising results.

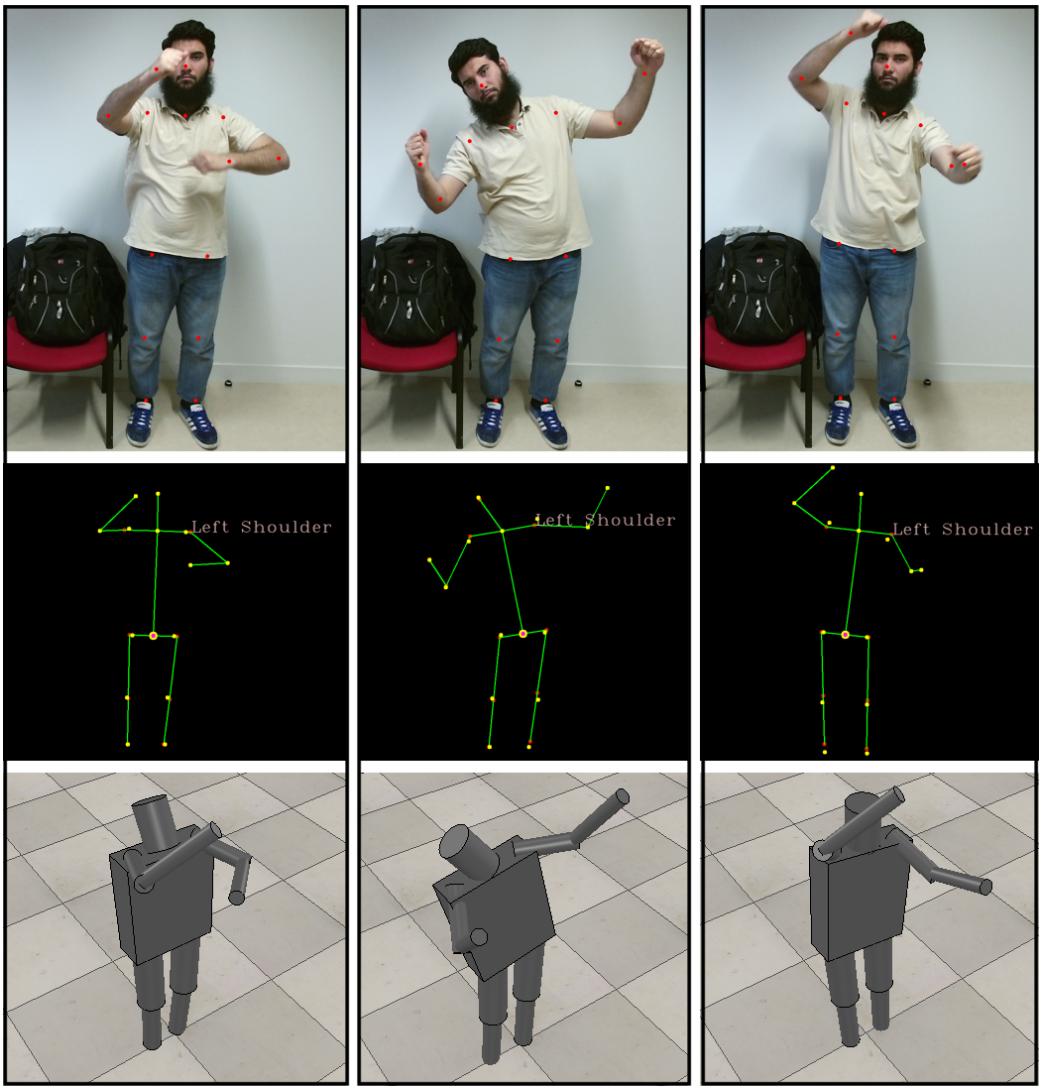


Figure 5.3 Illustrations of 3D pose estimations with movements mainly on upper-limbs. The skeletons in the second row with green lines (red dots) are 2D projections of the kinematic human model, while yellow dots in the same images are position and scale normalized 2D pose obtained from *openpose*. The optimization problem solves to minimize the discrepancy between these two skeletons to estimate 3D human pose.

A pose-initialization step may be required to initiate convergence. The estimator reaches its minimum within 5 iterations on a gesture with arms stretched horizontally. However, due to total lack of depth information either from the sensor or through learning from 3D pose datasets, the proposed pose-estimator occasionally falls into local minima specially when limbs move in *sagittal plane*. This phenomenon however is less evident in upper-limbs while

lower limbs are affected largely by this issue. Figure 5.3 illustrates the results of our pose estimation strategy with movements mainly on upper-limbs.

