Deep Vision for Prosthetic Grasp

Ghazal Ghazaei

School of Engineering

Newcastle University

A thesis submitted for the degree of

Doctor of Philosophy

September 2019

To my beloved parents, without whom none of my success would be possible.

Declaration

I hereby declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Ghazal Ghazaei

September 2019

Acknowledgements

The completion of this dissertation is attributed to contributions from many people who directly or indirectly affected me during the way. Therefore, I would like to gratefully acknowledge all the people who aided me during my PhD journey.

I would like to thank Kianoush Nazarpour for his support all over my PhD and making me believe in myself. He gave me complete freedom in my research and provided me with resources no matter how hard it was. Kia also helped me in exploring my research area by attending different universities and summer schools, in which I achieved a lot of experience and discovered new domains of research. Kia's guidance and support greatly encouraged me along the way. I am also thankful to Patrick Degenaar, who provided me helpful feedback in all respects of my PhD .

I would like to express my gratitude to Nassir Navab who hosted me in his great team at technical university of Munich, where I gained a lot of invaluable experience. I am wholeheartedly thankful to Federico Tombari, for giving me the great opportunity to be part of his team and sharing their resources with me. His guidance on my research opened new horizons to me and his computer vision team taught me a lot on research areas that I was not confident about before.

I would also like to express my gratitude to people at Newcastle university who their support is never forgotten. Specifically, I would like to thank Gill Weber for her great support and kindness. I am grateful to Barry Mecrow for facilitating my studies by his great support. I am also grateful to the people in the mechanical lab of Newcastle university, who aided me in improving the hand prosthesis design. Moreover, I am thankful to the two amputees who generously volunteered to make this work possible. Their perseverance, excitement and patience during the experiments inspired me enormously and motivated me for my future career decisions.

On a personal note, I am appreciative of all my friends for all their love and support. Specially, I would like to thank Fahimeh Dehkhoda, Kabita Adhikari, Ali Alameer, Mehrnoosh Mokhtarimehr, Emma Brunton and Safa Awny for making my life in Newcastle enjoyable.

Finally, my deep and sincere gratitude to my loving family, my mother, father, and brother, as well as Mahdi, more than words can describe, for their continuous and unparalleled love, help and support. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. They always kept my heart warm by selflessly encouraging and accompanying me throughout my life. This journey would not have been possible if not for them, and I dedicate this milestone to them.

I express my gratitude to the EPSRC and school of engineering at Newcastle university for funding my doctoral studies. I also sincerely thank NVIDIA ® for donating me a TITAN Xp GPU through which I carried out my deep learning experiments.

Abstract

The loss of the hand can limit the natural ability of individuals in grasping and manipulating objects and affect their quality of life. Prosthetic hands can aid the users in overcoming these limitations and regaining their ability. Despite considerable technical advances, the control of commercial hand prostheses is still limited to few degrees of freedom. Furthermore, switching a prosthetic hand into a desired grip mode can be tiring. Therefore, the performance of hand prostheses should improve greatly.

The main aim of this thesis is to improve the functionality, performance and flexibility of current hand prostheses by augmentation of current commercial hand prosthetics with a vision module.

By offering the prosthesis the capacity to see objects, appropriate grip modes can be determined autonomously and quickly. Several deep learning-based approaches were designed in this thesis to realise such a vision-reinforced prosthetic system. Importantly, the user, interacting with this learning structure, may act as a supervisor to accept or correct the suggested grasp. Amputee participants evaluated the designed system and provided feedback.

The following objectives for prosthetic hands were met:

1. Chapter 3: Design, implementation and real-time testing of a semi-autonomous vision-reinforced prosthetic control structure, empowered with a baseline convolutional neural network deep learning structure.
2. Chapter 4: Development of advanced deep learning structure to simultaneously detect and estimate grasp maps for unknown objects, in presence of ambiguity.
3. Chapter 5: Design and development of several deep learning set-ups for concurrent depth and grasp map as well as human grasp type prediction.

Publicly available datasets, consisting of common graspable objects, namely Amsterdam library of object images (ALOI) and Cornell grasp library were used within

this thesis. Moreover, to have access to real data, a small dataset of household objects was gathered for the experiments, that is Newcastle Grasp Library.

Statement of Originality

The contributions of this thesis have been supported by different journal and conference papers, which have been generated during the journey of my PhD study. They can be listed as follows:

Published

G Ghazaei, A Alameer, P Degenaar, G Morgan, and K Nazarpour, 2015. An Exploratory Study on the Use of Convolutional Neural Networks for Object Grasp Classification. 2nd IET International Conference on Intelligent Signal Processing (ISP). London.

G Ghazaei, A Alameer, P Degenaar, G Morgan and K Nazarpour, 2017. Deep Learning-based Artificial Vision for Grasp Classification in Myoelectric Hands. Journal of Neural Engineering, 14(3), p.036025.

G Ghazaei, F Tombari, N Navab and K Nazarpour, 2018. Grasp Type Estimation for Myoelectric Prostheses using Point Cloud Feature Learning. Workshop on Human-aiding Robotics, International Conference on Intelligent Robots and Systems (IROS).

G Ghazaei, I Laina, C Rupprecht, F Tombari, N Navab and K Nazarpour, 2018. Dealing with Ambiguity in Robotic Grasping via Multiple Predictions. 14th Asian Conference on Computer Vision (ACCV).

Awards

These awards are achieved based on the research carried out during this PhD:

IET William James Award 2017: An award presented to encourage, support and recognise outstanding work of PhD students demonstrating a high level of commitment and advanced understanding of Biomedical Engineering.

UNESCO Netexplo award 2017: The award identifies ten major science and technology innovations every year, which have the potential to impact society and business significantly.

Contents

List of Figures

List of Tables

Nomenclature

Acronyms/Abbreviations

ALOI	Amsterdam library of object images
ANOVA	analysis of variance
AR	augmented reality
BN	batch normalisation
BOC	between-object cross-validation
BSC	between-shape cross-validation
CNN	convolutional neural network
COIL	Columbia object image library
DoF	degree of freedom
EMG	electromyography
FCRN	fully convolutional residual network
GD	gradient descent
GMM	Gaussian mixture model
GPU	graphics processing unit
HoG	histogram of oriented gradients
IMU	inertial measurement unit
LDA	linear discriminant analysis
LRN	local response normalisation

LSE	least square error
MARE	absolute relative error
MHP	multiple hypotheses prediction
ML	machine learning
MTL	multi-task learning
MVC	maximum voluntary contraction
n-D	n-dimensional
PCA	principal component analysis
PDF	probability density function
R-CNN	region-based convolutional neural network
ReLU	rectified linear unit
ResNet	residual network
RFCN	region-based fully convolutional network
RMSE	root mean squared error
SD	standard deviation
SGD	stochastic gradient descent
SIFT	scale-invariant feature transform
SSD	single shot detector
SURF	speeded up robust features
SVM	support vector machine
WOC	within-object cross-validation
YOLO	you only look once

Chapter 1

Introduction

The natural ability of individuals is highly hampered by losing a hand. Prosthetic hands can play an indispensable role in the lives of these group of people by facilitating them in performing daily routines. Although great advances have been made in development of hand prostheses, the control procedure is still unnatural. Specifically, current commercial hand prosthetics require their users to switch between possible grip modes to attain a certain grip, which is an exhaustive procedure and restricts the functional degrees of freedom. Consequently, further improvement of hand prosthesis controllers is required to provide amputees with a human-like performance of an artificial hand [10, 21–24].

Benefiting from the recent developments in computer vision and deep learning, a novel approach towards the limitations of artificial hands can be devised. Artificial hands can be augmented with a vision module such that they can see the world. The aim of this thesis is to enhance the grip functionality of artificial hands by providing them with an artificial vision system. Such "smart" vision can recognise objects and therefore automatise the process of grasping by providing the users with the appropriate grasp type. To have such a vision module, a variety of advanced deep-learning based techniques are designed, developed and implemented in this thesis. The vision module can simply be augmented over a commercial artificial hand available in the market. The amputee user can utilise the system comfortably by pointing the hand to an object of interest. This act causes the vision module to take a snapshot of the target, process it and output a grip mode accordingly. In this way, the user skips the straining procedure of grasp selection and acts as a supervisor to accept or correct the proposed grasp. Having such a semi-autonomous vision-based control can potentially provide a quicker and more flexible decision with more possibilities of grasping.

1.1 Aims and Objectives

The main aim of this thesis is to improve the grasping performance of current hand prostheses by addition of a vision module such that more amputees could benefit from these systems. This aim can be met through the following objectives:

- Identify an optimal grip pattern more efficiently, in terms of accuracy and response time;
- Design a deep learning structure for grasp identification with potential to detect the appropriate grasp for unknown objects;
- Reduce the cognitive burden on the user through a semi-autonomous control of hand.

In this thesis, it was endeavored to achieve the aforementioned aims and objectives that all can lead to more dexterity of amputees when using hand prostheses. Different structures were designed and developed inspired by the most recent deep learning solutions. The capabilities of the implemented approaches were investigated thoroughly both in offline and real-time experiments with amputee users. In this way, the user's response was also regarded as a contributing factor for the improvement of the system.

1.2 Overview and Contributions

The thesis is organised chronologically such that first a basic platform for the presentation of a *Deep learning-based artificial vision* system is built and developed within the subsequent chapters. Each additional development resolves a specific limitation of this baseline platform to build up a system that can fulfill the final purpose of this thesis. Consequently, this thesis is presented as three main contributions.

1.2.1 Deep Learning-based Artificial Vision for Grasp Classification in Myoelectric Hands

The first contribution of this work concerns with building a system that enables semi-autonomous grip mode selection for an artificial hand augmented with an RGB camera. To this end, a convolutional neural network architecture [25] was designed for grasp classification of input RGB images of a target object. In this way, rather than their type, objects were categorised into four grasp groups: *tripod*, *pinch*, *palmar wrist neutral* and *palmar wrist pronated*.

This structure presents a proof-of-principle for the idea of *vision-based hand prosthesis*. This system was evaluated in three ways: offline, real-time computer-based and real-time with an amputee user in the loop. All experiments were analysed comprehensively to extract the advantages and disadvantages of the suggested structure. The results indicated promising performance improvement in control of hand prostheses. Therefore, this system was utilised as a foundation for the successive chapters.

1.2.2 Accurate Object Localisation and Grasp Map Estimation in Presence of Ambiguity

The second contribution initiates a new approach to the problem of grasping in order to boost the performance of the preceding system in a systematic way. To this aim, inspired by a successful robotic grasping platform presented in [17], this contribution focuses on grasp map estimation for human grasping.

An initial idea for a grasp map is the popular grasp rectangle representation [26] including width, height and center of a gripper. To not explicitly learn grasp rectangle parameters and exploit spatial information of the object together with the grasp, grasp rectangles were redefined to *grasp belief maps*. A fully convolutional residual network [27, 28] is then trained to learn an implicit image-dependent spatial model of the grasp.

An issue with such structure is availability of several valid grasps per object. To tackle this high ambiguity in grasp map estimation, a multiple hypothesis platform [29] was developed and adapted to our particular task. The final structure consists of a fully convolutional neural network reformulated by a multiple hypothesis prediction model and relevant meta-loss and optimisation procedure. As several grasp belief maps are produced by this model, a Gaussian mixture model (GMM) evaluation is applied to the output belief maps to opt the *best* belief map for each object for the purpose of comparison with other works and also practical usage of the suggested grasp belief maps.

1.2.3 Grip Pattern Classification for Prosthetic Hands using Estimated Grasp and Depth Maps

The last contribution concentrates on mapping the output of previous work (multiple grasp belief maps) to a human grasp pose. To that end, a contributing source of information can be depth which can be achieved by either a depth sensor or depth estimation of the input RGB

image. The latter can be done simultaneous with grasp map estimation through a multi-task fully convolutional learning framework.

Having the RGB-D information of the object image accompanied with the predicted grasp belief maps, the appropriate grasp type for the target object can be predicted. There are several possibilities for attaining a grip pattern from the different data modalities provided here. Therefore, a variety of architectures were implemented to examine the best solution for achieving object grasp types. The last two contributions lead to an integrated framework for object detection, grasp localisation and estimation of household objects for artificial hands.

1.3 Thesis Outline

The remaining of the thesis is organised accordingly:

- **Chapter 2** provides the basics of prosthetic hands, their development over years and the state-of-the-art structures in both academic and commercial areas. To better introduce the specific interest of this thesis, a comprehensive literature review over the relevant sensor-based methods for control of artificial hands and relevant research areas that can contribute to function of hand prostheses including computer vision as the main augmentation to prosthetic hands and robotic grasping as a parallel field of research is provided.

- **Chapter 3** proposes a semi-autonomous platform for visually augmented prosthesis grasping benefiting from a convolutional neural network-based grasp classification framework. The platform is evaluated by offline and real-time experiments and analyses.

- **Chapter 4** presents a solution for dealing with ambiguity in grasping novel objects through multiple hypotheses prediction. In this way, several grasp maps can be detected and estimated for an unseen object. A ranking method based on Gaussian mixture models was further devised to pick one of the predicted grasp maps.

- **Chapter 5** represents a concurrent depth and grasp belief map estimation platform, which is then used for more accurate grasp classification of unseen objects.

- **Chapter 6** discusses and summarises the contributions of this work and suggests future steps that can be taken for further improvements.

Chapter 2

Background and Literature Review

This chapter presents a comprehensive overview over prosthetic hands and the recent research on them. A more detailed review over the recent research on sensor-based artificial hands and its relevant domains, robotic grasping and computer vision, is also presented.

The chapter is arranged as follows: An introduction to prosthetic hands including the challenges transradial amputees face and the benefits of prosthetic hands followed by their progress during recent decades is presented in Section 2.1. Section 2.2 concentrates on the state-of-the-art solutions for control of hand prostheses in both commercial and academic domains and a comparison between these solutions.

Section 2.2.4 delves deeper into the focus of this thesis and introduces sensor fusion for prosthetic hands as a promising trend. As vision plays a significant role as a modality to be augmented over EMG data, Section 2.3 focuses on computer vision, the advancements there and basic building blocks in popular deep learning models. More attention is drawn to computer vision solutions with the possibility to be applied to artificial hands.

Finally, Section 2.4 provides a summary over the most recent research works in robotic grasping, which overlaps with human grasping task to a great extent.

2.1 Prosthetic Hands

Prosthetic limbs play an indispensable role in functional rehabilitation of amputees and people with congenital deficit. This role is even more accentuated when considering the pervasiveness of upper-limb referrals among the younger and more active age groups of people to whom the loss of a limb can be both highly costly due to life-time care and of great impact on their life quality. Hence, advanced hand prosthesis can provide individuals with limb difference with the

opportunity to return to their normal life style and career.

2.1.1 Upper Limb Loss

Every year a large number of people undergo amputation and end up with upper- or lower-limb loss or born with congenital deficit. In the UK, for example there are annually around 5000 referrals to prosthetics services, among which upper-limb amputations account for $\sim 6\%$ [1]. Also in the United States lower-limb amputation is more common (80%) than upper-limb (10%) or multiple-limb (10%) loss. This however does not affect the importance of upper-limb deficiency and its solutions as there are unique challenges and issues accompanying with this kind of amputation [30].

The statistics of UK NHS indicate that about 60% of all the upper-limb amputations are among the age groups who are younger than 54 [1]. Considering this age group as the active members of the society, the loss of a limb can have dramatic effects on the lives of these people. Upper-limb loss can have different extents and causes, which are illustrated in details in Figure 2.1.

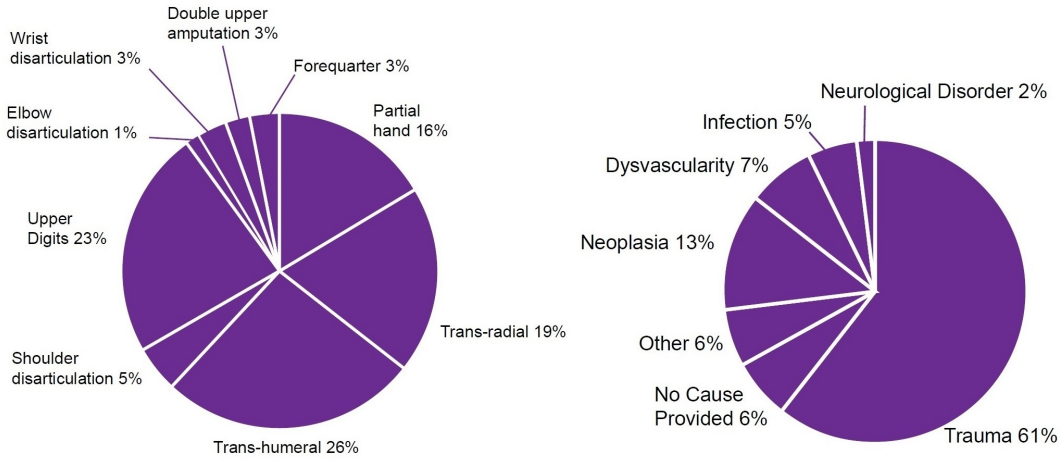


Figure 2.1: Details of upper limb deficiency in UK in 2004-2005 [1]: A) The level of amputation, B) The causes of amputation, in which trauma is the main one.

Prosthetic hands as a functional rehabilitation route can significantly improve the lives of people with transradial amputation. There are two main categories for functional upper-limb prostheses:

- **Body-powered prostheses**

  These systems provide motion of an artificial hand through cable and strap connections to the upper-body muscles.

- **Externally-powered prostheses**

  The most common type of externally-powered prosthetics are myoelectric prostheses. These devices are controlled via electromyography (EMG) signals recorded non-invasively from skin surface. This data is used as an electric command to the motors in order to move the artificial hand.

2.1.2 Evolution of Prosthetic Hands Over Time

For decades, humans have been attempting to find technological solutions for the rehabilitation of the hand amputees. Although there are several challenges accompanied with this endeavor, current progress of hand prostheses technology is considerable.

One of the earliest prosthetic hands belonged to Marcus Sergius, a roman general, who was able to return to battle thanks to this prosthesis [31]. Since then (218-201 BC), there were several relatively smart prosthetic hand designs, which raised the ability of knights in battlefield. Unfortunately, these designs were only dedicated to prosperous groups of society and thus very scarce. In 1818, for the first time the notion of *automatic* body-powered upper-limb prosthesis was introduced by a German dentist called Peter Baliff [31]. In this device, the motion of a terminal device attached to the amputation stump was evoked by intact muscles of the trunk and shoulder girdle leading to fluid body motions.

After World War I (1914-1918) the number of amputees and amputee rehabilitation programs raised immensely. World War II (1939-1945) boosted the attention to prosthetic limbs more than ever promoting the foundation of US Committee on Prosthetics Research and Development. In 1948, the Bowden cable body-powered prosthesis was developed replacing previous design's bulky straps with cables, which is the main reference for current body-powered prostheses. This prosthetic hand offered durability, portability and effective speed, motion and force ranges as a comparatively economical option to transradial amputees [31] (shown in Figure 2.2). The body-powered hand prostheses also enabled the amputees to use both their hands at the same time and are very common even these days. However, prolonged wearing can be cumbersome and a human-like appearance is missed in these prosthetic hands.

The first instance of development of externally powered prostheses, using pneumatic and electric power, was found in a German book titled *Limb Substitutes and Work Aids* in 1919 [31]. The designs were however too complex to be used in present prosthtic hands. The first myelectric prosthesis was introduced by a German student at Munich University in 1948, which did not receive sufficient appreciation at that time [31]. The first myoelectric prosthetic hand,

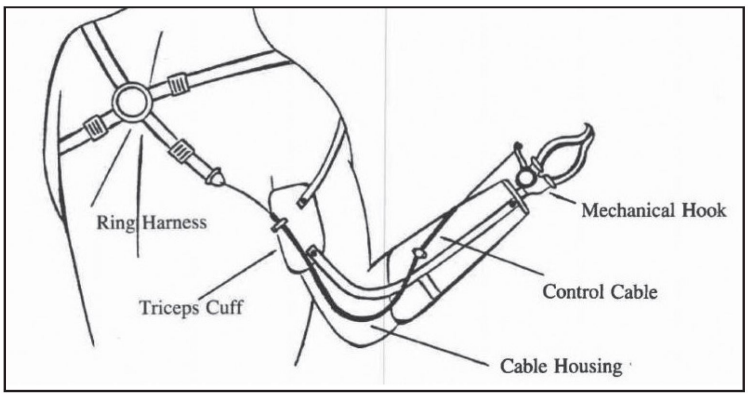


Figure 2.2: Bowden's cable body-powered prosthetic hand [2]

which was clinically accepted was the "Russian Hand" benefiting from portable batteries and electronics and a skin-colored cosmetic glove, while still having limitations such as heaviness, slow movement and unreliable electronics [31]. Until 1980s, myoelectric prostheses were widespread and lighter and more flexible hands with rechargeable batteries and more reliable electronics were offered. The scheme of a myoelectric prosthesis for transradial amputees is depicted in Figure 2.3.

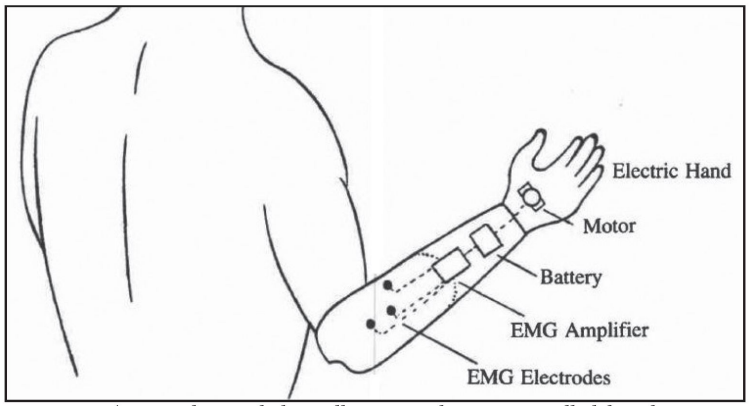


Figure 2.3: Myoelectric hand prosthesis controlled by electromyographic signals collected from the amputated limb stump [2].

Myoelectric prostheses are non-invasive featuring more human-like appearance of a hand and more comfort. Contrary to body-powered prostheses, myoelectric prostheses require regular charging and lack a sensory feedback. Besides, producing distinct signals is an exhausting task, which usually involves long training intervals [31]. In addition, these systems are mostly involving a delay during the task performance and not robust to environmental effects, *e. g.* sweating, electrode position change. It is also worth mentioning that myoelectric prostheses are considerably more costly than their body-powered version [31, 32].

Most recently in 2004, targeted motor reinnervation (TMR) was introduced as an innovative

intuitive artificial limb control approach. In this method, the amputated nerves are rerouted to intact muscles such that robust EMG signals are provided for the artificial limb (Figure 2.4). In this way, not only the system is more intuitive, but multiple joint movement and better flow is feasible [3, 31, 32].

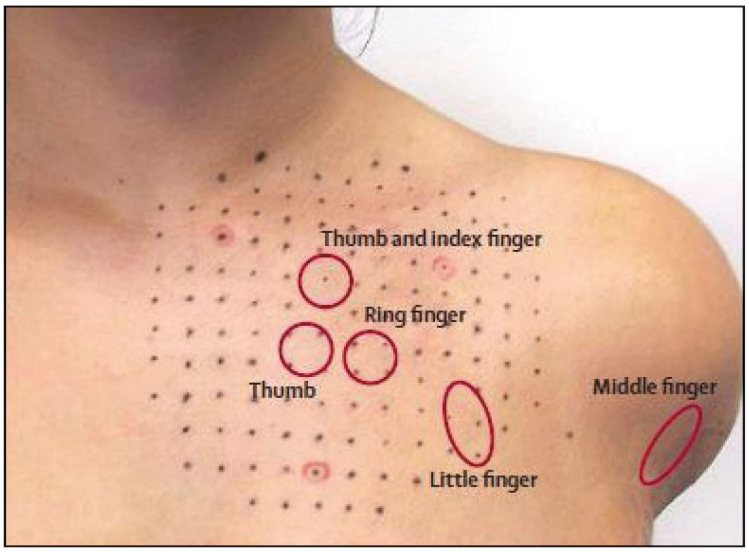


Figure 2.4: Cutaneous *map* created by TMR through which the amputated arm can *feel* the stimulation [3].

2.2 State-of-the-art Approaches Towards Prosthetic Hand Control

In this section more aspects of myoelectric hand prostheses as the most prevalent type of prosthetics and main focus of state-of-the-art artificial hands are discussed. These artificial hands utilise surface electrodes attached to skin around the remnant muscles of the amputated limb to measure EMG signals. A variety of methods are applied to process these EMG recordings and generate more robust control signals.

2.2.1 Myoelectric Control

Myoelectric control of artificial hands is associated with user's intention. Although electromyogram signals are recorded indirectly from the surface of muscles, they contain sufficient neural information of motor tasks similar to the information that can be gained by direct nerve recording. That is, the triggered muscle fibres and its motor neuron action potentials are correspond-

ing closely. The EMG signal is therefore calculated as the total electrical activity of muscle fibers [4].

The simplest strategy for myelectric control of a hand is a simple on-off controller activated by the information achieved from the EMG signal. A common method could be applying root mean square (RMS) or mean absolute value (MAV) to the EMG signal and use the comparison of the resulting amplitude with a predefined threshold to actuate the controller. The two common myoelectric control strategies are sequential control and simultaneous control. Although the latter is preferable, the majority of current hand prostheses are controlled sequentially and more effort is required for the application of simultaneous control in artificial hands [33]. Some available methods in myoelectric control are illustrated in the schematic diagram of Figure 2.5 and further explained in the following:

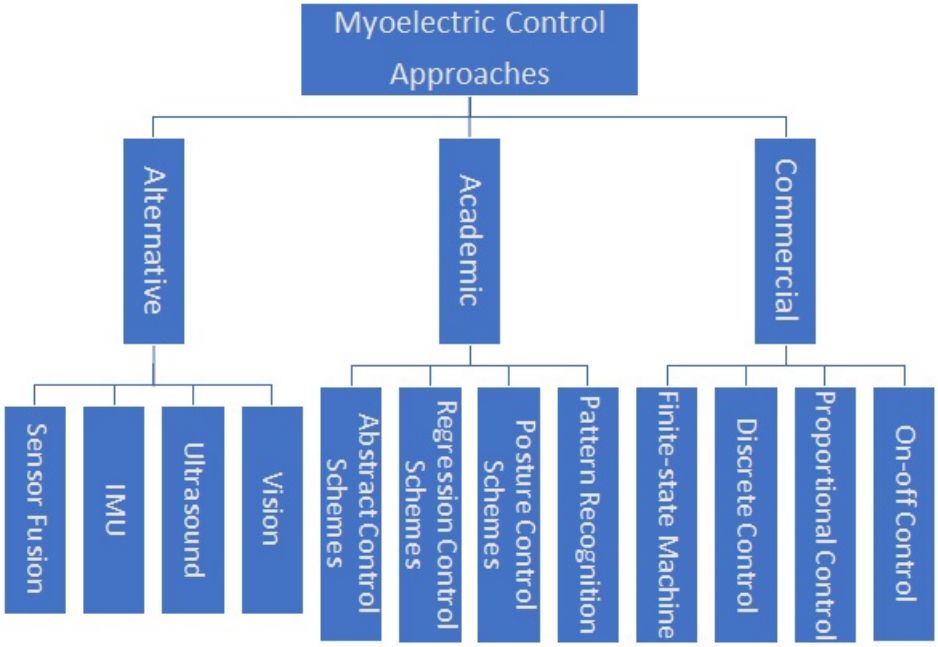


Figure 2.5: A summary of available myoelectric control strategies.

- **On-off control:** Activation of controller when the amplitude of EMG signal reaches a specific amplitude. This type of control is limited to two degrees of freedom (DoF).
- **Proportional control:** The intensity of EMG signal affects the motor voltage proportionally.
- **Direct control:** Similar to proportional control in which individual control of fingers may be possible. However, due to crosstalk[1] in EMG signals the task is highly challenging.

[1]Crosstalk is a phenomena in which the recorded signals are not generated from the assumed target muscle, but from other local muscle(s) [4].

- **Finite-state machine control:** Each hand gesture is defined as a state and the transitions between these states is also specified. Hence, the method is suitable only for a fixed number of hand postures.
- **Pattern recognition-based control:** This approach involves extraction of particular features from segmented EMG signals and feeding them to a classifier.
- **Posture control schemes:** Unique maps are created to transform EMG control signals in the principal component domain and the domain coordinates are also transformed into joint angels to produce desired hand postures. This method can provide simultaneous control of an artificial hand.
- **Regression control schemes:** Regression methods provide both concurrent and proportional control of a myoelectric hand by generating specific control signals such as joint angles [33].
- **Abstract control schemes:** Recently used for the control of hand prosthesis by [34], abstract decoding techniques induce changes to the inverse model, representing the relation of motor outputs to arbitrary control variables, rather than estimating motor commands in previous methods.

2.2.2 Commercial Approaches

Earlier commercial artificial hands were mainly utilising the on-off controller. Each EMG channel could be assigned to a function and when a certain threshold is reached the function is triggered. Despite providing intuitive control for the user, this method requires two signal sites for each function, *e. g.* hand flexion and extension should be specified to distinct electrode placements. Thus, the system is not practical for multifunction prostheses [4]. Nowadays however finite-state machine controllers are used to enable employment of multiple degrees of freedom (DoFs) from which the user can pick one at a time [4].

A more advanced method than on-off control is *level coding*, in which the whole range of muscle activity, from exceeding the threshold to full contraction are specified to multiple hand functions, each belonging to a particular interval. Although this method seem to suit better for mutifunctionality, in clinical practice the user ends up with atmost three DoFs to control the prosthesis reliably with less robustness than the direct control [4]. An example of this conventional myoelectric control system with two DoFs is illustrated in Figure 2.6. Two EMG channels are used for bidirectional control of one DoF. That is, the two channels can

be responsible for hand opening and closing when a certain threshold for each channel is met. In case of activation of both thresholds (co-contraction), the controller switches to the other DoF, which is wrist flexion/extension. This control scheme is implemented in several popular commercial mutifunction artificial hands, namely the Michelangelo hand [8] and i-limb [6] which are shown in Figure 2.7.

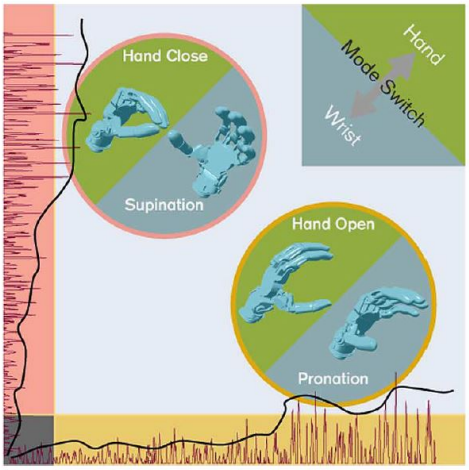


Figure 2.6: Two DoFs level coding approach, a typical myoelectric control structure applied to several conventional commercial systems [4].

The i-limb ultra [6] is one of the most advanced prosthetic hands. It offers proportional control of fourteen automated grips and gestures and auto-grasp feature to prevent objects from slipping.

Another powerful commercial prosthetic hand is the bebionic hand [5]. These hands are also provided with fourteen selected grip patterns. The bebionic benefits from four wrist options, proportional speed control, auto-grip and foldaway fingers. One of the most recent hand prostheses is the Vincent Evolution 2 [7], the first touch sensing hand prosthesis. This hand benefits from some kind of direct and proportional control and designed to produce 12 different grip patterns with only two EMG signals. The force feedback provides the potential to improve the grip reliability, while the effectiveness of feedback performance is not accurately confirmed yet [35]. Figure 2.7 presents the four modern commercial hands mentioned.

It is worth to note that the first commercial pattern recognition-based myoelectric system has been recently emerged in the market called the COAPT system [36]. It features the COAPT Complete Control® as an interface for myoelectric classification suited to a range of upper-extremity prostheses. This system can perform 3-6 different grasp types naturally [37].

In spite of the significant advancements of commercial prosthetics, there are still several shortcomings with these systems that hamper their acceptance among amputees [38]. Lack of

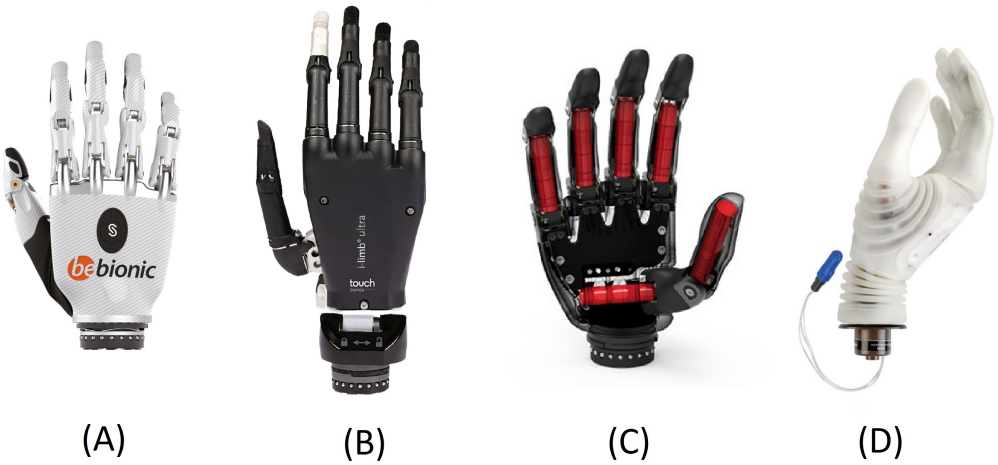


Figure 2.7: Four modern commercial hands: A) the bebionic hand [5], B) the i-limb ultra [6], C) the Vincent Evolution 2 [7] and D) the Michelangelo hand [8].

reliability, flexibility and a natural control procedure are among the many reasons for prosthesis rejection [4, 38]. Hence, the performance of current commercial prosthetic hands can still be improved in a variety of ways [10, 21–24].

Table 2.1 indicates the capabilities of current commercial artificial hands and their limitations. This comparison is valid only for the common commercial hand prostheses. The COAPT system [36], which was introduced to the market after the research ideas of this work were settled, requires a separate category. A careful review of the table leads to a better understanding over the motivation of devising a novel approach towards artificial hands.

2.2.3 Academic Research Approaches

As efforts for tackling the limitations of commercial hand prostheses, academic research has been involved with applying pattern recognition techniques to classify EMG signals for decades [4, 9]. More recently, simultaneous and proportional control (SPC) of multiple DoFs through regression-based myoelectric control paradigms [39, 40] was proposed to enable concurrent selection of DoFs. As intermediate solutions, alternative innovative multimodal schemes for control of artificial hands were also introduced recently [10, 11, 41–51].

Pattern Recognition Solutions

Myoelectric research focused on pattern recognition approaches for quite a long time [4, 9]. These methods assume that there are distinctive features, which represent each muscle activation and finding these specific features is the main task of pattern recognition schemes for myoelectric control.

Table 2.1: Highlighting the performance of the current state-of-the-art commercial artificial hands and their strengths and weaknesses.

Tasks	Performance	characteristics
Functionality	✓	Presence in market for a long time. Limited grasp types in practice.
Speed	X	Long delay for switching between different grip modes.
Robustness	✓	Robust On-off control. Not sensitive to environmental parameters.
Flexibility	X	Control simplicity. Lack of flexibility.
User friendly	X	Burdensome grasping task. The requirement of learning the control rule by the user.
Preparation time	X	Several tests with amputees (usually takes 3-4 days).

Pattern recognition-based myoelectric control systems consist of four main modules [9]:

- **Data segmentation:** Involves various techniques for provision of an appropriate signal for further processing in successive steps.
- **Feature extraction:** As one of the most crucial steps of pattern recognition, the feature extraction module should find the most distinctive and robust features and feed them to a classifier for grasp selection.
- **Classification:** This module categorises the extracted features into previously defined classes. The classifier should be sufficiently robust to classify features considering their changes subject to variation in physical and physiological conditions.
- **Controller:** The controller produces the final signal for performance of the task. Post-processing steps may also be included in a controller module to have a smooth output.

Figure 2.8 illustrates a myoelectric control system, which performs tasks based on the output gained by applying pattern recognition to the input EMG signals.