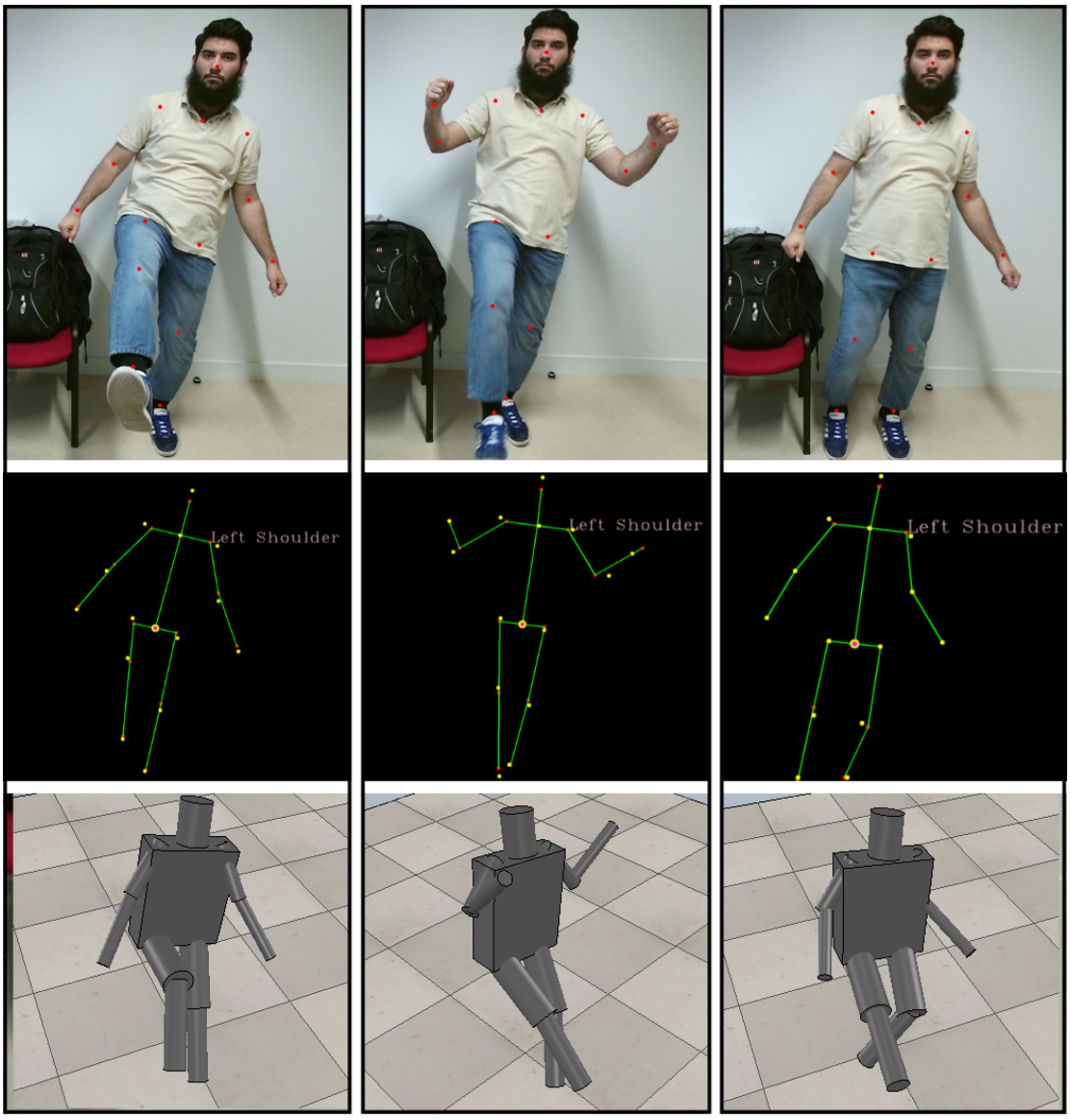


Figure 5.4 3D pose estimations with full-body articulations. It can be seen that proposed strategy fails to solve depth ambiguity for (lower) limbs movements in *saggital* plane. Also the optimization falls into local minima in some cases as shown.

Since depth ambiguity in the lower-limbs is more noticeable, this can corrupt proper estimation of upper-limbs pose as well if a combined objective function is used. Therefore 3 individual modes of operation are devised in our work with individual objective functions (1) upper-body pose estimation, (2) lower-body pose estimation and (3) full-body pose-

estimation. This helps to segregate the estimation tasks hence the problems can be dealt individually to improve results.

5.8 Conclusion

The proposed strategy for 3D human pose estimation from a monocular camera demonstrate promising results. It is suspected that the errors related to lower-limb pose estimation are due to (1) proposed scale-normalization strategy based on trunk sizes ratio and (2) lack of learned depth perception. A learning based method as described in Section **??** can be employed to address both of these issues, however these potential solutions are not implemented to compare results with the strategy we presented in this chapter. Moreover, quantitative analysis of the results with a ground-truth pose from systems like *MoCap* is not performed and is intended to be done in the future work.

Chapter 6

Discussion and Conclusion

In this thesis, we explored deep learning solutions for vision-based human gestures detection problem intended for human-robot interaction scenarios. We proposed novel solutions to detect static hand gestures robustly as well as to recognize dynamic gestures with the state-of-the-art results. Moreover, the problem of 3D human pose estimation from monocular cameras is also addressed in this thesis.

Achievements

We started with static hand gestures recognition problem. Kinect V2 was opted as the main sensor for static hand gestures problem. Initially, a simple CNN architecture was designed to recognize four static hand gestures. One of the novelties in this part of our research i.e., background substitution, actually originated from this early design of our detector. Previously, hands were segmented with the help of Kinect V2 depth map and hand image background were replaced with gray shades. Moreover, the gestures were performed only by a single user. Later in this work, we presented the idea of background substitution with random pattern and indoor architecture images. Thus a new dataset was developed named *opensign* with 10 static hand gestures taken from American Sign Language performed by 10 volunteers. However, such a method made gesture recognition a complex problem for a simple CNN architecture. Thus, the concept of *transfer learning* was opted. We fine-tuned Inception V3 CNN, which is already trained on ImageNet, on background substituted hand images created through our dataset *opensign*. We trained this network and obtained validation accuracy of 96.4% on 10 static hand gestures. Thus a robust background invariant static hand gestures algorithm has been developed.

To extend our work on systems with monocular cameras, we subsequently focused on strategies which rely only on pure RGB images. We delved deeper into dynamic gestures

detection problem and proposed a spatial attention-based multi-modal CNN-LSTM strategy which recognizes upper-body dynamic gestures from pure RGB inputs. Three separate neural networks were developed and trained to estimate depth of user's neck (for scale normalization of the skeleton) as well as sizes of hands bounding boxes through position-normalized upper-body 2D pose. These networks were trained on depth data obtained from our static hand gestures dataset *opensign*. Once the pose is "position and scale normalized", an augmented pose vector is formalized with estimated velocity and acceleration parameters (in pixel domain) from a window of input frames. Moreover, the spatial attention mechanism, takes raw pose information to localize hands in the scene. Extracted hand locations combined with the estimated size of bounding boxes, spatial attention module focuses on augmented pose and hand cropped images for dynamic gestures detection. We trained our network with the proposed attention strategy on *Chalearn 2016 isolated gestures recognition* dataset and obtained the state-of-the-art performance with test accuracy equals to 86.75% outperforming all reported results on this dataset. We also trained our network on *Praxis cognitive assessment* dataset on correctly performed dynamic gestures and obtained the state-of-the-art results on this dataset with 99.6% test accuracy.

Afterwards, we briefly explored the problem of 3D pose estimation from monocular cameras in this research. A hybrid strategy is proposed that optimizes an objective function to minimize the discrepancy between scale and position normalized 2D skeleton originally obtained from *openpose*, and virtual 2D projection of a human kinematic model. Although, the qualitative analysis of results from the proposed strategy demonstrated issues due to complete lack of depth perception either from the sensor or learned from any 3D dataset, the overall results were very promising.

To validate our (static) gestures detection algorithm, we developed an asynchronous inter-process distributed system for a real-time integration of *openpose*, *OpenPHRI* and our static hand gestures detection module. We mocked-up an industrial scenario and performed a robotic experiment with a Kuka LWR 4+ arm. *OpenPHRI* was used to control the robotic arm with embedded safety features complemented by the depth information obtained through Kinect V2 exploited in our static hand gesture detector. Asynchronous integration of different modules guaranteed real-time operation with overall frame-rate of 20 fps.

Limitations and Future Work

Our static hand gestures module has a limited vocabulary of gestures. Although we have developed scale normalization and hand bounding boxes estimation strategy from pure RGB images, it is yet to be implemented in our static hand gestures module. It also lacks the

capability to model/detect transitional gestures which will be among the features we would like to implement in the future.

The dynamic gesture detector we proposed, only classifies gestures performed by a single person in the camera frame. However the proposed method can be extended to recognize multi-person activities with fixed number of people expected to appear in the scene. In the future work, we plan to develop a real-time dynamic gesture detector which will allow us to integrate it into our proposed human-robot interaction framework shown in Figure 3.3. Moreover, temporal-attention mechanism is planned to be developed and combined with our spatial-attention strategy for more accurate results in dynamic gestures classification. An end-to-end network training strategy will be opted in the future work.

In our 3D human pose estimation work, two technical issues were suspected to be the reason of optimization falling in local minima, or inaccurate estimations which were: (1) scale normalization strategy driven by the ratio between 2D (pixel) lengths of trunks and (2) absence of depth perception from a specialized sensor or learned. The learning-based strategy to perform scale normalization proposed in Section **??** appears more suitable to be used instead of the one proposed in Section 5.4. This can also be extended to estimate approximate depth of each skeletal joint. Thus its implementation in 3D pose estimation algorithm is necessary before we quantify our results and expect it to work in real experiments. The integration of our proposed human pose estimation strategy with V-REP allows human-robot interaction simulations in a virtual environment. However accurate localization of user in the scene is needed to be performed for such experiments in simulations.

Conclusion

We conclude this thesis by reiterating the significance of employing vision sensors in modern robotic systems where close/physical human-robot collaboration is expected. The technological development, specially in the domain of portable devices, allows manufacturers to equip specialized and miniaturized vision sensors in modern robotic systems as well. Thus the choice of sensors should not be limited to conventional RGB cameras or even to recent infrared based (or time-of-flight) depth sensors only, but it can also be extended to polydioptric camera rigs as presented in [195]. However, we believe that core vision strategies should be designed to work with monocular systems so as to work even at the time of crises i.e., in the case of unexpected malfunctioning of specialized sensors. Moreover, learning-based methods have demonstrated the state-of-the-art performances in the field of object detection, face identification and in our case gestures recognition also. It is observed that humans are able to approximate depth perception quite well even if they loose one of their eyes in an accident.

Taking inspiration from human cognitive capabilities, it is imperative to complement the vision algorithms with these (deep) learning-based methods for greater accuracy and for tasks which were near impossible before the advent of these approaches.

Bibliography

[1] Jun Wan, Sergio Escalera, Gholamreza Anbarjafari, Hugo Jair Escalante, Xavier Baró, Isabelle Guyon, Meysam Madadi, Juri Allik, Jelena Gorbova, Chi Lin, et al. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3189–3197, 2017.

[2] Farhood Negin, Pau Rodriguez, Michal Koperski, Adlen Kerboua, Jordi Gonzàlez, Jeremy Bourgeois, Emmanuelle Chapoulie, Philippe Robert, and Francois Bremond. Praxis: Towards automatic cognitive assessment using gesture recognition. *Expert Systems with Applications*, 2018.

[3] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke. Human-machine-interaction in the industry 4.0 era. In *IEEE Int. Conf. on Industrial Informatics*, pages 289–294, July 2014.

[4] Brian Gleeson, Karon MacLean, Amir Haddadi, Elizabeth Croft, and Javier Alcazar. Gestures for Industry: Intuitive Human-Robot Communication from Human Observation. In *Proceedings of the 8th ACM/IEEE Int. Conf. on Human-robot Interaction*, pages 349–356, Piscataway, NJ, USA, 2013. IEEE Press.

[5] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, Vol. 40:pp. 1–13, 2016.

[6] Craig Schlenoff, Zeid Kootbally, Anthony Pietromartire, Marek Franaszek, and Sebti Foufou. Intention recognition in manufacturing applications. *Robotics and Computer-Integrated Manufacturing*, Vol. 33:pp. 29–41, 2015.