The effective improving factors in pattern recognition-based myoelectric controllers are mainly the feature extraction and classifier modules. Thanks to the great progress in the field

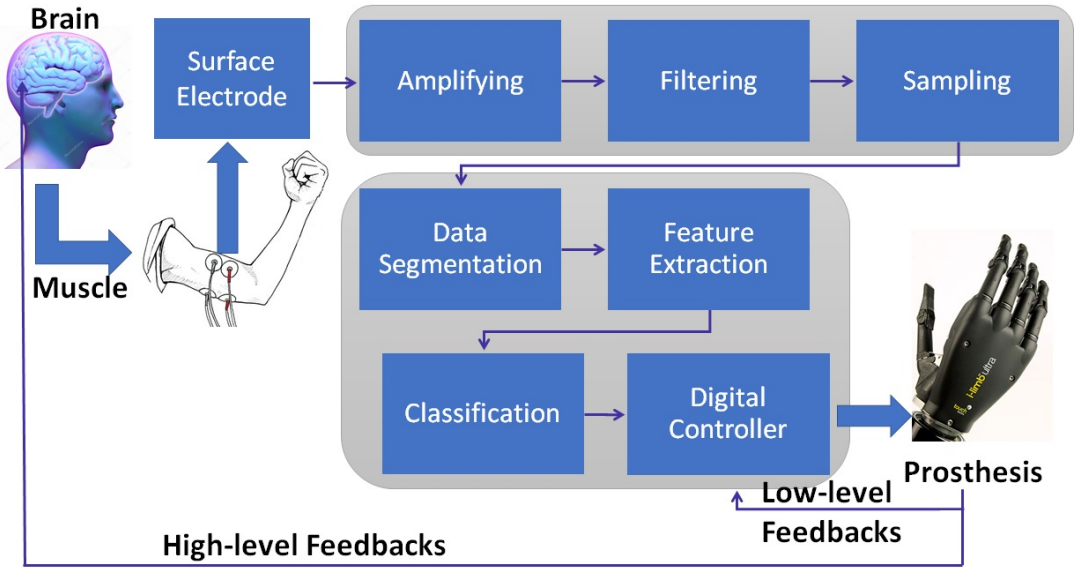


Figure 2.8: A pattern recognition-based myoelectric control system including its functions and modules [9].

of machine learning [52,53], both feature extraction and classification techniques has improved exceedingly during recent decade [33,54]. There are a variety of techniques to perform feature extraction. To name a few, variations of fast Fourier transform (FFT) [55, 56], entropy-based solutions like hidden Markov model (HMM) [57] and feature projection (FP) methods such as principle component analysis (PCA) [56, 58] and linear discriminant analysis (LDA) [59] are among the common feature extraction methods [33] for EMG signals.

Advanced classification techniques involve both supervised and unsupervised classifiers such as Bayesian classifier [60], Fuzzy logic [61], support vector machine (SVM) [62], k nearest neighbors [63], LDA [62] and a large number of other types of classifiers [33].

Some works also benefit from end-to-end systems such as neural networks [52–55, 64] and convolutional neural networks [15, 25, 65–70], which gained high attentions in various areas recently. Artificial neural networks benefit from great learning capacity [52–54] and can be used for feature extraction, classification and end-to-end application of both tasks via a single network.

Considering these developments in pattern recognition-based myoelectric controllers, great offline classification performances as high as 90% classification accuracy with large number of possible classes (> 10) [4, 71] was achieved. These promising techniques however are rarely implemented in any commercial systems [4, 37] as they still mostly suffer from specific limitations, which hamper their introduction to industrial systems [72] (except for the COAPT system released recently [36]). Two major factors in creation of these limitations are EMG signal char-

acteristics' variation and the limb position effect. Both issues are caused by the assumption that users can produce distinguishable signals in a reproducible manner, which in practical set-ups is not feasible. The former refers to changes in the EMG signals caused by parameters such as fatigue, sweating and electrode displacement [4, 72]. The latter attributes the performance drop in real-world trials to the limb position variations [73]. In addition, the control is still sequential preventing from execution of simultaneous motions [4].

Finally yet importantly, the users need some training sessions with pattern recognition-based systems in order to learn to produce distinct commands suitable for the pattern recognition structure. The more sophisticated the system gets, the higher sensitivity it suffers from, leading to the limited DoFs used by users at the end. These issues can potentially be overcome by a change of focus and augmentation of EMG signals with extra modalities [72].

Simultaneous and Proportional Control (SPC) Approaches

The SPC methods are provided as a solution to unnatural control in pattern recognition-based approaches. To this end, regression-based myoelectric control schemes are offered to enable concurrent control of multiple classes. Some methods involve learning a regression function based on kinematics provided by the mirror hand [4]. Both supervised and unsupervised learning techniques can be utilised for learning a map between kinematics and EMG signals.

2.2.4 Multimodal Schemes

Although pattern recognition and SPC solutions provide better dexterity for the amputees than conventional control schemes and yield highly promising results, these methods are still incapable of catching up with the functionality of modern commercial artificial hands. Therefore, the control of commercial hand prostheses is still based on the conventional control schemes. As such, the user can benefit from comparatively simple and intuitive control, while increase in number of DoFs brings about several challenges such as the requirement of repeated co-contractions, which is followed by user's fatigue. Another limitation in conventional control systems can be the EMG corsstalk and variable amplitude estimation [4]. Considering all the commercial prosthetic hand controllers, natural, proportional and simultaneous control of a large number of degrees of freedom is still not available [37].

As an alternative solution, additional modalities are employed to provide extra information for grasp performance. Some of these modalities are used as substitutes to EMG signals and some others are used as an augmentation over the present signals. Some examples include skin

movement analysis via accelerometry signals [47,51], force myo-graphy [44], ultrasound imaging [45], near-infrared spectroscopy (NIRS) [46], use of radio-frequency identification (RFID) tags [43], arm movement trajectory and inertial measurement (*e. g.* i-moTM), electrooculography (EOG) [48] and computer vision [10, 11, 41, 42, 49, 50].

All the previous approaches, namely the conventional, pattern recognition and SPC, utilise the myoelectric controller passively, while letting the user to be responsible for generating all the signals and execute all the actions. In several multimodal approaches however the burden is moved to the controller, which can perform autonomous decisions. In this way, the user benefits from a *semi-autonomous control*, in which he/she acts as a supervisor for the tasks carried out by the controller. The main challenge in such a system can be the user's preference for being engaged in the grasp act performance. Thus, there is always a trade-off between user and controller task specification [4].

Sensor Fusion

In spite of the efforts for development of sophisticated advanced control strategies with higher performances, Cipriani *et al.* [74] demonstrated that amputee users prefer systems with less complications. That is why commercial hand prostheses mostly benefit from conventional control systems [5, 6, 8]. In parallel to this finding, the usage of additional modalities to EMG data mostly endeavors to ease the task of grasping on the user's part.

The very first non-invasive solutions for provision of user with less burdensome tasks were developed by Tomović *et al.* [75] and later Nightingale *et al.* [76], in which the hand is augmented with pressure/touch sensors to produce an appropriate grasp type.

Fougner *et al.* [47] suggested that the addition of an accelerometer to the forearm in combination with the EMG electrodes provides supplementary information to the data provided by sEMG (surface EMG). These sensors are not only economic, but also lead to a more simplified myoelectric control system.

Krasoulis *et al.* [77] improved the previous system further with addition of further modalities recorded by gyroscopes and magnetometers [78]. They concurrently recorded sEMG signals by 12 EMG electrodes and acceleration, rotational velocity and orientation by integration of each EMG sensor with a 9 DoF inertial measurement unit (IMU) sensor. Each IMU sensor consists of an accelerometer, a gyroscope and a magnetometer. It was indicated that IMU sensors provide helpful sensory data and enhance the performance of sEMG signals for grasp classification using a multiclass linear discriminant analysis classifier both in online and offline

analyses [78].

As one of the works benefited the most from sensor fusion Marković *et al.* [79] offered a novel control technique for prosthetic hands employing data fusion. Their system is facilitated with a variety of modalities, namely, myoelectric recording, computer vision, inertial measurements and embedded prosthesis sensors (position and force) to provide real-time simultaneous, proportional and semi-autonomous control of an artificial hand. The RGB-D information recorded by depth camera contributes to estimation of objects' attributes (shape, size and orientation), which can further be combined with prosthesis orientation and user behaviour via inertial sensing. Such an advanced platform led to less than 1% cumulative trial failure rate. It is worth to note that only palmar and lateral grasps were performed during the experiments.

Vision

Although vision is also considered as a sensory information, a separate section is specified to relevant works involved with computer vision due to the significant progress of this field and its effect on multi-sensor artificial hands.

The cognitive vision system developed by Došen *et al.* [10,41,80] is one of the first vision-based control solutions for prosthetic hands. Continuing their novel approach on artificial hands, several vision-based hand control systems have been designed since then [11,42,49,50]. In the following, more details over these vision-based methods are presented and the main aspects of each system are highlighted.

Cognitive Vision System for Control of Dexterous Prosthetic Hands The research done by Klisic *et al.* and Došen *et al.* [10,41,80] follows similar approach, therefore they are all covered in this section. A dexterous hand (CyberHand) is provided with vision and an autonomous controller. After triggering the hand and controlling its orientation by the user, a cognitive vision system (CVS) captures an image of the object and records the measured distance to the target to be fed to the high level controller. The high level controller automatically proposes the best grasp type and size through a rule-based reasoning structure. The selected grasp is then implemented by the embedded hand controller using closed-loop position(force) control. Figure 2.9 demonstrates the control system architecture for the proposed method in [10].

The CVS includes a low-cost web camera, an ultrasound distance sensor and a laser pointer. After processing the captured image of the laser-dotted object, the measured distance and calculated dimensions provide an estimation of the object's size. The estimation is used as an input for the fuzzy control, in which based on the size of the object, specific grasp type and aperture

size is chosen.

This approach was tested on 13 healthy subjects using CyberHand [81] for 18 objects placed at two different distances. In 84% of the trials the system proposed the correct grasp type and size besides reducing the burden on user. Therefore, one can conclude that augmenting current hand prostheses with a controller empowered by vision is effective and can improve the myoelectric prosthetic hands. Štrbac *et al.* [82] also benefited from a similar approach using a stereo vision system consisting of two CCD cameras and a laser diode, which led to $\sim 90\%$ grasp classification accuracy.

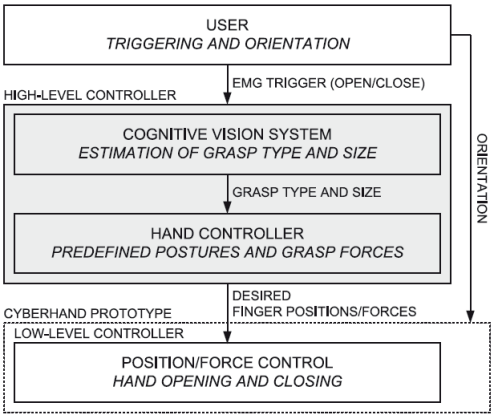


Figure 2.9: Control system architecture developed for vision-based prosthesis controller designed by Došen *et al.* (taken from [10])

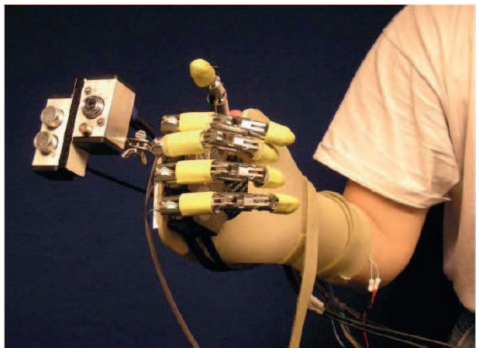


Figure 2.10: CyberHand attached onto an orthopaedic splint facilitated with cognitive vision system (CVS) and EMG electrodes (taken from [10])

Utilisation of Gaze Information for Efficient Grasp Prediction Giordaniello *et al.* [50] acquired sEMG and data glove singals together with visual scene recording and eye tracking data during several grasping experiments. Gaze tracking is accomplished via fixation pointing such that the user stares at the target object for a specific period of time.

They indicated that sEMG data includes sufficient information for hand movement and object classification. Their experiments indicate that gaze information together with visual information over the field of view can be highly beneficial for object detection and recognition and therefore boosting the performance of hand prostheses by enhancing their robustness [50].

Continuing the previous work, Gigli *et al.* [49] implemented a multimodal system consisting of sEMG and visual cues to improve the performance of hand movement prediction [49]. Gaze information is used for target object segmentation, which is then given to a deep convolutional network (VGG-16 pretrained on ImageNet [15]) for high level feature representation. These features are integrated with sEMG features and given to a classifier for grasp prediction.

The results presented in [49] suggests that augmentation of sEMG with visual information elevates the grasp classification accuracy from 80% to 84%. The procedure of blending sEMG with visual cues includes weighting each modality such that the best combination of features are achieved. During experimental evaluations, 60% sEMG and 40% visual information worked the best for most of users. It is worth noting that the experiments were done with able-bodied subjects.

Stereovision and Augmented Reality (AR) for Closed-loop Control of Grasping in Hand Prostheses A semi-autonomous prosthetic hand control mechanism was developed in [11] with stereovision cameras and augmented reality (AR), as shown in Figure 2.11. The control in this system consists of two levels. The high level control includes the controller and stereovision cameras, which leads to autonomous control of the system and outputs the hand grasp type, size, and orientation. The low level control involves the user as a supervisor through the AR glasses and embedded stereo cameras. The user is able to correct the decisions suggested by the controller by flexion (extension). To enable such procedure, the user is provided with a visual feedback of the status of the hand and online correction.

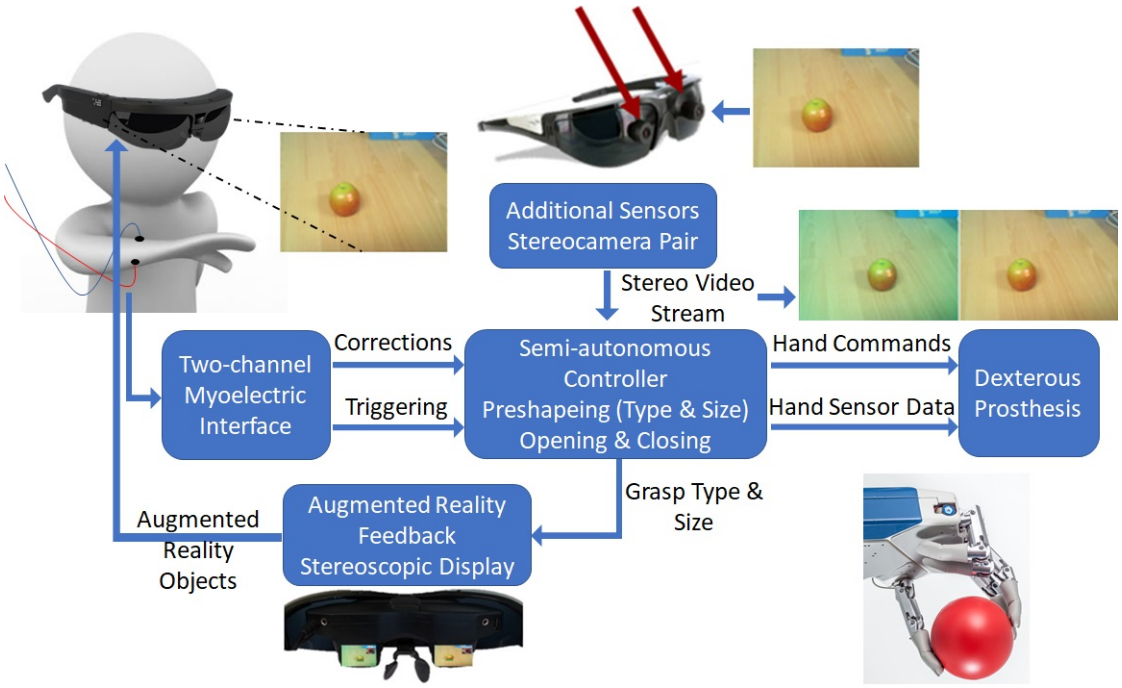


Figure 2.11: System architecture for using stereovision and AR for closed-loop control of hand prosthesis [11].

Two input images and the current preshape of the prosthetic hand are provided for the computer vision module through the AR glasses and preshape control module respectively. The two input images are passed through further processing techniques, including depth estimation,

segmentation, point cloud generation, geometrical model approximation and finally virtual object generation. In this way object properties are precisely estimated. Further usage of current prosthesis preshape with the visual data provides the AR feedback, which is embedded over the input stereo images presented to the user through the AR glasses.

This system can reduce the cognitive burden on the user by using two data modalities, namely myoelectric signals and artificial vision. However, it is debatable whether wearing an AR glasses in preferable for amputee users in real life or not. The complexity of the system is another concern according to [74].

Vision for Grasping

Among the available multimodal solutions for myoelectric hands, the wealth and performance of vision-based approaches outweigh the other solutions [10,11,41,42,49,50]. This observation could be explained as follows:

Firstly, vision is an economical modality to enhance myoelectric control. Addition of a camera to an artificial hand does not demand fundamental amendments to the design of these hands and the required modules are inexpensive. Secondly, computer vision is developing rapidly, while computer vision solutions are blending into everyday life structures. Additionally, vision is a rich and robust source of information and unlike EMG, a small disruption to the input signals does not cause a significant change in the overall representation. Moreover, visual data is inherently distinct from EMG and can be employed as a supplement to produce more robust decisions within a multimodal framework. Finally yet importantly, visual information plays an indispensable role in the grasping act. Humans almost always have a look at their object of interest before grasping it, which provides them with sufficient knowledge for deciding on the grasp act and producing the right command.

It is therefore rational to utilise vision together with EMG data for production of more robust and accurate predictions. The next section provides some background on computer vision and the most recent research works in this field.

2.3 Recent Trends in Computer Vision

To emphasise on the impact of computer vision and its capabilities, a comprehensive investigation through computer vision and its progress over the recent years is provided in this section.

Computer vision has become an indispensable field in science and technology through providing human-like capability of seeing and visually sensing the world for a machine or a com-

puter. An autonomous system can be provided with artificial vision when the information from an image or a video stream is extracted and processed efficiently. Thanks to numerous applications of computer vision in different aspects of our lives, such as industry, safety, health, security and recreation [83], this field has been developed considerably during the recent decades [84].

In today's definitions it is hard to differentiate the fields of image processing, computer vision, pattern recognition and deep learning [14] with specific margins. Deep learning systems specifically aim at mimicking human brain [85]. There are of course a wealth of computer vision techniques that are not involved with deep learning, but the more deep learning progresses, the more it is applied to its relevant fields such as computer vision. Despite all the advances of both fields, computer vision technologies are still too far from human-like processing of visual information [84]. In the following some computer vision methods helpful for the task of grasp/object recognition are presented.

2.3.1 Feature Extraction

Similar to pattern recognition techniques, feature extraction methods in computer vision also explore specific properties of every image point and compute an abstract representation of image information based on those properties [86]. These features can vary from simple attributes such as edges, corners, blobs and ridges to more sophisticated ones.

In order to represent an image in an object recognition task, there are three approaches: model-based, shape-based and appearance-based [87]. The appearance-based methods can be divided into two categories: local and global. Local features are properties of an image in a single point or region, particularly colour or gradient. In contrast, global features represent an image as a whole. Appropriate local features are invariant to changes in the scene, illumination, rotation, and size for accurate object recognition. By utilising local features for a complex or combined description of an image, the feature descriptors can be constructed [87].

The feature extraction methods can be divided into two main categories of hand-designed and learning based approaches.

Hand-designed Features

During initial trials of scientists for image feature extraction, several image matching algorithms, mostly based on corner and edge detectors have been developed. The early corner detectors are mostly focused on identifying image locations with large gradients in all directions (*e. g.* Harris corners) [88]. Several methods has been suggested for providing invariance

for these features, such as matching with correlation windows and rotationally invariant descriptors [89]. Further developments led to current local feature detectors and descriptors [90, 91].

Local Feature Detectors and Descriptors These methods extract the main features of the reference image and the objects to be matched and find all the possible matches between them. These features mostly include surface patches, corners and linear edges [90, 91]. Among all the feature extraction techniques, the scale invariant feature transform (SIFT) [92] is discussed here as it is still counted as one of the most robust and powerful feature descriptors.