[7] Iñaki Maurtua, Aitor Ibarguren, Johan Kildal, Loreto Susperregi, and Basilio Sierra. Human-robot collaboration in industrial applications: Safety, interaction and trust. *Int. Journal of Advanced Robotic Systems*, Vol. 14 (2017), 2017.

[8] Siddharth S. Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: A survey. *Artificial Intelligence Review*, Vol. 43:pp. 1–54, Jan 2015.

[9] Pedro Neto, J. Norberto Pires, and António Paulo Moreira. High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition. *Industrial Robot*, Vol. 37:pp 137–147, 2010.

[10] Valeria Villani, Fabio Pini, Francesco Leali, and Cristian Secchi. Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, Vol. 55:pp. 248 – 266, March 2018.

[11] A. Mehrabian. *Nonverbal Communication*. Aldine Publishing Company, 1972.

[12] G. Canal, S. Escalera, and C. Angulo. A real-time human-robot interaction system based on gestures for assistive scenarios. *Computer Vision and Image Understanding*, Vol. 149:pp. 65–77, 2016.

[13] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira. Gesture Recognition for Human-Robot Collaboration: A Review. In *Proceedings of the 7th Swedish Production Symposium*, pages 1–12, 2016.

[14] Francis Quek, David McNeill, Robert Bryll, Susan Duncan, Xin-Feng Ma, Cemil Kirbas, Karl E McCullough, and Rashid Ansari. Multimodal human discourse: gesture and speech. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 9(3):171–193, 2002.

[15] Siddharth S Rautaray and Anupam Agrawal. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54, 2015.

[16] Xenophon Zabulis, Haris Baltzakis, and Antonis A Argyros. Vision-based hand gesture recognition for human-computer interaction. *The universal access handbook*, 34:30, 2009.

[17] Fabien Baradel, Christian Wolf, and Julien Mille. Human action recognition: Pose-based attention draws focus to hands. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 604–613, 2017.

[18] Maryam Asadi-Aghbolaghi, Albert Clapes, Marco Bellantonio, Hugo Jair Escalante, Víctor Ponce-López, Xavier Baró, Isabelle Guyon, Shohreh Kasaei, and Sergio Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pages 476–483. IEEE, 2017.

[19] Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212, 2014.

[20] Jungong Han, Ling Shao, Dong Xu, and Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[21] Javier Ruiz-del-Solar, Patricio Loncomilla, and Naiomi Soto. A Survey on Deep Learning Methods for Robot Vision. *arXiv (2018)*, arxiv:1803.10862, 2018.

[22] Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, Robin Passama, and Andrea Cherubini. Towards Real-time Physical Human-Robot Interaction using Skeleton Information and Hand Gestures. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018.

[23] Osama Mazhar, Benjamin Navarro, Sofiane Ramdani, Robin Passama, and Andrea Cherubini. A real-time human-robot interaction framework with robust background invariant hand gesture detection. *Robotics and Computer-Integrated Manufacturing*, 60:34–48, 2019.

[24] R. Yang, S. Sarkar, and B. Loeding. Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32:pp. 462–477, March 2010.

[25] B. Navarro, A. Fonte, P. Fraisse, G. Poisson, and A. Cherubini. In Pursuit of Safety: An Open-Source Library for Physical Human-Robot Interaction. *IEEE Robotics Automation Magazine*, Vol. 25:pp. 39–50, June 2018.

[26] Jun Wan, Yibing Zhao, Shuai Zhou, Isabelle Guyon, Sergio Escalera, and Stan Z Li. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64, 2016.

[27] ISO 10218-1:2011 - Robots and robotic devices - Safety requirements for industrial robots - Part 1: Robots. ISO; 2011. 2011.

[28] ISO 10218-2:2011 - Robots and robotic devices - Safety requirements for industrial robots - Part 2: Robot systems and integration. ISO; 2011. 2011.

[29] ISO/TS 15066:2016 - Robots and robotic devices - Safety requirements for collaborative industrial robot systems and the work environment: Collaborative robots. ISO/TS; 2016. 2011.

[30] Alessio Levratti, Giuseppe Riggio, Antonio De Vuono, Cesare Fantuzzi, and Cristian Secchi. Safe navigation and experimental evaluation of a novel tire workshop assistant robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 994–999, 2017.

[31] Lorenzo Sabattini, Alessio Levratti, Francesco Venturi, Enrica Amplo, Cesare Fantuzzi, and Cristian Secchi. Experimental comparison of 3D vision sensors for mobile robot localization for industrial application: Stereo-camera and RGB-D sensor. In *12th IEEE Int. Conf. on Control Automation Robotics Vision*, pages 823–828, 2012.

[32] Jeremy A Marvel and Rick Norcross. Implementing speed and separation monitoring in collaborative robot workcells. *Robotics and Computer-Integrated Manufacturing*, Vol. 44:pp. 144–155, 2017.

[33] Vittorio Rampa, Federico Vicentini, Stefano Savazzi, Nicola Pedrocchi, Marcellso Ioppolo, and Matteo Giussani. Safe human-robot cooperation through sensor-less radio localization. In *12th IEEE Int. Conf. on Industrial Informatics*, pages 683–689, 2014.

[34] Federico Vicentini, Massimiliano Ruggeri, Luca Dariz, Alessandro Pecora, Luca Maiolo, Davide Polese, Luca Pazzini, and Lorenzo Molinari Tosatti. Wireless sensor networks and safe protocols for user tracking in human-robot cooperative workspaces. In *23rd IEEE Int. Sym. on Industrial Electronics*, pages 1274–1279, 2014.

[35] Federico Vicentini, Matteo Giussani, and Lorenzo Molinari Tosatti. Trajectory-dependent safe distances in human-robot interaction. In *Emerging Technology and Factory Automation*, pages 1–4. IEEE, 2014.

[36] Przemyslaw A Lasota, Gregory F Rossano, and Julie A Shah. Toward safe close-proximity human-robot interaction with standard industrial robots. 2014.