Scale Invariant Feature Transform (SIFT)

The SIFT algorithm [92] transforms an image data into scale-invariant coordinates relative to local features. In addition to scale, SIFT features are invariant to rotation, affine distortion, change in 3-D viewpoint, noise, and illumination. In order to obtain the SIFT features, there are four steps:

1. **Scale-space extrema detection:**

   The scale space, a representation of image at different scales, is constructed and all the scales and locations are examined as a difference of Gaussian (DoG) function is applied for detection of the points with potential of invariance to scale and orientation (DoG extrema).

2. **Keypoint localisation:**

   Using Taylor expansion for the DoG function, the location of extrema is obtained. Unstable extrema with low contrast and also edge responses are discarded. Consequently, the number of selected points is reduced to a selection of good keypoints.

3. **Orientation assignment:**

   Gradient magnitude and orientation is calculated for each sample point in the region around each keypoint. The histogram of local gradient directions for regions around each keypoint is computed at the selected scale. The direction of local gradients is determined by the peaks in the orientation histogram. Each keypoint is assigned with a location, scale and orientation. This location, scale and orientation assignment is used in all the future operations on the image, which yields invariance to these transformations.

4. **Keypoint descriptor:**

In the final stage, a highly distinctive descriptor is computed for the local image region that is invariant to local shape distortion and change in illumination. Having a large database of features provides a good probability of finding the correct match for a single feature.

Figure 2.12 shows the corresponding steps to be taken in order to obtain SIFT detectors and descriptors. At least three features are required to find a correct matching feature between two image objects. Figures 2.13 and 2.14 demonstrate extracted SIFT features for a sample object and matching between this object and similar one in presence of clutter and illumination, scale, rotation and view point variations. As illustrated in Figure 2.13, several features are extracted for the object, while some are more distictive and of more importance. Although the scene image to be matched includes a different scissor in a different environment, the object is properly matched with its reference.

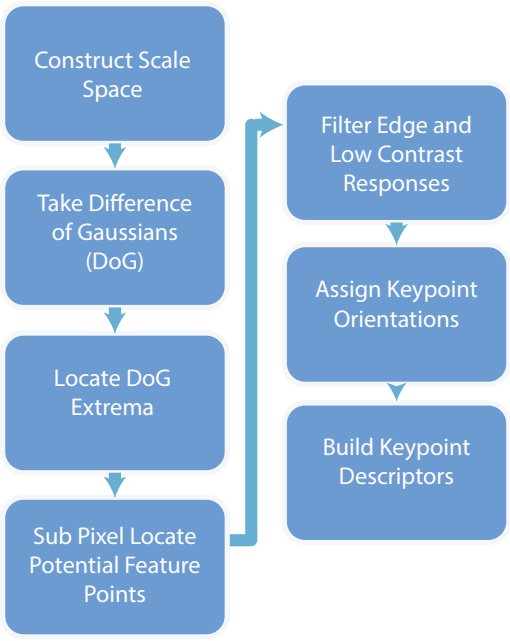


Figure 2.12: SIFT algorithm overview

DEspite SIFT being highly beneficial in applications such as panorama stitching, object matching, image retrieval and visual navigation, it is not useful for real-time applications due to the high computational complexity of the algorithm. There are similar feature detectors or descriptors with faster processing times such as SURF (speeded up robust features) [93] or HoG (histogram of oriented gradients) [94]. These algorithms are less computationally burdensome than SIFT, while they are still not as efficient as learning-based methods in real-time tasks. Additionally, they are sensitive to noise and background, blurring, lighting changes and are effective only for specific applications such as human detection [94].

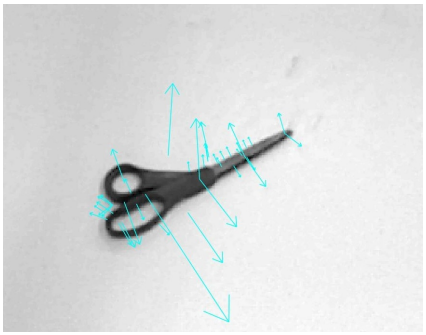


Figure 2.13: Extracted SIFT features for a sample object (a scissor).

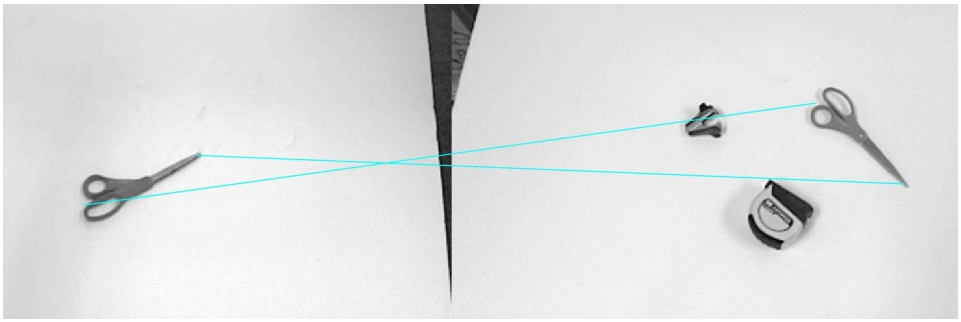


Figure 2.14: Feature matching for a scissor with variations in scene, view point, position, illumination and scale.

Furthermore, feature descriptors may be effective for detecting *seen* objects, but they cannot provide a match for objects that are never seen before. That is why, there is a limited usage of hand-designed feature descriptors in practical applications for object detection/recognition.

Learning-based Feature Extraction

Artificial Neural Networks (ANNs) Neural networks, biologically inspired from human brain's neural networks, have the ability of estimating a model (function) when they are given a set of training inputs and their relevant outputs. That is, if properly trained, a neural network can even predict the output for an unseen input. Figure 2.15 represents a shallow neural network architecture. It includes three input units, four hidden units and two output units.

Deep Neural Networks (DNNs) Images are usually large scale matrices. In addition, standard (shallow) neural networks usually include only one hidden layer. Due to the high dimensionality of data in images and demanding high capacity to learn various features, more than one hidden layer in a neural network is usually required that leads to the concept of deep neural networks. Having more layers, more complex features are obtained, since each extra hidden layer produces more sophisticated features and learns higher amount of abstraction than the

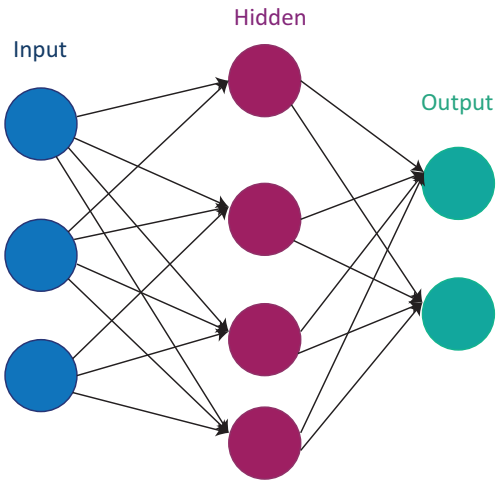


Figure 2.15: A shallow artificial neural network (ANN)

previous one. The more complex the features are, potentially the more accurate predictions and more complex tasks are performed. Nevertheless, adding hidden layers is not always working as efficient as it is expected due to some problems such as difficulty in training, slow convergence and over-fitting. Deep networks are also computationally intensive especially when the input is an image. There have been a variety of solutions that are suggested to address this problem. To name a few, deep boltzmann machines (DBMs) [95], Deep auto-encoders [96] and convolutional neural networks (CNNs) [12, 25, 97–100] are among the popular deep learning solutions for object recognition [101].

Compared to other deep learning techniques, deep CNNs benefit from a great ability of learning, they are well suited to image classification problems and indicated noticeable performance in a variety of challenging object recognition tasks [15]. There are several characteristics of CNNs, which led to such performance, which are further explained in the next section.

Convolutional Neural Network The CNN structure is inspired from the visual cortex, in which cells are sensitive to small sub-regions of the visual field called receptive fields. Figure 2.16 represents the CNN architecture proposed by LeCun in [12].

The CNN structure employs three architectural ideas, which provide it with some extent of shift, scale and distortion invariance: local connectivity, parameter sharing and pooling or subsampling. To illustrate, local connectivity is based on the idea of receptive fields. It refers to the idea that images are stationary and there are patches of an image repeating along the image. Parameter sharing deals with feature maps, in which the units share the same parameters leading to reduction in the number of parameters.

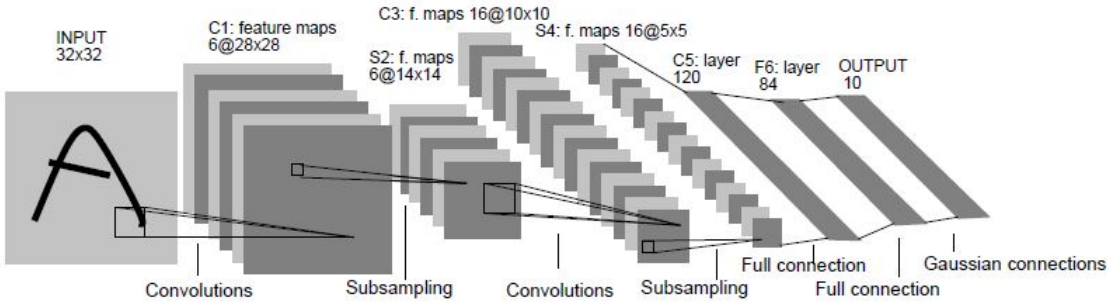


Figure 2.16: Architecture of LeNet-5, a CNN, used for digit recognition by LeCun. Each plane is a feature map, *i. e.* a set of units whose weights are constrained to be identical (taken from [12]).

There are two types of layers in a CNN: convolutional layers and pooling layers. The former utilises different number of filters for extracting desired distinctive features, such as edges. These filters are applied across the image and the result is the feature maps that have the same number as the filters. Each feature map extracts a certain feature regardless of its location in the image. Subsequently, a non-linear down-sampling is often applied to the feature maps in the next layer (pooling). There are different approaches for this non-linear down-sampling, where max pooling and average pooling are the most prevalent ones. Firstly, each feature map is partitioned into a set of non-overlapping rectangles. Then, if max pooling is the technique applied, in each sub-region, the maximum value is selected as a representative of the whole rectangle. Otherwise, for average pooling, the average of all the pixels in each rectangle is selected. Pooling provides translation invariance as small translations occurring in the same pooling region (rectangle) are neglected. Eventually, the network is followed by a fully connected layer for classification of outputs.

Further mathematical background on the CNNs, their building blocks, training procedure and layer varieties are explained in the Appendix A.1.

2.3.2 Deep Learning

The main ideas of deep learning were available for several years, while two factors hampered its proliferation, namely data availability and computational scale [13]. The former is still a challenge for many applications, but for many other ones thanks to today's digitalised era more data is available since 1998 [12]. Figure 2.17 indicates the favorable effect of data on deep networks and how it can affect the achievable performance. The computational capacity of devices also has thrived increasingly with graphical processing unites (GPUs). Overcoming

these constraints, deep learning techniques has surged exceedingly over the recent decade and solved many tasks that were intellectually burdensome for humans. On the other hand, tasks done by human intuitively are still great challenge for the deep learning-based structures [14].

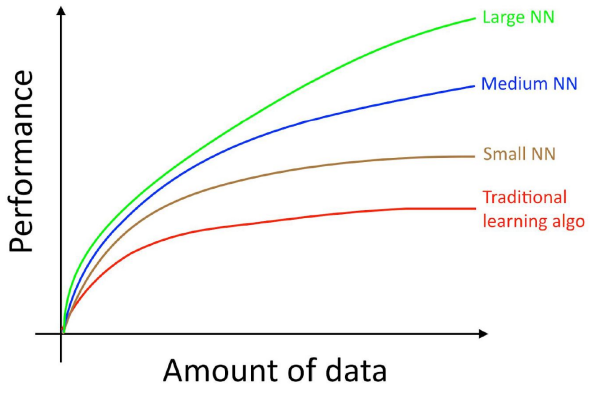


Figure 2.17: The relation between the amount of data and performance in learning algorithms with different depths [13]

As mentioned, deep neural network models are mainly extended or more complex versions of neural networks. The effect of depth of a neural network can vary based on the application, but generally it leads to more abstraction. Figure 2.18 illustrates the importance of network depth by showing the complexity of features learned within each layer. It can be observed that the initial layers focus on more detailed and at the same time simpler types of features such as edges, corners and blobs, while the concern of deeper layers is the overall information relevant to each class such as particular object patches.

When having shallow networks or using hand designed features, it would be highly difficult to find a mapping from an image, including high number of pixel values, to a predefined class. Deep learning can ease this mapping task by providing a series of nested simpler mappings [14].

2.3.3 Object Recognition

As an exploration to better substitutes for feature descriptors such as SIFT [92], SURF [93] and HOG [94], Fukushima *et al.* suggested *NeoCognitron* [102] as a biologically inspired hierarchical and shift-invariant pattern recognition model to extract richer visual features than the mentioned methods. One limitation of this model is the lack of supervision, which hinders an end-to-end learning. Rumelhart *et al.* [103] and later Lecun *et al.* continued on working on this problem until 1989, when Lecun *et al.* [104] indicated the potential of stochastic gradient descent (SGD) via backpropagation for training an extended model of neocognitron called con-

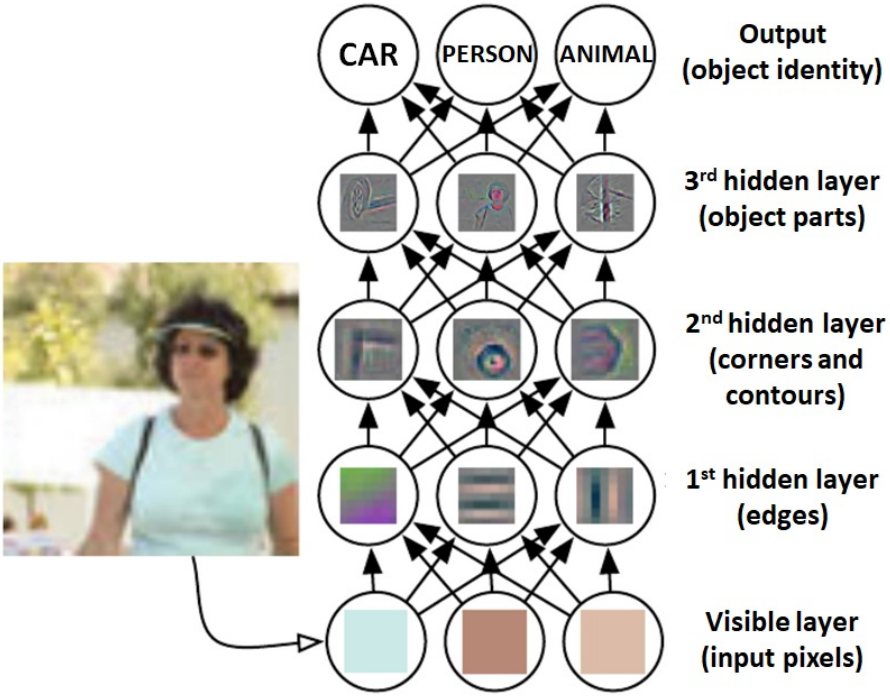


Figure 2.18: Depiction of a deep learning model including the information learned within each level of abstraction (layer) [14]

volutional neural networks. Some of the most important neurally-inspired models in CNNs are NeoCognitron [105, 106], HMAX [107, 108] and LeNet-5 [12].

Practical utilisation of deep learning started with development of recognition challenges, such as the MNIST [109], a large dataset of handwritten digits (0-9). Lecun proposed a deep learning architecture called LeNet to perform on this dataset [12]. The importance of this architecture however was not widely appreciated until 2012. Similarly, initial deep learning solutions proposed benchmarks based on the MNIST dataset. Later in 2009, ImageNet dataset was gathered to confront deep nets with a more challenging recognition problem. ImageNet consists of thousands of images in 1000 categories. Alex Krizhevsky*et al.* [15] won this challenge in 2012 with an innovative network architecture known as *AlexNet*, which outperformed the previous solutions by a significant margin. Since then, CNNs has become the gold standard for image classification. Figure 2.19 represents the architecture of AlexNet, the great breakthrough for its time, which led to the further proliferation of deep networks.

Continuing the great success of AlexNet, a variety of network architectures were designed and got popular, to name a few, VGG [110], Inception [111] and ResNet [112] are among the famous deep networks with high performances. Each architecture brings up particular design considerations, which facilitates the learning procedure within the network and boosts the over-

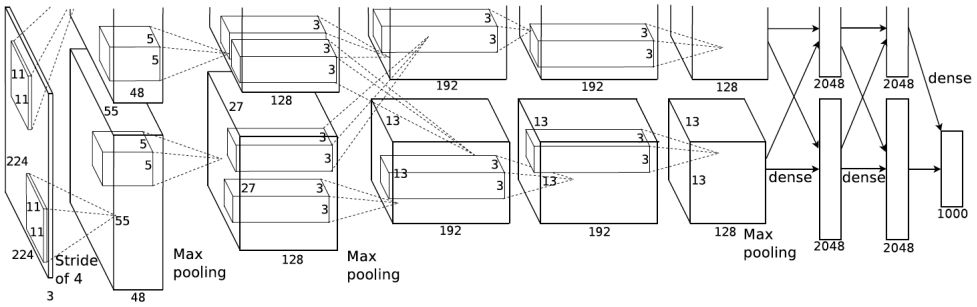


Figure 2.19: The architecture of AlexNet [15]

all object recognition performance in ImageNet. For instance, ResNet architecture features residual connections, which ease the possibility of having deeper models with less parameters while improving the gradient flow.

Another trend in deep learning is to use transfer learning [113] to boost the learning performance of a network. The common practice is to use randomised weight initialisation for training a network. With transfer learning however initial weights can be richer and extract more informative features that can accelerate the training procedure and provide better convergence.

To further illustrate, a trained deep neural network can act as a feature extractor when the output layer is removed. If the network is trained properly, these features are sufficiently rich to be given to a classifier for finalising the classification task. As ImageNet is a huge dataset with a large variety of objects, the networks trained on ImageNet learn helpful features, which are much richer than the randomised weights normally used for training from scratch [113]. Hence, a deep network trained on ImageNet can already be a good feature extractor. When having a different recognition task from that of the ImageNet, these weights are still good initialisations. They can be further updated by fine-tuning the network on a new task/dataset. This approach not only leads to a boost in overall performance but saves a large amount of time as the network converges to a solution in a shorter time.

2.3.4 Object Detection

After the great success of deep CNNs for image classification tasks [15] efforts has been done to exploit them for object detection. As some initial endeavors, sliding-window detectors were implemented [114, 115], in which object detection is modeled as a classification problem. The spatial resolution within the sliding-window and the delay during the slide procedure are limi-

tations of this approach.

Firstly presented by Girschick *et al.* [16] in 2014, regions with the CNN features (R-CNNs) outperformed state-of-the-art approaches in object detection tasks, specifically on the well-known Pascal VOC object recognition challenge [116]. Two papers were subsequently published with the focus on speed enhancement of this approach: Fast R-CNN [117] and faster R-CNN [118].

The main contribution of the R-CNN structure is the adaptation of a deep neural network trained on image classification to perform object detection. This adaptation can be summarised in few steps: 1) receiving an input image, 2) production of region proposals (regions of interests (RoIs)), 3) feeding the RoIs into the network separately, which leads to a vector of values in the output, 4) training a classifier to propose a label and confidence for each RoI. Figure 2.20 indicates the basic idea of the first R-CNN.

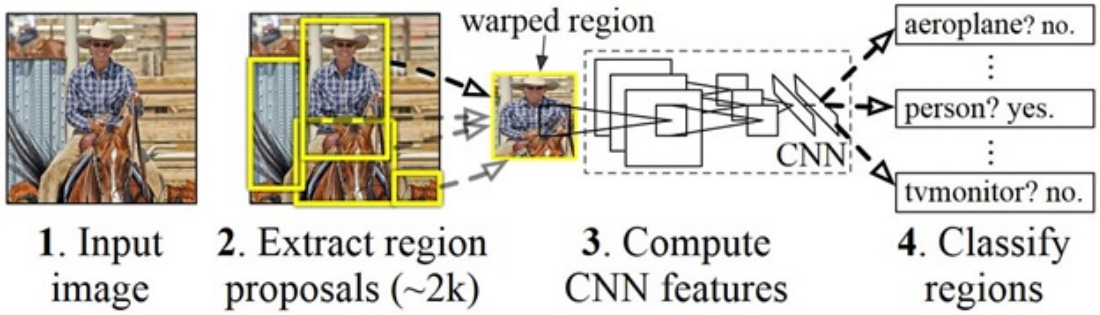


Figure 2.20: The representation of the first R-CNN developed by by Girschick *et al.* [16].

The R-CNN structure performs well in terms of accuracy. However, the approach is computationally expensive, as for each RoI the neural network has to be evaluated once. To resolve this issue, fast R-CNN [117] was designed such that it evaluates the convolution layers once per image by using the RoI pooling layer. This pooling layer projects the RoI onto the convolutional feature map and generates the ideal output size of the subsequent layer by performing max pooling. This approach leads to about 213 times test speed-up and 9 times training speed-up without losing accuracy. The faster R-CNN [118] builds upon the fast R-CNN and replaces the region proposal method with a CNN that itself learns the region proposals.

More recently, "you only look once" (YOLO) [119], region-based fully convolutional networks (R-FCN) [120] and single shot multibox detector (SSD) [121] and mask R-CNN [122] were suggested as the best available object detection architectures. YOLO and SSD are among the popular single shot detectors, which treat the object detection problem as a regression one in contrast to the region-based detection methods mentioned, in which detection is handled as

a classification problem. Picking each method depends on the desirable accuracy and speed, as region-based methods are more appropriate for achieving higher accuracy while the single shot detectors are more concerned with the speed.

2.4 Recent Trends in Robotic Grasping

Unlike the research on limb prosthetics, computer vision has been widely used in robotic grasp and object manipulation tasks [17, 26, 123–133]. An autonomous agent requires the ability to grasp items to be able to interact with its surrounding environment. Specifically, many envisioned applications in the fields such as personal robotics and advanced industrial manufacturing require grasping and manipulation of objects as a necessary skill. Nevertheless, robotic grasping is a highly challenging task, as it involves several components to be applied simultaneously, namely perception, planning and control. Even under simplified working conditions and scenarios, robots are far from human performance in grasping. Indeed, humans can reliably grasp a vast class of objects characterised by complex shapes independent of their position and orientation, while robots are still struggling at this task.

For what concerns the perception part, localisation of reliable and effective grasping points on the object surface from visual data is crucial to increase the chance of a successful grasp by means of the employed end effector, such as a robotic hand or a gripper. This visual task has gained great attention in recent years, with a wealth of methods proposed in literature [17, 26, 112, 123–137] and the creation of specific benchmarks such as the one proposed in [17] to evaluate the performance of various approaches.

Before the advancement of deep learning and its success in computer vision applications, grasp estimation solutions were mostly based on analytic methods [138]. Some of these approaches, such as Graspit! [133], are dependent on the presence of a full 3-D model to fit a grasp to it, thus not feasible for real-time applications. Kootstra *et al.* [123] developed an *early* cognitive vision architecture for grasping unknown objects. Without any segmentation or preprocessing steps, they were able to generate two- and three-finger grasps based on contours and surface structure provided by stereo cameras. With further improvement of depth sensors, there are also recent methods that leverage geometrical information to find a stable grasp point using single-view point clouds [129].

In addition, the combination of both learning techniques and 3-D shape information has led to interesting results. Saxena *et al.* [125] provided the capability of grasping novel (unseen) objects for robotic hands by utilising a stereo camera. Without building a 3-dimensional model,

they estimated the 3-D location of the best grasp by triangulation. The grasp location estimator algorithm was trained on synthetic images in a supervised learning regime. Later on, Kopicki *et al.* [124] provided a one-shot learning mechanism for recognising the most appropriate grasp for novel objects. They generated thousands of grasp candidates for images taken by a depth camera and optimised the combination of two learned model types: a contact model and a hand-configuration model. In [132], a deep learning based approach is used to estimate a 3-D model of the target object from a single-view point cloud and suggest a grasp using 3-D planning methods such as Graspit! [133]. Mahler *et al.* [135] developed a quality measure to predict successful grasp probabilities from depth data using a CNN. Asif *et al.* [134] extracted distinctive features from RGB-D point cloud data using hierarchical cascade forests for recognition and grasp detection.

The most recent robotic grasp estimation research works are focused solely on deep learning techniques. Lenz *et al.* [17] were one of the pioneers in applying deep learning methods to robotic grasping problems. The method uses a two-step cascade system with two deep networks operating on RGB-D input images. The first network predicts the candidate grasp rectangles and the second one chooses the optimal grasp points. The main limitation of this model is the comparatively long computation time for providing the optimal grasp (13.5 seconds). Figure 2.21 presents the grasp detection procedure implemented in [17].

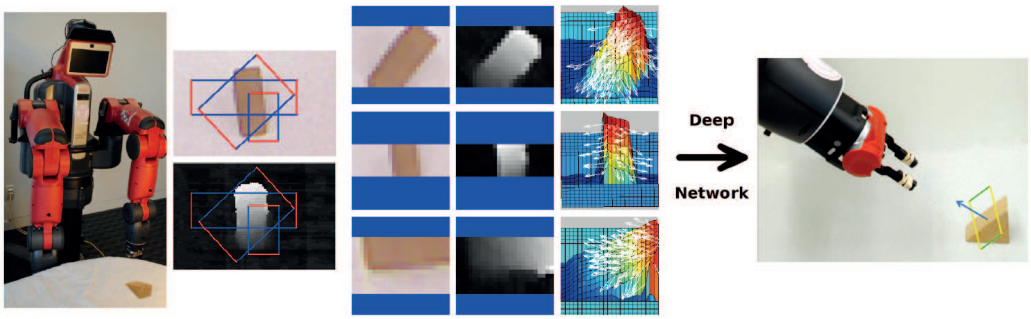


Figure 2.21: Detecting and executing grasps in [17] (taken from [17]).

Wang *et al.* [136] followed a similar approach using a new multi-modal deep CNN model. Another work [137] used RGB-D data to first extract features from a scene using a ResNet-50 architecture [112] and then a successive shallower convolutional network applied to the merged features to estimate the optimal point of grasping. A recent work in robotic grasp detection has also built upon object detection method in YOLO [118, 139] to directly predict candidate grasp bounding boxes. More recently, Guo *et al.* [127] came up with a hybrid deep network combining visual and tactile sensing for robotic grasp detection.

2.4.1 Robotic Manipulator versus Hand Prosthesis Control

Although research in robotic hands takes advantage of the cutting-edge developments of machine vision more competently than the hand prostheses research, most works in this field are focused on two- or three-finger grippers. Therefore, the main perception problem of robotic grasping research is the detection of grasp points rather than the type of grasp, while for hand prostheses the type of grasp is the most crucial concern of recent research to provide natural actions. Another significant distinction is that amputees' limitation is mainly the arrangement of fingers to perform an appropriate gesture, while they can plan the grasps dexterously. Conversely, robots require to plan for even the simplest grasp acts. Having these distinctions in mind, the design of vision-based prosthetic hands should be able to exploit the machine vision advancements to their highest potential, while taking care of the user in every step. That is, simplicity, speed and user-friendliness would be the main features of a hand prosthetic and unnecessary complications may be eliminated.

2.5 Conclusion

As discussed in Section 2.1, prosthetic hands can significantly contribute to boosting the quality of life for amputees. The performance of current artificial hands however is not fulfilling the amputees' expectations yet. Therefore, in Section 2.2.4, several solutions for enhancement of present commercial prosthetic hands are suggested from literature. A great deal of research in hand prostheses is inclined to augmentation of current myoelectric hand prosthetics with additional modalities [10, 11, 41–51]. Among those, vision-based prosthetic hands are gaining popularity [10, 11, 41, 42, 49, 50] thanks to simple and economic production of a vision module, presence of advanced methods in computer vision and the inherently distinct, robust and natural representation of information via images compared to EMG. As amputees face difficulties in transferring their desired command to a prosthesis hand to perform a grasp, the visual data can be fed into the artificial hand for an autonomous decision relieving them from this burdensome step.

The research works in both domains of prosthesis hands and robotic grasping support this claim that vision is vital for better interaction of humans with their surrounding and therefore can be beneficial to improve the control of a prosthesis hand while grasping. Consequently, the focus of this work is augmentation of commercial myoelectric hand prosthesis with visual data and using this modality such that the performance of artificial hands is improved.

Chapter 3

Convolutional Neural Networks for Grasp Classification

This chapter presents a computer vision structure employing deep learning to perform grasp classification in artificial hands. The goal is to offer a system through which amputees can easily grasp and move common household objects via a two-channel myoelectric prosthetic hand. To gain this goal, a deep learning-based artificial vision system is developed to augment the grasp functionality of a commercial prosthesis such that objects are classified merely based on their appropriate grasp pattern avoiding object category recognition or dimension measurement. This approach is first implemented and evaluated in offline to investigate its feasibility and capabilities. Further real-time experiments with and without amputee subjects are performed to evaluate the performance of the system in real-world scenarios. Comprehensive analysis of all the achieved results are provided for better decision over the capacity of the proposed platform and possible improvements.

3.1 Introduction

3.1.1 Object Recognition

Visual data, as mentioned in previous chapters, can be highly beneficial as an additional modality to EMG data for performance elevation in artificial hands. Thanks to the advancements of object recognition techniques, the visual information of an object can be simply captured, given to a deep network and classified into its appropriate category. The main limitation of this approach is that for a classification task, the classes should be already known by the model. That is, when a deep network trained for object recognition faces a new item not included in the

possible output categories, it mistakenly classifies the item as one of the present classes. Hence, including all the graspable objects as output classes of a deep network is highly challenging since there are numerous categories of objects in the world. This is an exhaustive task as the relevant data should be gathered and annotated accordingly. Moreover, increasing the number of output classes for a deep network usually leads to lower per-class performance. On the other hand, humans can grasp almost any object by a simple glance at them without respect to their familiarity to the object. Considering the output of a hand prosthesis as a grasp act, a deep network can be designed to predict a grip pattern to be fed into the hand controller. Thus, the object recognition task can efficiently be simplified to a grasp recognition one.

3.1.2 Grasp Recognition with CNNs

In order to translate the advancements of deep learning in object recognition to perform the grasp recognition task for better control of hand prostheses, we took advantage of the flexibility of deep networks in automatically learning patterns. That is, during the grasp identification task, objects are categorised based on their appropriate grasp type learned from their abstract representation rather than their object category or accurate dimension measurements. It should be noted that throughout this thesis the concepts of grasp recognition, grasp identification and grasp classification are used interchangeably.

To classify objects based on their suitable grip patterns, they should be first manually categorised for this task. There is usually a preferable grasp manner for every kind of object, as an example, to grasp a tiny pebble, people always utilise their thumb and forefinger called a *pincer* or *pinch* grip. In this chapter, four common and sufficiently distinct grasp patterns are selected: *pinch*, *tripod*, *palmar wrist neutral* and *palmar wrist pronated*. These grips can effectively be applied to any graspable object. That is, since each grasp class is suited to specific object features, *e. g. pinch* for granular items or *palmar wrist pronated* for large spherical or cubical items, using this subset of grasp categories, most available objects can be apprehended in a stable manner. Besides, grasp classes are proposed based on the object appearance and a CNN model requires adequate object images representing each class. The opted grasp categories seemed to be a good fit to available datasets of graspable objects.

In this way, instead of having a huge number of object categories, all the available objects are classified into 4 aforementioned grasp classes. Consequently, when the network encounters a novel item, it can always assign it to one of the present grip classes and therefore generalises to unseen shapes. To further illustrate this generalisation and as a comparison with object

recognition task, a tiny pebble belongs to “pebble” category, while a pen lid belongs to “pen lid” group. Performing grasp recognition however assigns both objects to *pinch* class. Moreover, introducing a novel item to the network, such as a dice, which was never seen by the network during training, leads to erroneous output in object recognition as one of the present classes are picked as the prediction. Contrarily, undertaking grasp identification provides the correct output *pinch* again. The distinction between object and grasp recognition tasks are highlighted in figure 3.1.

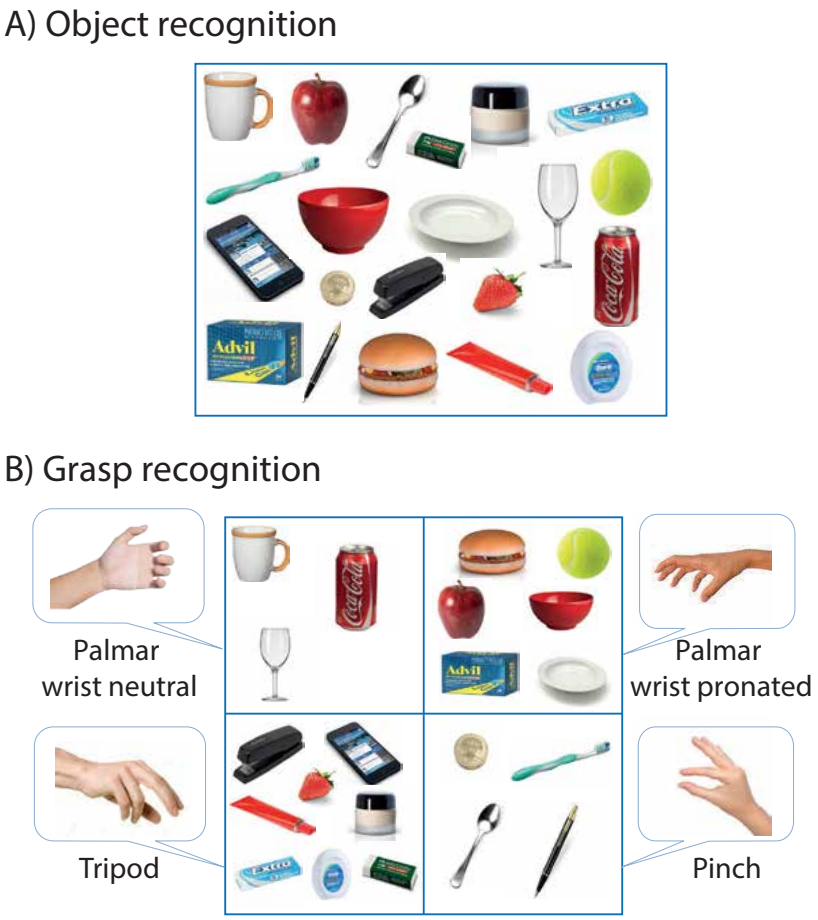


Figure 3.1: Object versus grasp recognition

The design and flexibility of deep networks perfectly suits to the solution of grasp classification. That is, the weights are learned automatically during training based on the given outputs. Hence, the deep network can learn through training to extract grasp-relevant features and construct a high level abstract representation based on those features such that no hand engineering, shape considerations or measurements are needed. Hence, this approach is conceptually different from object recognition as object details are of less importance while the general appearance such as size and orientation demand more attention from the network.

As highlighted in previous chapters, CNNs [12] indicated exceptional performance in recognition tasks (in some applications better than humans) [15, 112]. Therefore, a CNN structure

can be designed to learn this abstract representation effectively.

3.2 Image Datasets

To train a deep network, typically large amount of data is required. To feed adequate information to a specific CNN structure, Columbia Object Image Library (COIL100) [18] was initially opted. The COIL100 dataset contains images of household objects provided in full 360° view categorised in 100 classes of objects.

Later, Amsterdam Library of Object Images (ALOI) [140] was used as a more comprehensive dataset with more variety of objects. The ALOI dataset was captured in a very similar way as the COIL100 (360° view of objects were provided). It includes 110, 250 images of 1000 object categories.

The black background of the images in both datasets aids the recognition procedure such that the network merely concentrates on the object characteristics during learning. However, this feature can be counted as a demerit while testing in a real-world scenario as clutter is normally present in such cases. Nonetheless, the implementation is a proof-of-principle for usage of deep learning for grasp classification in hand prosthetics and the background issue can be easily resolved by fine-tuning the network on cluttered scenes.

To provide comparable testing setup in a real-world situation, the ALOI dataset was augmented by the dataset gathered at Newcastle university called Newcastle Grasp Library. In this way, the algorithm can be evaluated practically on both seen and unseen objects. Both image datasets are explained in the following.

3.2.1 Columbia Object Image Library (COIL100)

The COIL100 dataset [18] consists of 7200 images including 100 categories of objects. There are 72 different poses per object, which are obtained by taking photos of each item against a black background at each 5° rotation through a turntable that covers 360°. The objects have a wide variety of complex geometric and reflectance characteristics. The images are size normalised and the images with the dimension of 32×32 have been used for experiments with the COIL100 (only offline experiments involved the COIL100 dataset).

3.2.2 Amsterdam Library of Object Images (ALOI)

The ALOI dataset [140] consists of images of 1000 common objects, from which 250 objects are captured at a second zoom rate. Therefore, these 250 objects were eliminated for more consistency with respect to object sizes. For every object in the ALOI, there are 72 images taken at 5° intervals covering a full 360° view. All images have black background and 768×576 pixels resolution. They are taken at 124.5cm distance and 30cm altitude from the objects. Not all the 750 remaining objects were used for implementation as some objects were not graspable or specific to a single grip pattern. Hence, a selection of 473 objects were *subjectively* picked and categorised in four grasp classes of *pinch*, *tripod*, *palmar wrist neutral* and *palmar wrist pronated*. As a preparation step, all the images were converted to grey-scale and then downsampled to 36×48 pixels resolution. Figure 3.2 A illustrated a sample of objects present in the ALOI dataset.