[37] Alessandro De Luca and Fabrizio Flacco. Integrated control for phri: Collision avoidance, detection, reaction and collaboration. In *4th IEEE RAS & EMBS Int. Conf. on Biomedical Robotics and Biomechatronics*, pages pp. 288–295. IEEE, 2012.

[38] Carlos Morato, Krishnanand N Kaipa, Boxuan Zhao, and Satyandra K Gupta. Toward safe human robot collaboration by using multiple kinects based real-time human tracking. *Journal of Computing and Information Science in Engineering*, Vol. 14, 2014.

[39] Sotiris Makris, Panagiota Tsarouchi, Dragoljub Surdilovic, and Jörg Krüger. Intuitive dual arm robot programming for assembly operations. *CIRP Annals, Manufacturing Technology*, Vol. 63:pp. 13–16, 2014.

[40] Zhengyou Zhang. Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia*, Vol. 19:pp. 4–10, April 2012.

[41] Y. Yang, H. Yan, M. Dehghan, and M. H. Ang. Real-time human-robot interaction in complex environment using kinect v2 image recognition. In *IEEE Int. Conf. on Cybernetics and Intelligent Systems and IEEE Conf. on Robotics, Automation and Mechatronics*, pages 112–117, July 2015.

[42] Marc G. Carmichael, Dikai Liu, Antony Tran, Richardo Khonasty, and Stefano Aldini. Robot Co-worker for Abrasive Blasting: Lessons Learnt in Worker Posture Estimation. In *IEEE Int. Conf. on Robotics and Automation*, May 2018.

[43] J. Sell and P. O'Connor. The Xbox One System on a Chip and Kinect Sensor. *IEEE Micro*, Vol. 34:pp. 44–53, Mar 2014.

[44] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik. Intel(R) RealSense(TM) Stereoscopic Depth Cameras. In *IEEE Conf. on Computer Vision and Pattern Recognition Workshops*, pages 1267–1276, July 2017.

[45] Guanglong Du and Ping Zhang. Markerless human–robot interface for dual robot manipulators using kinect sensor. *Robotics and Computer-Integrated Manufacturing*, Vol. 30:pp. 150–159, 2014.

[46] G. Du, M. Chen, C. Liu, B. Zhang, and P. Zhang. Online Robot Teaching with Natural Human-robot Interaction. *IEEE Transactions on Industrial Electronics, 2018*, 2018.

[47] Elias Matsas and George-Christopher Vosniakos. Design of a virtual reality training system for human–robot collaboration in manufacturing tasks. *International Journal on Interactive Design and Manufacturing*, Vol. 11:pp. 139–153, May 2017.

[48] Elias Matsas, George-Christopher Vosniakos, and Dimitris Batras. Prototyping proactive and adaptive techniques for human-robot collaboration in manufacturing using virtual reality. *Robotics and Computer-Integrated Manufacturing*, Vol. 50:pp. 168–180, 2018.

[49] M. Simão, P. Neto, and O. Gibaru. Natural control of an industrial robot using hand gesture recognition with neural networks. In *42nd Annual Conf. of the IEEE Industrial Electronics Society*, pages 5322–5327, Oct 2016.

[50] Jagdish Lal Raheja, Mona Chandra, and Ankit Chaudhary. 3D Gesture based Real-time Object Selection and Recognition. *Pattern Recognition Letters, 2017*, 2017.

[51] Y. LeCun, Fu Jie Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 97–104, June 2004.

[52] K. Ehlers and K. Brama. A human-robot interaction interface for mobile and stationary robots based on real-time 3D human body and hand-finger pose estimation. In *IEEE Int. Conf. on Emerging Technologies and Factory Automation*, pages 1–6, Sept 2016.

[53] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, Vol. 20:pp. 1371–1375, 1998.

[54] P. Neto, D. Pereira, J. N. Pires, and A. P. Moreira. Real-time and continuous hand gesture spotting: An approach based on artificial neural networks. In *IEEE Int. Conf. on Robotics and Automation*, pages 178–183, May 2013.

[55] Qutaishat Munib, Moussa Habeeb, Bayan Takruri, and Hiba Abed Al-Malik. American sign language (ASL) recognition based on Hough transform and neural networks. *Expert systems with Applications*, Vol. 32:pp. 24–37, 2007.

[56] C Charayaphan and AE Marble. Image processing system for interpreting motion in American Sign Language. *Medical Engineering and Physics*, Vol. 14:pp. 419–425, 1992.

[57] Dana H Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, Vol. 13:pp. 111–122, 1981.

[58] A. Joshi, H. Sierra, and E. Arzuaga. American sign language translation using edge detection and cross correlation. In *IEEE Colombian Conf. on Communications and Computing (COLCOM)*, pages 1–6, Aug 2017.

[59] Pei Xu. A Real-time Hand Gesture Recognition and Human-Computer Interaction System. *arXiv (2017)*, arxiv:1704.07296, 2017.

[60] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86:pp. 2278–2324, Nov 1998.

[61] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2016.

[62] Hong Cheng, Lu Yang, and Zicheng Liu. Survey on 3D Hand Gesture Recognition. *IEEE Trans. Circuits Syst. Video Techn.*, Vol. 26:pp. 1659–1673, 2016.

[63] Michael Van den Bergh and Luc Van Gool. Combining RGB and ToF cameras for real-time 3D hand gesture interaction. In *IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72, 2011.

[64] Iasonas Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Markerless and efficient 26-DOF hand pose recovery. In *Asian Conf. on Computer Vision*, pages 744–757. Springer, 2010.

[65] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Convolutional Pose Machines. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2016.

[66] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. In *IEEE Conf. on Computer Vision and Pattern Recognition*, 2017.