3.2.3 Newcastle Grasp Library (NCL)

This dataset was created to provide accessibility to the test objects. That is, when doing real-time experiments real objects are needed and the same objects as the ones in the ALOI dataset were not available. Therefore, 71 objects were photographed in a similar setup as the ALOI to provide comparable images that can be used for online experiments. To this end, a Crayfish 55 turntable (Seabass, UK) and a Canon Kiss X4 DSLR camera (resolution 18 Megapixel, 3456×5184 pixels) were arranged to synchronously capture images of each object at 5° intervals against a black background. The camera and turntable positions were fixed throughout the photography to take the object sizes into account. Hence, the camera was placed in 60cm distance and 15cm altitude from the objects. This setting led to object images with comparable sizes as the ALOI ones. As a final step to equate the data with the processed ALOI data, the images were converted to grey-scale and downsampled to a resolution of 36×48 pixels. These images are then ready to be given to a CNN for training. Figure 3.2 B demonstrates all the objects belonging to Newcastle Grasp Library.

It is worth noting that the 71 additional objects in Newcastle Grasp Library were also picked from the four grasp groups used previously. The objects are elected such that the balance in the total number of items in each grip class is preserved and the class sizes are comparable. Table 3.1 illustrates the number of objects corresponding to each grip mode.

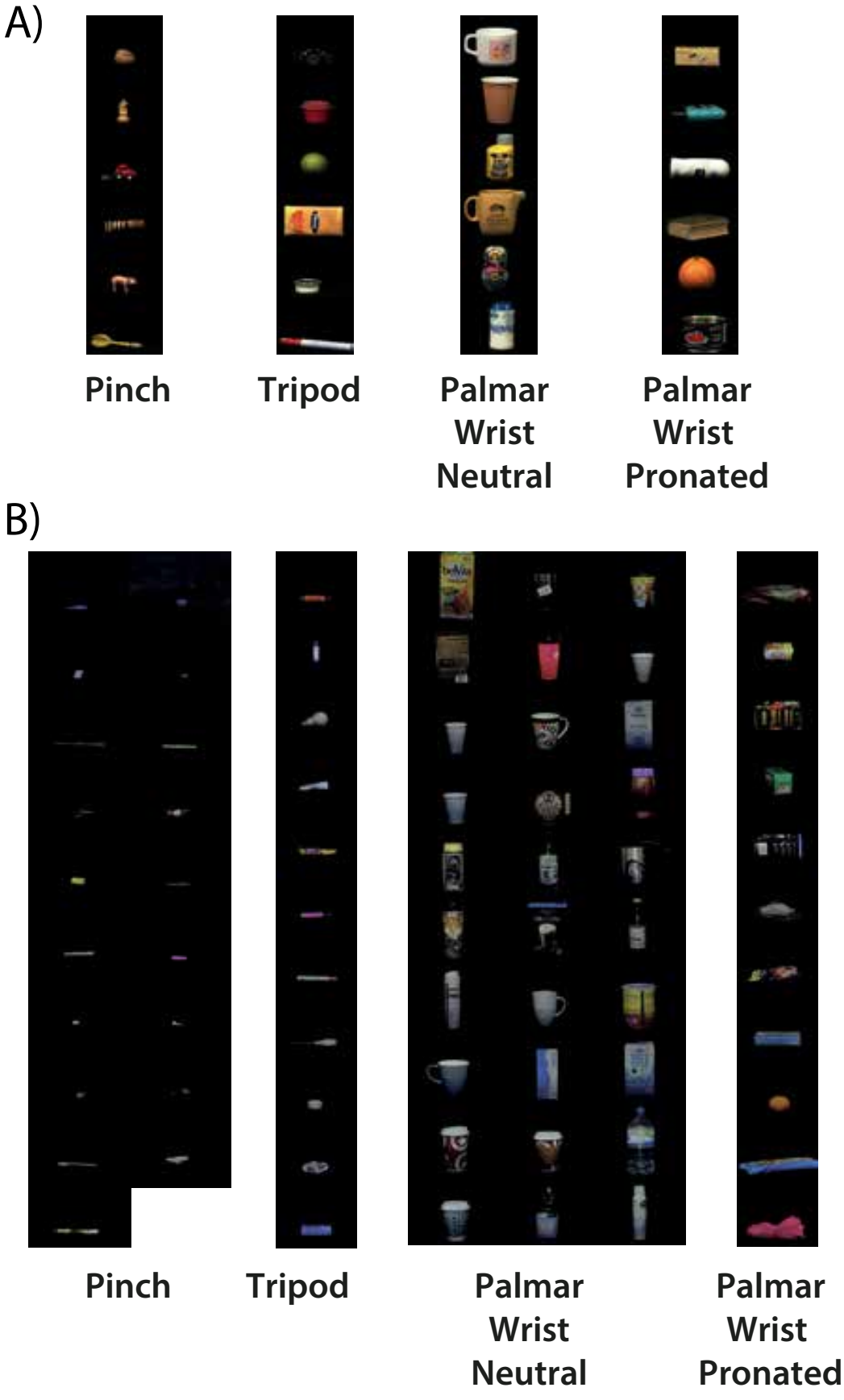


Figure 3.2: The objects used in this work separated based on their preferred grip pattern A) A subset of the objects used from the ALOI dataset; B) All the objects used from the Newcastle Grasp Library.

Table 3.1: The number of objects per grasp group in the ALOI and Newcastle Grasp datasets.

Grasp type \ Dataset	ALOI	NCL
Pinch	90	19
Tripod	163	11
Palmar Wrist Neutral	83	30
Palmar Wrist Pronated	137	11
Overall	473	71

3.3 Grasp Recognition with Convolutional Neural Networks - Initial Experiments

In this section, the initial steps taken to implement a CNN-based vision system to estimate an appropriate grip pattern for objects in the COIL100 dataset are explained in detail. These experiments are carried out on a comparatively small dataset to investigate the possibility of having a vision-based grasp recognition solution as well as a variety of solutions for grasp classification of unseen objects. As such, several offline settings including different architectures, parameters and components of CNN models are examined and reported.

3.3.1 CNN Architectures for COIL100 Dataset

Initial offline experiments were carried out on the COIL100 dataset [18] to investigate the feasibility of grasp classification for household objects. Figure 3.3 indicates the COIL100 dataset categorised based on grip patterns. As it is noticeable, the classes are highly imbalanced.

It is worth noting that for the COIL100 dataset a variety of architectures were examined and the ones providing the best performance were reported, which are different from the architecture presented in the following sections and used for experiments involving the ALOI dataset. Figures 3.4 and 3.5 demonstrate the architectures that were used with the COIL100 dataset.

Each input image I was converted to grey-scale and resized to an image with $N = 32$ rows by $M = 32$ columns. Then, Gaussian and median filtering were applied to the image respectively for noise removal and smoothing. Finally, Z-score normalisation discussed in Equation A.6 was utilised for image distribution balancing and further training enhancement.

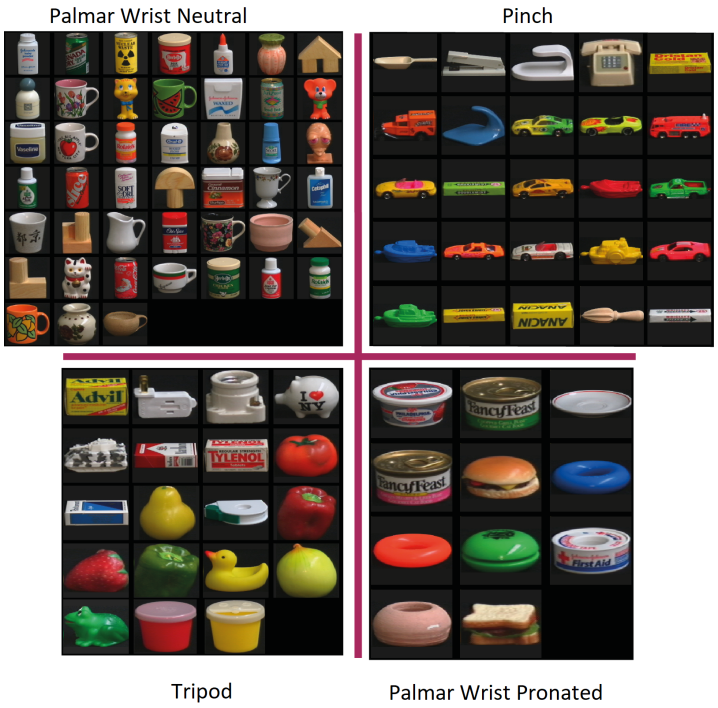


Figure 3.3: The COIL100 [18] dataset categorised based on four grip patterns.

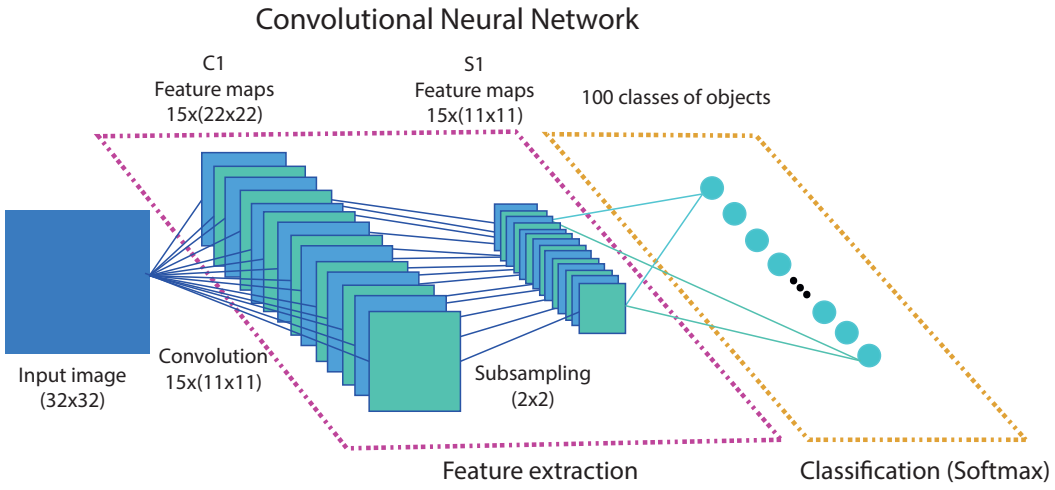


Figure 3.4: The one-layer CNN including 15 filters implemented for object classification with the COIL100 dataset.

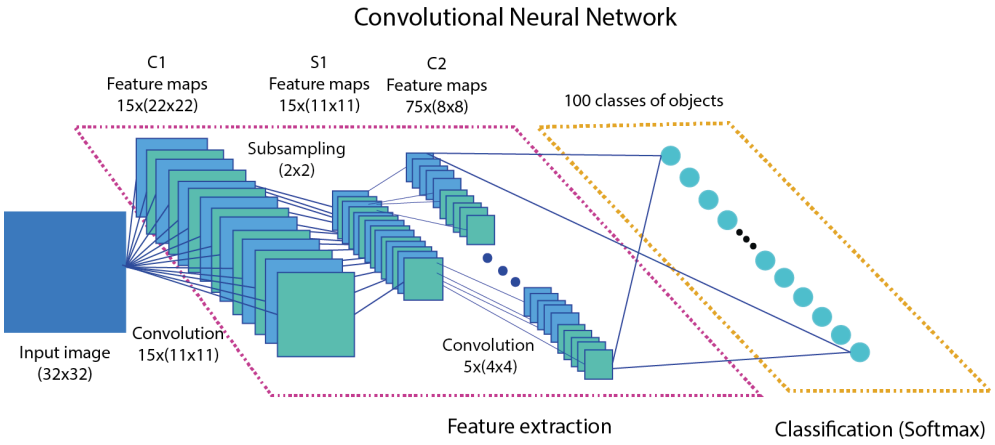


Figure 3.5: The two-layer CNN used for object recognition with the COIL100 dataset.

3.3.2 Object Recognition

In order to have a baseline for the experiments, objects were firstly classified based on their corresponding object category. That is, different views of each object were given to one- and two-layer CNN structures adapted for object recognition as depicted in Figures 3.4 and 3.5. To evaluate the classification performance of the proposed structures on the COIL dataset, each model was trained on 90% of views for each object category among the 100 categories and tested on the remained 10% of the views.

3.3.3 Grasp Classification

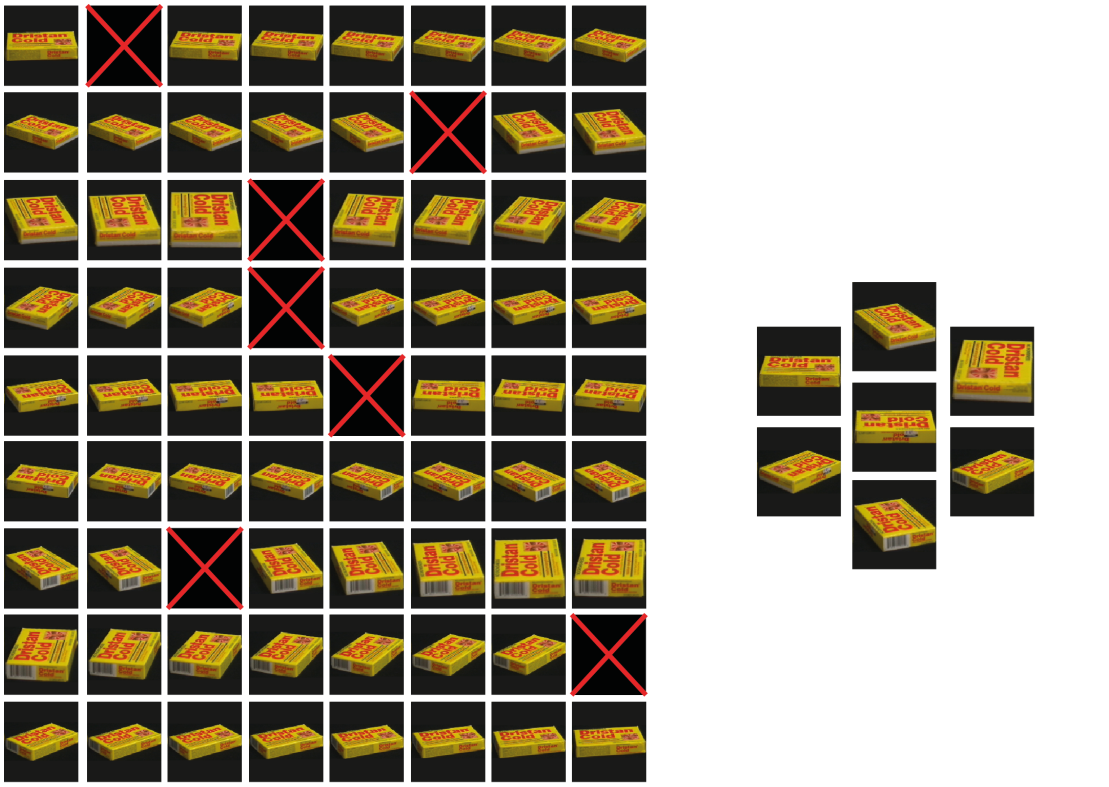
Grasp identification requires the models illustrated in Figures 3.4 and 3.5 to be modified slightly. To illustrate, the last layer of the CNN architectures, the Softmax layer, which is responsible for the classification task is adapted such that there are four grasp categories in the output.

The grasp classification performance of the models are evaluated with respect to view or object category novelty through within- and between- object cross-validations introduced in Section B and further described in the following.

Within-Object Cross-validation

In this cross-validation, the training set includes 90% of the views for each object in each grasp class (65 out of 72 images for each object). The remaining 10% of the views for each object were specified to the test set. Results are validated in 10 folds and each fold was repeated with 10 different random weight initialisations to investigate the effect of weight initialisation on the results. Hence, the algorithm was trained and tested 100 times for WOC in total. In this

way, the variability of results based on weight initialisation and view selection can be achieved. Figure 3.7 represents randomly selecting 10% of the views for each object as the test set, as the remaining 90% of views belong to the training set. The views in the test set are considered as unknown to the algorithm, so it is rational to expect lower classification performance for objects with distinct appearance in different poses.



WOC Train set
90% of 72 different poses = 65 poses

WOC Test set
10% of 72 different poses = 7 poses

Figure 3.6: Random view selection of objects in COIL100 for test and train splits in the WOC setting .

Between-Object Cross-validation

In this validation setting, the training set includes 90% of all the object categories in each grasp group with all of their different views (*e. g.* 17 objects of *tripod* class with their 72 different poses are selected for training). The remaining 10% of the object categories are given to the test set (*e. g.* 2 objects in *tripod* class belong to the test set). Therefore, every object in the test set is considered as unknown to the algorithm. The same kind of cross-validation with regard to weight initialisation as WOC setting was carried out resulting in 10 folds with 10 various weight initilisations. Figure 3.7 illustrates the process of randomly selecting 10% of object classes for test set in the BOC setting.

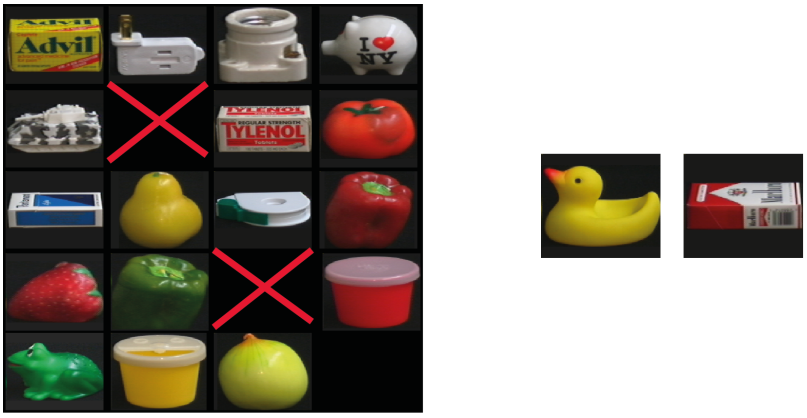


Figure 3.7: Random object selection of test and train splits for BOC setting.

Stability Analysis with Respect to Weight Initialisation and Object/View Selection

During each test, 10-fold cross-validation was carried out twice, one across random weight initialisation and one across random test object/view selection. The algorithm was run 100 times; firstly a random set of objects or views was selected for the test setting in the BOC or WOC respectively, then the initial weights were configured randomly. In the next run the weight initialisation was changed, while keeping the previous test objects/views. After trying 10 different set of weights for a group of test objects/views, a new set of objects/views for testing and accordingly training was selected. This procedure was followed for 10 different set of random objects/views.

This cross-validation approach provides more reliable set of results, as the performance changes dramatically by testing different test objects/views. Weight initialisation also seem to have a considerable effect on the results.

Chance Level

Chance level is the accuracy expected by random choices. Specifically, here the chance level is the probability of an object belonging to a class by arbitrarily selecting an object. In the object classification case, where there are 100 classes, the chance level would be 1%, while for grasp classification the chance level should be 25% as there are four grasp classes. However, since the number of objects in each group is different, the probability of a random object belonging to each class would be dependent on the number of test objects in that class. Hence, in the WOC test, for *pinch*, *tripod*, *palmar wrist pronated* and *palmar wrist neutral* the chance level would be 25%, 19%, 11% and 45% respectively. For BOC setting, the chance level values are different,

because the number of test objects in each grasp group is modified as well. Accordingly, there are 2, 2, 1 and 4 test objects in *pinch*, *tripod*, *palmar wrist pronated* and *palmar wrist neutral* groups respectively leading to the sum of 9 objects and the chance level of about 22%, 22%, 11% and 44% correspondingly. Table 3.2 shows the chance level for each test.

Table 3.2: Average grasp recognition chance level of the system on COIL100 dataset for four grasp categories.

Grasp type	Chance level (WOC)	Chance level (BOC)
Pinch	25%	22%
Tripod	19%	22%
Palmar Wrist Neutral	45%	44%
Palmar Wrist Pronated	11%	11%

3.3.4 Results and Discussion

The results explained in this section are only reported for the COIL100 dataset and the architectures developed and evaluated on it. As this set of results are preliminary offline results, which their analysis provided more insight for system improvement, the results and discussion are presented jointly to provide better reasoning for the next section. The final designed structure is presented in the next section.

Table 3.3 represents the results achieved with different architectures in three different settings: object recognition, WOC and BOC. Two different CNN architectures presented in Section 3.3.1 were examined using different activation functions ($tanh$ and ReLU), pooling techniques (max, average and stochastic pooling) and local response normalisation. This examination provides the opportunity to recognise the best settings for future implementations.

Object Recognition

According to Table 3.3 1 the average accuracy achieved for object recognition was above $99.06 \pm 0.0712\%$ and $97.65 \pm 0.322\%$ for one- and two-layer CNN structures respectively. There is a drop in average accuracy for two-layer CNN compared to one-layer CNN. Several reasons can contribute to the better performance of the shallower CNN. One viable explanation can be the simplicity of the task and train images, which refuses the requirement of extra abstraction level that the two-layer CNN provides. This is worth mentioning that although good results were achieved for the object identification task, it is not possible to generalise a network

Table 3.3: The results for different architectures, where the best performance is in bold. All architectures are CNN based, while XYZWH stands for:
X: Number of convolution layers
Y: Number of pooling layers
Z: Activation function (T:tanh, S:sigmoid, R:ReLU)
W: Pooling method (M:max pooling, A:average pooling, S:stochastic pooling)
H: Using (or not using) local response normalisation, where using "L" indicates using local response normalisation and "N" represents not using it.

1) Average accuracy performance of different CNN models on COIL100 dataset for object recognition.

Architecture	Average accuracy (%)
11TAN	83.1
11RMN	**99.062**
22TAN	55.23
22RMN	97.65
21RMN	99.06
22RML	95.24

2) Average accuracy (%) performance of different CNN models on COIL100 dataset for each grasp in WOC setting.

Architecture	Pinch	Palmar Wrist Pronated	Tripod	Palmar Wrist Neutral
11TAN	79.43	81.54	76.15	75.07
11RMN	92.06	99.42	91.8	95.3
22TAN	75.43	70.52	69.36	67.39
22RMN	90.54	93.88	89.21	90.05
21RMN	**96.84**	**97.9**	**96.6**	**97.9**
22RML	90.99	98.18	89.76	94.03

3) Average accuracy (%) performance of different CNN models on COIL100 dataset for each grasp in BOC test.

Architecture	Pinch	Palmar Wrist Pronated	Tripod	Palmar Wrist Neutral
11TAN	71.22	21.79	48.28	47.01
11RMN	89.38	18.78	53.04	54.99
22TAN	61.59	46.25	51.59	51.7
22RMN	80.00	28.00	57.13	45.84
21RMN	**82.72**	**32.66**	**54.45**	**53.17**
22RML	87.11	16.87	53.99	54.89

trained on object classification to unknown categories of objects, which is the main concern of this work.

Grasp Classification

Object recognition provides a measure for the simplicity of the dataset and a baseline for comparing the grasp classification settings with it. The following sections provide further reasoning for the achieved results for grasp recognition task.

Within-Object Cross-validation

The performance of the one-layer CNN in the WOC setup reaches 94% grasp recognition average accuracy. To be more specific, there is above 91.8% average accuracy ($\pm 1.32 - 4.09\%$ across view selection and $\pm 0.13 - 0.44\%$ across weight initialisation) for each grasp type.

The best accuracy was achieved for *Palmar wrist pronated* grasp. This observation could be due to the less variety of objects in this group and the fact that most of the objects belonging to this group are symmetric in shape. Therefore, different views do not represent distinct shapes, which can be unfamiliar to the algorithm. On the other hand, the *palmar wrist neutral* grasp is the second best group in average accuracy. A main reason can be the high number of training examples in this group. This high number of samples aided the algorithm to have a better chance of learning, even when considering the high variety of objects in that grasp group. As stated, data is a key factor in training the deep networks.

Finally, a comparable performance on objects of *tripod* and *pinch* groups is observed in the WOC setting. The main reason could be the huge difference in the appearance of objects from different view points; *e. g.* a toy car seems rectangular from 0° view, while it seems as a squared shape from 90° angle. Having more training data for these two groups could be helpful in improving the average accuracy.

The best average accuracy for the two-layer CNN is about 97.32% and it is above 96.6% ($\pm 0.8 - 2.76\%$ across view selection and $\pm 0.17 - 0.4\%$ across weight initialisation) for each grasp.

Two-layer CNN outperformed the one-layer CNN and reduced the difference between grasp groups while following the same order in performance for each group. This could be due to the fact that the increase in the depth of a network can contribute to learning more parameters. Having higher level of abstraction and neglecting the details could provide a better invariance to the variety in poses, leading to better performance of the two-layer architecture. There is

always a trade-off between the sufficient depth of the network, the model complexity and the network's generalisation capability.

Setting the filter dimension of the second pooling layer to the same size as the that of previous layer (2) degrades the performance to $\sim 90\%$. This observation indicates that this pooling layer caused elimination of helpful information during sub-sampling. Consequently, no pooling layer was added to the two-layer CNN.

It can be noticed that CNN is more robust to weight initialisation than view selection, as the standard deviation is a higher value across view selection than weight initialisation in all the three architectures. This seems plausible, since pose selection can lead to testing a view (pose), which is barely familiar to the trained algorithm and makes the recognition task more challenging.

Between-Object Cross-validation

The average accuracy gained by one-layer CNN for BOC test is 54.05% in general, while the two-layer CNN provided 55.75% average accuracy. There is a huge difference between the algorithm performance in the WOC and BOC validation settings, which clearly illustrates the challenging task of grasp recognition for unknown objects.

The average accuracy for each grasp type is above chance level for all the grip modes. However, the second grasp type *palmar wrist pronated* seems problematical to be recognised for the algorithm. Looking into the number of objects in that group provides a rational explanation for this unacceptable performance. That is to say, due to having only 10 training objects for the *palmar wrist pronated* grasp, the algorithm could not learn the grasp type suitable for each object. There is also only one test object in order to evaluate the performance of the algorithm. Not being able to recognise a test object, there is a high probability that all the poses for that object are not identified. That is why, such poor results are observed for that group. It can be assumed that proliferating the number of objects in this category, or generally in all the categories, could cause an acceptable boost in the achieved results.

The best performance in the BOC setting was obtained for *pinch* grasp type. Contrary to the WOC setting, all the objects poses were learned during training and therefore no difficulty in grasp recognition of different views was encountered. Thus, as the object shapes in *pinch* grasp group are chiefly similar to each other, a good learning performance is obtained.

On the other hand, *tripod* and *palmar wrist neutral* grasp groups are indicating similar average recognition rates. These results could be due to the high variety of objects in both

groups that causes slower learning, even with many objects for training in *palmar wrist neutral* group. The initial solution could be increasing the number of objects in both groups. It is worth noting that the deep networks are following a similar procedure of learning as humans. That is, seeing a wider variety of objects by humans leads to more convenient recognition of novel objects. Similarly, as the task in BOC setting is grasp recognition for unknown objects, the more objects the algorithm is trained for, the better it generalises for novel objects. Therefore, it can be expected that even if there is a very high diversity of objects in one grasp group, training the algorithm for adequate number of samples can bring about satisfactory estimation capability of the algorithm.

3.3.5 Conclusion

In this section a variety of network architectures with different pooling structures, normalisation techniques and activation functions were examined to comprehensively evaluate the performance of the grasp recognition solution for hand prosthesis. For all the three settings of object recognition and grasp classification in WOC and BOC settings, the best performance was gained with the CNN with two convolution and one max pooling layers including ReLU activation function and no local response normalisation.

Promising results for the WOC validation setting were achieved. Although indicating promising classification performance for some groups, the BOC setting included unreliable results for the grasp classes with few amount of data. The huge imbalance in the available training data for different classes is the possible reason for this unacceptable performance. Hence, with provision of a better dataset including more variety and abundance of objects and a balanced distribution of images among grasp classes better results can be achieved.

3.4 Grasp Recognition with Convolutional Neural Networks - Comprehensive Experiments

The tests performed on the COIL100 dataset were mostly trial and error steps to come up with the right dataset and architecture. After performing extensive analysis and figuring out the shortcomings and capabilities of the architecture, dataset and the task at hand, more comprehensive analysis was performed on the augmented ALOI dataset (ALOI dataset with additional images from Newcastle grasp library), which includes significantly larger number of object images than the COIL100.

The boost in the amount of available data together with better familiarity with the network capabilities, components and parameter tuning techniques can lead to a more robust and reliable grasp classification platform. Therefore, not only this section includes a more comprehensive dataset, but also the architecture designs are mostly defined based on the experiments of the previous section. Additionally, the structure proposed in this section was analysed in three ways: offline, online computer-based and online with an amputee user in the loop.

3.4.1 Feature Extraction with CNNs

Each input image I was initially converted to grey-scale and resized to an image with $N = 36$ rows by $M = 48$ columns. Gaussian and median filtering were then respectively applied to the image for noise removal and smoothing. For better training and balancing the distribution, processed images were normalised using Z-score normalisation as explained in Equation A.6.

Firstly, the simplest CNN architecture was used, a CNN with one convolution (C_1) and pooling layer (S_1), called one-layer CNN here. To observe the effect of depth , achievable accuracy and computational complexity as well as the amount of abstraction and generalisability required and to provide a balance between these parameters, convolution layers were incremented continuously. These additions however were effective only when having two convolution (C_1 and C_2) and one pooling S_2 layers at the end, called two-layer CNN here. Therefore, the experiments are all based on these two architectures, one- and two-layer CNNs. Figure 3.8 manifests the two-layer CNN structure used in this work including the details of kernels, feature maps and their corresponding dimensions.

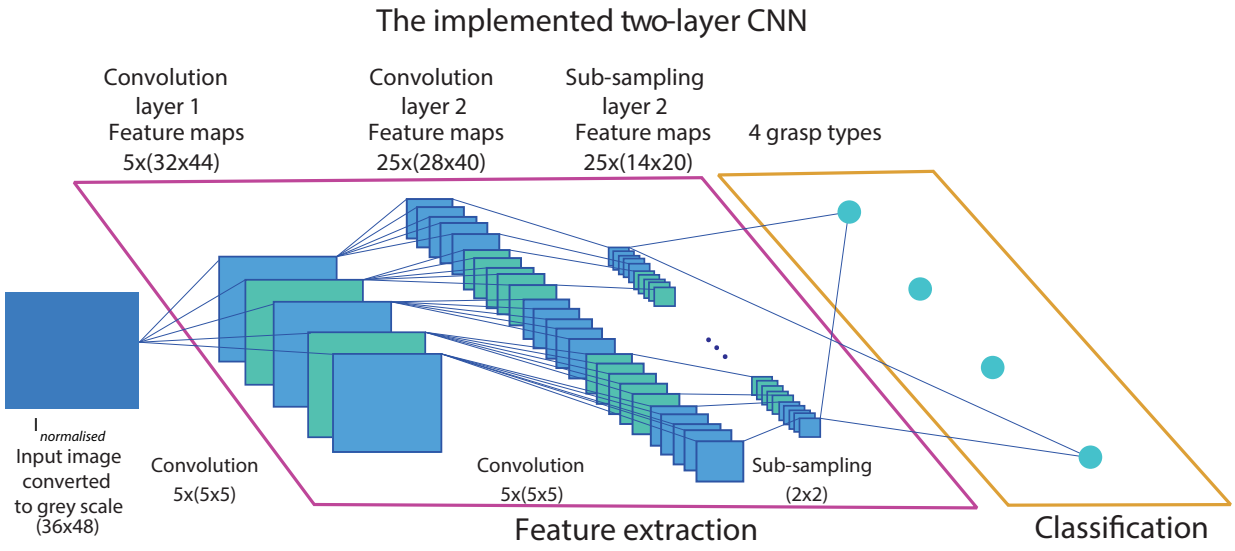


Figure 3.8: The implemented two-layer CNN architecture applied to the augmented ALOI dataset.

Normally deep networks have more number of layers providing them with more depth and amount of abstraction. Although a variety of deeper architectures having different number of layers and filters were implemented, the results were not improved through the increase in the amount of depth or filter sizes. The possible reasoning behind this matter is discussed thoroughly in the discussion Section 3.7.3. The networks that provided the best grasp classification performance are reported in this work. In the following more details over the developed CNN structure are described.

A variety of activation functions such as sigmoid, hyperbolic tangent and rectified linear unit were examined. However, due to the privileges of ReLU and also our empirical observations, it was opted as the activation function for implementation in all the CNNs.

In both one- and two-layer CNNs, five filters ($\mathbf{K}_j^1$, $j = 1, \cdots, 5$) of size 5×5 were utilised and max pooling [141] was applied to the convolved feature maps for sub-sampling the output by a factor of two. Max pooling was preferred over other pooling methods as it guarantees that activation with high values which are usually important components within each feature map are passed to the next layer.

3.4.2 Classifier - Softmax Regression

A Softmax layer explained in Section A.1.7 was added at the end of the feature extraction steps to ease end-to-end training of the whole network and maximise the amount of automation in learning.

3.4.3 Training

During training, the network weights were updated based on the mini-batch momentum gradient descent algorithm (Section A.1.8) in an end-to-end manner. A Tikhonov regularisation term (Section A.1.9) was also added to the cost function for preventing over-fitting by enforcing sparsity on the weights $\mathbf{K}_j^l$.

3.4.4 Cross-validation

As described in Section B two kinds of cross-validation are performed in this thesis: Within- and Between-Object cross-validations. More details of these validations specific to the augmented ALOI dataset are illustrated in the following. These cross-validations are performed on both CNN settings (one- or two-layers).

Within-Object Cross-validation

For this validation, 90% (65 of 72) of the available views for each object are specified for training, while the remained 10% of the views per object were used for evaluation. These views are chosen randomly for 10 different folds so that network's sensitivity to the choice of view is qualified properly. Figure 3.9 presents WOC train and test splits for a sample object in Newcastle Grasp Library.

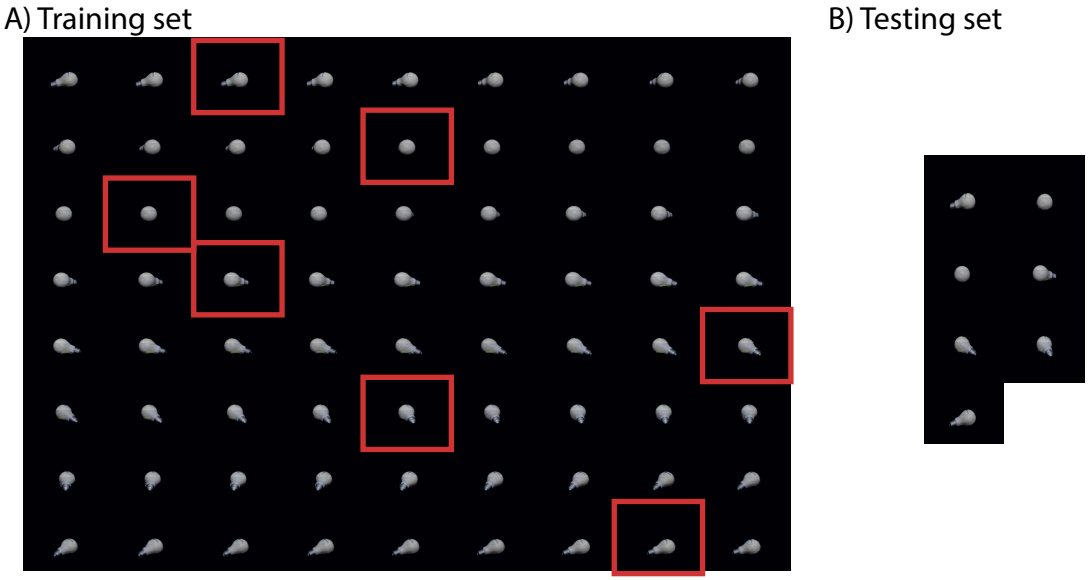


Figure 3.9: Within-Object Cross-validation. Train (A) and test (B) splits of an example fold in 10 folds are shown for a light bulb present in Newcastle Grasp Library. The test views are randomly picked and shown in red boxes.

Between-Object Cross-validation

For the ALOI dataset, $\sim$90% of all the items in each grasp group with all of their 72 views were included in train set. The remaining $\sim$10% of the items, including all their poses, were placed in the test set. As an illustration, 124 out of 137 objects in *palmar wrist pronated* class were included in training set, while the rest of objects, 13 out of 137, were used for testing.

As the objects present in Newcastle dataset are less abundant than the ALOI and they were required for real-time testing, a different portion of objects of Newcastle dataset were picked for test split. That is, 4 objects were randomly picked in each grasp group and allocated to the test set with all their poses. The remained objects in each grasp group were used for training. This selection is clearly illustrated for a sample object in Figure 3.10. The object categories are always selected randomly and the procedure was repeated 10 times independently.