[67] Paschalis Panteleris, Iason Oikonomidis, and Antonis A Argyros. Using a Single RGB Frame for Real Time 3D Hand Pose Estimation in the Wild. In *IEEE Winter Conf. on Applications of Computer Vision*, pages 436–445. IEEE, March 2018.

[68] Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

[69] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[70] Heng Wang, Muhammad Muneeb Ullah, Alexander Klaser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. *Proceedings of the British Machine Vision Conference 2009*, pages 124.1–124.11.

[71] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011.

[72] H. Wang and C. Schmid. Action recognition with improved trajectories. In *2013 IEEE International Conference on Computer Vision*, pages 3551–3558, 2013.

[73] Heng Wang, Dan Oneata, Jakob Verbeek, and Cordelia Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, 119(3):219–238, 2016.

[74] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[75] Vadim Kantorov and Ivan Laptev. Efficient feature extraction, encoding and classification for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2593–2600, 2014.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[79] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012.

[80] Moez Baccouche, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. In *International workshop on human behavior understanding*, pages 29–39. Springer, 2011.

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

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

[83] Natalia Neverova, Christian Wolf, Giulio Paci, Giacomo Sommavilla, Graham Taylor, and Florian Nebout. A multi-scale approach to gesture detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 484–491, 2013.

[84] Natalia Neverova, Christian Wolf, Graham W Taylor, and Florian Nebout. Multi-scale deep learning for gesture detection and localization. In *European Conference on Computer Vision*, pages 474–490. Springer, 2014.

[85] Sergio Escalera, Xavier Baró, Jordi Gonzalez, Miguel A Bautista, Meysam Madadi, Miguel Reyes, Víctor Ponce-López, Hugo J Escalante, Jamie Shotton, and Isabelle Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *European Conference on Computer Vision*, pages 459–473. Springer, 2014.

[86] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[87] Melvyn A Goodale and A David Milner. Separate visual pathways for perception and action. *Trends in neurosciences*, 15(1):20–25, 1992.

[88] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015.

[89] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015.

[90] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[91] Juan C Nunez, Raul Cabido, Juan J Pantrigo, Antonio S Montemayor, and Jose F Velez. Convolutional neural networks and long short-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*, 76:80–94, 2018.

[92] Ronald A Rensink. The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42, 2000.

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

[94] Fabien Baradel, Christian Wolf, and Julien Mille. Pose-conditioned spatio-temporal attention for human action recognition. *arXiv preprint arXiv:1703.10106*, 2017.

[95] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[96] Jiang Wang, Zicheng Liu, Ying Wu, and Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1297. IEEE, 2012.

[97] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[98] Serena Yeung, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *International Journal of Computer Vision*, 126(2-4):375–389, 2018.

[99] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[100] Qiguang Miao, Yunan Li, Wanli Ouyang, Zhenxin Ma, Xin Xu, Weikang Shi, and Xiaochun Cao. Multimodal gesture recognition based on the resc3d network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3047–3055, 2017.

[101] Du Tran, Jamie Ray, Zheng Shou, Shih-Fu Chang, and Manohar Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017.

[102] Edwin H Land. The retinex theory of color vision. *Scientific american*, 237(6):108–129, 1977.

[103] Hao-Shu Fang, Shuqin Xie, Yu-Wing Tai, and Cewu Lu. Rmpe: Regional multi-person pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2334–2343, 2017.

[104] Basura Fernando, Efstratios Gavves, José Oramas, Amir Ghodrati, and Tinne Tuytelaars. Rank pooling for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):773–787, 2016.

[105] Pradyumna Narayana, Ross Beveridge, and Bruce A Draper. Gesture recognition: Focus on the hands. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5235–5244, 2018.

[106] Zhipeng Liu, Xiujuan Chai, Zhuang Liu, and Xilin Chen. Continuous gesture recognition with hand-oriented spatiotemporal feature. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3056–3064, 2017.

[107] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[108] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3941–3950, 2017.

[109] Damien Michel, Ammar Qammaz, and Antonis A Argyros. Markerless 3d human pose estimation and tracking based on rgbd cameras: an experimental evaluation. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, pages 115–122. ACM, 2017.

[110] Ammar Qammaz, Damien Michel, and Antonis Argyros. A hybrid method for 3d pose estimation of personalized human body models. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 456–465. IEEE, 2018.

[111] A Agarwal and B Triggs. 3d human pose from silhouettes by relevance vector regression. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[112] Ross Girshick, Jamie Shotton, Pushmeet Kohli, Antonio Criminisi, and Andrew Fitzgibbon. Efficient regression of general-activity human poses from depth images. In *2011 International Conference on Computer Vision*, pages 415–422. IEEE, 2011.

[113] Catalin Ionescu, Joao Carreira, and Cristian Sminchisescu. Iterated second-order label sensitive pooling for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1661–1668, 2014.

[114] Sijin Li and Antoni B Chan. 3d human pose estimation from monocular images with deep convolutional neural network. In *Asian Conference on Computer Vision*, pages 332–347. Springer, 2014.

[115] Bugra Tekin, Artem Rozantsev, Vincent Lepetit, and Pascal Fua. Direct prediction of 3d body poses from motion compensated sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 991–1000, 2016.

[116] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International journal of computer vision*, 87(1-2):75, 2010.

[117] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *European conference on computer vision*, pages 702–718. Springer, 2000.

[118] Raquel Urtasun, David J Fleet, and Pascal Fua. 3d people tracking with gaussian process dynamical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 238–245. IEEE, 2006.

[119] Alexandru O Balan, Leonid Sigal, Michael J Black, James E Davis, and Horst W Haussecker. Detailed human shape and pose from images. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[120] Marta Sanzari, Valsamis Ntouskos, and Fiora Pirri. Bayesian image based 3d pose estimation. In *European conference on computer vision*, pages 566–582. Springer, 2016.

[121] Ijaz Akhter and Michael J Black. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1446–1455, 2015.

[122] Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In *European Conference on Computer Vision*, pages 561–578. Springer, 2016.

[123] Damien Michel, Costas Panagiotakis, and Antonis A Argyros. Tracking the articulated motion of the human body with two rgbd cameras. *Machine Vision and Applications*, 26(1):41–54, 2015.

[124] Thomas Helten, Meinard Muller, Hans-Peter Seidel, and Christian Theobalt. Real-time body tracking with one depth camera and inertial sensors. In *Proceedings of the IEEE international conference on computer vision*, pages 1105–1112, 2013.

[125] Mao Ye, Xianwang Wang, Ruigang Yang, Liu Ren, and Marc Pollefeys. Accurate 3d pose estimation from a single depth image. In *2011 International Conference on Computer Vision*, pages 731–738. IEEE, 2011.

[126] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. 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, 2013.

[127] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7025–7034, 2017.

[128] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016.

[129] Ching-Hang Chen and Deva Ramanan. 3d human pose estimation= 2d pose estimation+ matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7035–7043, 2017.

[130] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016.

[131] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017.

[132] Hao-Shu Fang, Yuanlu Xu, Wenguan Wang, Xiaobai Liu, and Song-Chun Zhu. Learning pose grammar to encode human body configuration for 3d pose estimation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[133] Xingyi Zhou, Xiao Sun, Wei Zhang, Shuang Liang, and Yichen Wei. Deep kinematic pose regression. In *European Conference on Computer Vision*, pages 186–201. Springer, 2016.

[134] Catalin Ionescu, Fuxin Li, and Cristian Sminchisescu. Latent structured models for human pose estimation. In *2011 International Conference on Computer Vision*, pages 2220–2227. IEEE, 2011.

[135] Sijin Li, Weichen Zhang, and Antoni B Chan. Maximum-margin structured learning with deep networks for 3d human pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 2848–2856, 2015.

[136] Bugra Tekin, Isinsu Katircioglu, Mathieu Salzmann, Vincent Lepetit, and Pascal Fua. Structured prediction of 3d human pose with deep neural networks. *arXiv preprint arXiv:1605.05180*, 2016.

[137] Jamie Shotton, Ross Girshick, Andrew Fitzgibbon, Toby Sharp, Mat Cook, Mark Finocchio, Richard Moore, Pushmeet Kohli, Antonio Criminisi, Alex Kipman, et al. Efficient human pose estimation from single depth images. *IEEE transactions on pattern analysis and machine intelligence*, 35(12):2821–2840, 2012.

[138] Atul Kanaujia, Cristian Sminchisescu, and Dimitris Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[139] Rómer Rosales and Stan Sclaroff. Learning body pose via specialized maps. In *Advances in neural information processing systems*, pages 1263–1270, 2002.

[140] Liefeng Bo and Cristian Sminchisescu. Twin gaussian processes for structured prediction. *International Journal of Computer Vision*, 87(1-2):28, 2010.

[141] Ankur Agarwal and Bill Triggs. Recovering 3d human pose from monocular images. *IEEE transactions on pattern analysis and machine intelligence*, 28(1):44–58, 2005.

[142] Ahmed Elgammal and Chan-Su Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[143] Greg Mori and Jitendra Malik. Recovering 3d human body configurations using shape contexts. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7):1052–1062, 2006.

[144] Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. Vnect: Real-time 3d human pose estimation with a single rgb camera. *ACM Transactions on Graphics (TOG)*, 36(4):44, 2017.

[145] Gregory Rogez, Philippe Weinzaepfel, and Cordelia Schmid. Lcr-net++: Multi-person 2d and 3d pose detection in natural images. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[146] Alexandros Makris and Antonis Argyros. Robust 3d human pose estimation guided by filtered subsets of body keypoints. In *2019 16th International Conference on Machine Vision Applications (MVA)*, pages 1–6. IEEE, 2019.

[147] Yu Chen, Tae-Kyun Kim, and Roberto Cipolla. Inferring 3d shapes and deformations from single views. In *European Conference on Computer Vision*, pages 300–313. Springer, 2010.

[148] Peng Guan, A. Weiss, A. O. Bãlan, and M. J. Black. Estimating human shape and pose from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1381–1388, Sep. 2009.

[149] Gerard Pons-Moll, Andreas Baak, Juergen Gall, Laura Leal-Taixe, Meinard Mueller, Hans-Peter Seidel, and Bodo Rosenhahn. Outdoor human motion capture using inverse kinematics and von mises-fisher sampling. In *2011 International Conference on Computer Vision*, pages 1243–1250. IEEE, 2011.

[150] Feng Zhou and Fernando De la Torre. Spatio-temporal matching for human detection in video. In *European Conference on Computer Vision*, pages 62–77. Springer, 2014.

[151] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer. A joint model for 2d and 3d pose estimation from a single image. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3634–3641, June 2013.

[152] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[153] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schieke. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *European Conference on Computer Vision (ECCV)*, 2016.

[154] Xiaolin Wei, Peizhao Zhang, and Jinxiang Chai. Accurate realtime full-body motion capture using a single depth camera. *ACM Transactions on Graphics (TOG)*, 31(6):188, 2012.

[155] Hashim Yasin, Umar Iqbal, Bjorn Kruger, Andreas Weber, and Juergen Gall. A dual-source approach for 3d pose estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4948–4956, 2016.

[156] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[157] S. Das, M. Koperski, F. Bremond, and G. Francesca. Action recognition based on a mixture of RGB and depth based skeleton. In *14th IEEE Int. Conf. on Advanced Video and Signal Based Surveillance*, pages 1–6, Aug 2017.

[158] Christian Zimmermann, Tim Welschehold, Christian Dornhege, Thomas Brox, and Wolfram Burgard. 3D Human Pose Estimation in RGBD Images for Robotic Task Learning. In *IEEE Int. Conf. on Robotics and Automation*, 2018.

[159] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[160] R. C. Luo and Y. C. Wu. Hand Gesture Recognition for Human-Robot Interaction for Service Robot. In *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pages 318–323, Sept 2012.

[161] M. Van den Bergh, D. Carton, R. De Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, and M. Buss. Real-time 3D Hand Gesture Interaction with a Robot for Understanding Directions from Humans. In *RO-MAN*, pages 357–362, July 2011.

[162] Grazia Cicirelli, Carmela Attolico, Cataldo Guaragnella, and Tiziana D'Orazio. A Kinect-based Gesture Recognition Approach for a Natural Human Robot Interface. *Int. Journal of Advanced Robotic Systems*, Vol. 12:pp. 22, 2015.

[163] Matteo Munaro, Filippo Basso, and Emanuele Menegatti. Tracking people within groups with RGB-D data. *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages pp. 2101–2107, 2012.

[164] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *Int. Journal of Computer Vision*, Vol. 115:pp. 211–252, 2015.

[165] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[166] François Chollet et al. Keras. https://keras.io, 2015.

[167] Robbin Battison. Lexical Borrowing in American Sign Language. 1978.

[168] A Cherubini, R Passama, B Navarro, M Sorour, A Khelloufi, O Mazhar, S Tarbouriech, J Zhu, O Tempier, A Crosnier, P Fraisse, and S Ramdani. A Collaborative Robot for the Factory of the Future: BAZAR. *Int. Journal of Advanced Manufacturing Technology, Special Issue "Design and Management of Digital Manufacturing & Assembly Systems in the Industry 4.0 era, 2019 (to appear)*, 2019.

[169] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[170] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding. Data mining with big data. *IEEE transactions on knowledge and data engineering*, 26(1):97–107, 2013.

[171] G Sreenu and MA Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6(1):48, 2019.

[172] Hye Sun Park, Do Joon Jung, and Hang Joon Kim. Vision-based game interface using human gesture. In *Pacific-Rim Symposium on Image and Video Technology*, pages 662–671. Springer, 2006.

[173] Xu Zhang, Xiang Chen, Wen-hui Wang, Ji-hai Yang, Vuokko Lantz, and Kong-qiao Wang. Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors. In *Proceedings of the 14th international conference on Intelligent user interfaces*, pages 401–406. ACM, 2009.

[174] Lionel Pigou, Sander Dielean, Pieter-Jan Kindermans, and Benjamin Schrauwen. Sign language recognition using convolutional neural networks. In *European Conference on Computer Vision*, pages 572–578. Springer, 2014.

[175] Malima, Ozgur, and Cetin. A fast algorithm for vision-based hand gesture recognition for robot control. In *2006 IEEE 14th Signal Processing and Communications Applications*, pages 1–4, 2006.

[176] Jagdish Lal Raheja, Radhey Shyam, Umesh Kumar, and P Bhanu Prasad. Real-time robotic hand control using hand gestures. In *2010 Second International Conference on Machine Learning and Computing*, pages 12–16. IEEE, 2010.

[177] Isabelle Guyon, Vassilis Athitsos, Pat Jangyodsuk, and Hugo Jair Escalante. The chalearn gesture dataset (cgd 2011). *Machine Vision and Applications*, 25(8):1929–1951, 2014.

[178] Jamie Shotton, Andrew Fitzgibbon, Andrew Blake, Alex Kipman, Mark Finocchio, Bob Moore, and Toby Sharp. Real-time human pose recognition in parts from single depth images. In *CVPR*, pages 1297–1304, June 2011.

[179] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19:4–12, April 2012.

[180] Derek Denny-Brown. The nature of apraxia. *Journal of Nervous and Mental Disease*, 1958.

[181] Osama Mazhar. OpenSign - Kinect V2 Hand Gesture Data - American Sign Language, 2019.

[182] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[183] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[184] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[185] Paschalis Panteleris, Iason Oikonomidis, and Antonis Argyros. Using a single rgb frame for real time 3d hand pose estimation in the wild. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 436–445. IEEE, 2018.

[186] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[187] Oriol Vinyals, Suman V Ravuri, and Daniel Povey. Revisiting recurrent neural networks for robust asr. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4085–4088. IEEE, 2012.

[188] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[189] Ilya Sutskever, James Martens, and Geoffrey E Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[190] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[191] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[192] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[193] Huogen Wang, Pichao Wang, Zhanjie Song, and Wanqing Li. Large-scale multimodal gesture recognition using heterogeneous networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3129–3137, 2017.

[194] Ting-Kuei Hu, Yen-Yu Lin, and Pi-Cheng Hsiu. Learning adaptive hidden layers for mobile gesture recognition. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[195] Osama Mazhar, Ahmad Zawawi Jamaluddin, Cansen Jiang, David Fofi, Ralph Seulin, and Olivier Morel. Design and calibration of a specialized polydioptric camera rig. *Frontiers in ICT*, 4:19, 2017.

[196] Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, and Robin Passama. A framework for real-time physical human-robot interaction using hand gestures. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 46–47. IEEE, 2018.

Published Papers

- Osama Mazhar, Benjamin Navarro, Sofiane Ramdani, Robin Passama, and Andrea Cherubini. A real-time human-robot interaction framework with robust background invariant hand gesture detection. *Robotics and Computer-Integrated Manufacturing*, 60:34–48, 2019

- Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, Robin Passama, and Andrea Cherubini. Towards Real-time Physical Human-Robot Interaction using Skeleton Information and Hand Gestures. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018

- Osama Mazhar, Sofiane Ramdani, Benjamin Navarro, and Robin Passama. A framework for real-time physical human-robot interaction using hand gestures. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, pages 46–47. IEEE, 2018