Further details about the exact number of objects of each grasp group and each dataset

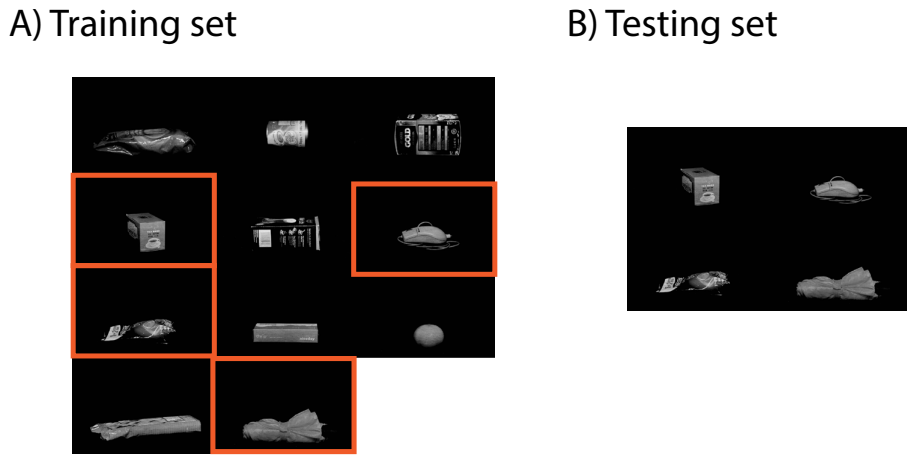


Figure 3.10: Between-Object Cross-validation. The test and train splits for the objects belonging to the *palmar wrist pronated* class in Newcastle Grasp Library are shown. This grip group consists of 11 objects and 4 of them are picked for test randomly. The 7 remained objects and all their views are given to the train set.

selected for test and train sets of the BOC is reported in Table 3.4.

Table 3.4: The BOC evaluation details for both ALOI and Newcastle datasets.

Grasp type \ Dataset	ALOI		Newcastle	
	Train	Test	Train	Test
Pinch	81	9	15	4
Tripod	147	16	7	4
Palmar wrist neutral	75	8	26	4
Palmar wrist pronated	124	13	7	4

3.4.5 Statistical Analysis

To evaluate the results thoroughly, statistical analysis of the results was carried out using a two-way repeated-measures ANOVA. The goal was to measure the impact of cross-validation type (BOC versus WOC) and number of layers (1 versus 2) in offline grasp classification performance. To perform this investigation, each fold among the 10 cross-validation folds was considered an independent sample.

3.4.6 Computer-based Real-time Performance Analysis

The deep learning-based vision system is implemented in real-time using the learned parameters of the designed CNN in BOC setting. As explained in Section B.1.2, this validation setting eval-

uates the capability of proposed networks in grasp recognition of unseen objects and therefore is aimed at real-life situations, in which objects are usually novel to the user.

This computer-based experiment is designed to evaluate the performance of the designated model in a real-world setting independent of user's behaviour. To do so, an inexpensive web camera (Logitech Quickcam Chat) was placed on a photography tripod stand such that object is situated at 60cm distance and 15cm height from it. This setting was intended to resemble the photography setting of Newcastle dataset. The camera could be activated by a laptop, which was connected to it via USB connection. In this experiment the camera resolution was set to 640×480 pixels.

The experiment was initiated with a click on command button on a MATLAB®-based graphical user interface (GUI), which activates the camera capturing. The captured image was first processed passing through object detection, background removal, resizing and normalisation steps and then introduced to the two-layer CNN trained in a BOC setting. The preprocessing steps are illustrated in Figure 3.11. This procedure was carried out for 6 different objects in each grasp category (24 objects in total) for 7 random views of each object. In this test, 16 out of 24 objects (66%) were novel to the trained model.

The main parameters discarded here by having a computer-based real-time experiment are distance between camera and object, user's motivation, EMG data quality and acquisition performance and physical fatigue. Neglecting these parameters as a first step can provide the overall system with a recognition potential comparable with offline experiment results.

3.4.7 Real-time Test Platform with Amputee Users in the Loop

The final experiment involves an amputee user in a similar setup as that of previous test in Section 3.4.6 to evaluate the system in a real situation considering all the contributing parameters. Contrary to previous experiments, in this test the camera was attached to a hand prosthetic.

The study was approved by the Newcastle University ethics committee and the participants signed a consent form to attend this experiment. In the following, different components of the whole setting are explained.

Subjects

Two amputee volunteers attended in the real-time experiment. The volunteers used split hook prostheses for their daily lives and therefore had very little experience, only few experiments in laboratory, with myoelectric hands. More details about the volunteers can be found in Table 3.5.

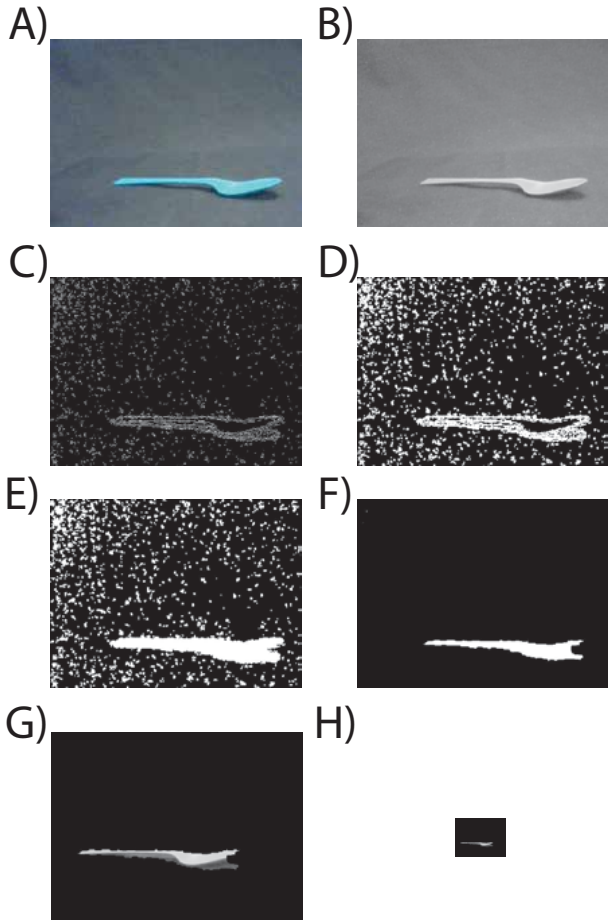


Figure 3.11: Image preprocessing steps for real-time experiments: A) Original image, taken by the webcam, B) Grey-scale transformation, C) Sobel edge detection, D) Dilation, E) Filling the closed spaces, F) Erosion and filtering the extra noises, G) Multiplication of the mask calculated in F to the original image in A to detect the object and translation to the lower centre of the image, H) Downsampling to 36×48 pixels.

Overall Control Structure and System Components

The general flow diagram of designed real-time experiments is depicted in Figure 3.12A. The implemented programme is illustrated in Figure 3.12B. The test procedure composed of 6 blocks, from which the last block benefited from error correction feature. The error correction routine including its required operations and relevant connections to other tasks is specifically highlighted in a grey box in Figure 3.12B. For blocks 1-5 the grey box and its relevant connections can be ignored. Other than the error correction attribute of block 6, same control flow is followed in all the blocks.

Table 3.5: Experiment subjects' information

Identifier	Gender	Age	Cause of amputation	Years since amp.	Missing limb	Prosthesis use
M	Male	28	Car accident	6	Right	Split hook
D	Male	54	Cancer (Epithelioid Sarcoma)	18	Right	Split hook

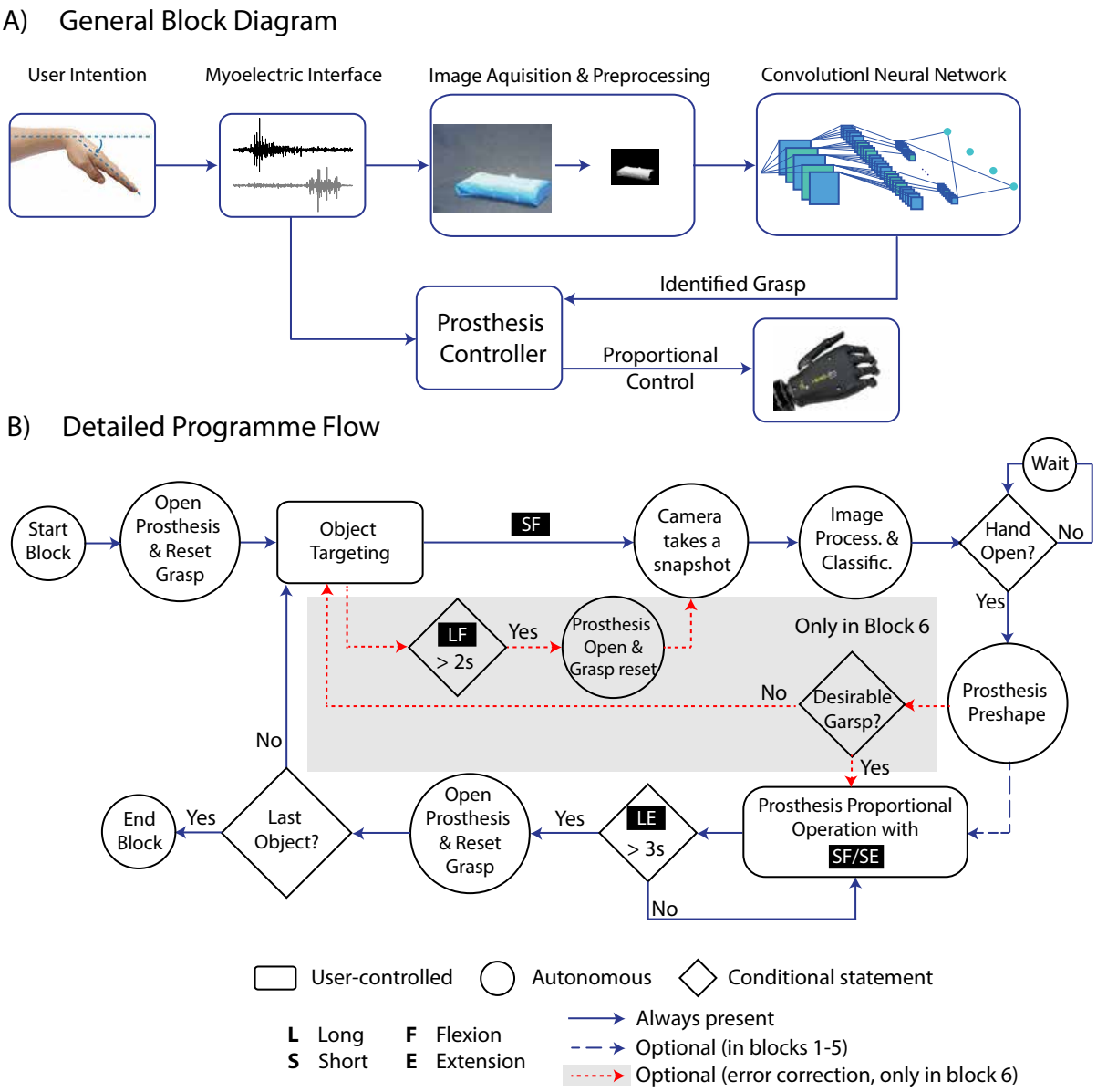


Figure 3.12: Overall control structure. A) The implemented real-time structure illustrated in block diagram; B) Detailed programme flow that was operated via a standard two-channel myoelectric interface.

The general grasping procedure is explained in the following. The user can trigger the webcam and capture a snapshot of a target object by producing a short (300ms) flexion of wrist muscles. The snapshot is then processed and given to the two-layer CNN trained in BOC setting. The CNN suggests a grasp, which causes a preshape in the prosthetic hand. The subject can control the prosthesis proportionally as the EMG signals from the wrist flexor and extensor muscle groups are recorded for prosthesis control. Long (3s) extension resets the grasp and opens the prosthesis at any time.

Thanks to the error correction feature in block 6, the user can benefit from more supervision by discarding a suggested grasp and re-aiming the hand at the target object. A long (2s) flexion

of the wrist muscle re-opens the prosthesis, resets the grasps and captures a new snapshot. The error correction procedure can be repeated until the desired grasp type is achieved.

The Measurement and Preprocessing of EMG Signals

After skin preparation, two Delsys® Trigno™ Lab Wireless EMG electrodes were located on the wrist flexor and extensor muscle groups on the forearm to record the relevant muscle activities. A band-pass filter between 20Hz and 450Hz was applied to surface EMG signals. The filtered signals were then sampled at 2kHz via a Trigno Digital SDK.

The output EMG signals were then converted to analogue control signals bounded between [0,1] such that 0 indicates muscles being at rest and 1 represents contraction at a comfortable level. The latter is normally 10-15% of the maximum voluntary contraction by the user, which can cause sensitivity due to the low amplitude. This issue however was shown not to be problematic in [142–145], since subjects can effectively learn to contract their muscles for reliable computer task or prosthesis control performance. A possible explanation can be that at this low percentages of maximum voluntary contraction (MVC), the magnitude of the signal-dependent motor noise is negligible [146].

A rectangular window was then applied to a 500ms of each rectified EMG channel for smoothing and computation of a control signal c in every 100ms interval according to the following equation

$$c_k = \alpha_k \sum_{500ms}^{\delta=0} |EMG_k(t-\delta)| \tag{3.1}$$

where $|EMG_k(t)|$ denotes the rectified activity of muscle k at time t. The α_k coefficient is used for normalisation of control signals by the comfortable contraction level.

3.4.8 Experiment Set-up

The participants were familiarised with the system during a short (15 min) block, in which they had to imagine the act of their wrist flexion and extension alternatively. The users could view the raw EMG data produced by their muscles visually. At the same time, the system was calibrated and an activation threshold for the two control signals was empirically tuned based on the strength of subjects' muscle signals. In this way, the subject's capability in comfortably contracting the two muscle groups independently was confirmed. As the subjects benefited from real-time visual feedback of their muscles' activity on a monitor, they were asked to make

sufficient effort to reach the threshold for an EMG channel by activating its relevant muscle group while maintaining the control signal of the other muscles group below its threshold. This calibration technique is further explained in [142, 144]. Due to a change of electrode postures for subject D, the control signals were recalibrated in the middle of the experiment.

Processing System Specifications

All the experiments, offline and real-time, were implemented in MATLAB®. A personal computer with an Intel Core i5-47670 CPU (3.4GHz), running on a 64-bit Windows 7 operating system, with 32GB RAM was used for the offline and real-time computer-based tests.

For the real-time test including amputee subjects, a Lenovo laptop with an Intel Core i7-4559U CPU (2.10GHz), running a 64-bit Windows 7 operating system, with 8GB RAM was used.

i-limb

For real-time tests with the presence of amputee subjects, an open source i-limb Ultra hand prosthesis (Touch Bionics, an Össur HF company) was used. Proportional control of individual digits in the prosthetic hand was provided wirelessly via Bluetooth with a MATLAB-based driver. A pair of rechargeable batteries (7.4V) were used for the hand power source.

Wrist Rotator

As the i-limb hand does not include a module for wrist rotation by itself, a prosthetic wrist rotator (Motion Control, Inc, USA) was augmented to the i-limb. This module provides clockwise and counter-clockwise rotation of the hand. To actuate the wrist, an in-house built bidirectional (H-bridge) drive was designed and implemented. The power for the wrist module was supplied via a 7.4V doubly insulated power supply. Rectangular TTL (5V) pulses generated with a USB-6002 data acquisition system (National Instruments, USA) controlled the rotation of the rotor.

Webcam

The same Logitech Quickcam Chat webcam used for computer-based experiment was attached to the dorsum of the i-limb in the real-time experiment with the presence of amputee participants. A double-sided velcro was attached to the i-limb to enable attaching a webcam to it.

This way the user was able to capture images of object by pointing out to them. These images (640×480 pixels resolution) were then recorded into a recording laptop via a USB connection.

3.4.9 Experimental Protocol

To follow similar protocol as previous experiments, subjects were asked to sit in front of a table, on which an object was placed with approximate 60cm and 15cm longitudinal and altitudinal distance respectively from the subject. Such setting ensured that the objects are readily available for reaching and manipulating and the taken images were comparable with the training images. In each trial, an object was placed in front of the participant by an experimenter and he tried to grab it and move it to the target location. Therefore, there was no occlusions in the scene.

The real-time test with amputees attendance consisted of 6 blocks, in which participants gripped, transferred and placed 24 common objects (6 in each grasp group). The objects were placed in front of the subject one by one in a pseudo-randomised manner. The same exact order of objects was repeated in each block.

The user could benefit from visual feedback on both EMG signals and webcam's view in blocks 1 and 2. That is, the raw EMG singals measured by electrodes and their calculated control signals were demonstrated to the user on a monitor. In addition, they could see the field of view covered by webcam through the video stream and learn how to point the hand to the object. They could also see the snapshot taken after camera activation and the proposed grasp type suggested by the model in the computer screen.

For blocks 3 and 4, the camera feedback was removed and the raw EMG signals and their corresponding control signals remained on the screen.

In block 5, there was no visual feedback for users at all, closely resembling a real-world scenario. No visual feedback was also offered to participants in block 6 similar to block 5. Block 6 however differed from block 5 as the users could benefit from the error correction feature. They could repeatedly discard a suggested grasp when observed it in hand preshaping stage and re-aim at the target object and capture a new snapshot. In this way, the CNN had the opportunity to be fed with a better input image and provide a more accurate grasp recognition. This feature however was not evaluated for subject D due to technical reasons.

Such composition of familiarisation (blocks 1 to 4) and testing (blocks 5 and 6) procedure provided a natural and efficient flow for the experiment. Consequently, the data and results of all the blocks were analysed and assessed comprehensively.

In the experiments including subject M, a 3-seconds interval in the beginning was consid-

ered in each trial so that the subject can comfortably be prepared for the trial. After the subject performed few pick and place steps, it was observed that the user's enthusiasm distracted him from waiting for the end of this interval. Therefore, this approach was not effective as the subject's wrist flexor muscles were already flexed at the beginning of each trial to capture a snapshot. In the experiment including subject D, this issue was fixed through a slight change in the protocol. That is, the start of trials were notified to the user via an audio beep. The second change in the protocol for subject D was shrinking the preshape period of the prosthesis in favor of better responsiveness. In the results reported in Section 3.5, the impact of these adjustments are observable in total trial duration. These adjustments however did not influence the overall performance and more importantly the deep learning structure as it provided grip modes for target objects only based on a single snapshot of a target object.

3.5 Results

As explained in previous section, there are three main sets of experiments: Offline, computer-based real-time and real-time experiments with amputee subjects. Therefore, the results are also reported based on these categories.

As a clarification step, the datasets used in each test, the condition and origin of test and train images are included in Table 3.6.

Table 3.6: The datasets used in different experimental conditions including how relevant test images were captured. NCL stands for the Newcastle grasp library.

Dataset / Condition	Train dataset	Test dataset
Offline	ALOI+NCL(DSLR)	ALOI+NCL(DSLR)
Real-time, computer	ALOI+NCL(DSLR)	Webcam
Real-time, amputee	ALOI+NCL(DSLR)	Webcam

3.5.1 Offline Grasp Classification

The first set of results includes offline tests, for which a combination of the ALOI and Newcastle Grasp library (with full resolution) were used. The offline grasp classification was performed to firstly investigate the feasibility of using a CNN for grasp recognition of common objects. After achieving promising results, more structured tests were carried out to further fine-tune the CNN structure and realise the best architecture and parameters for real-time experiments.

The results of both WOC and BOC schemes are illustrated in Figure 3.13. The results include the WOC and BOC experiments with the one- and two-layer CNN structures on the combined ALOI and Newcastle grasp libraries.

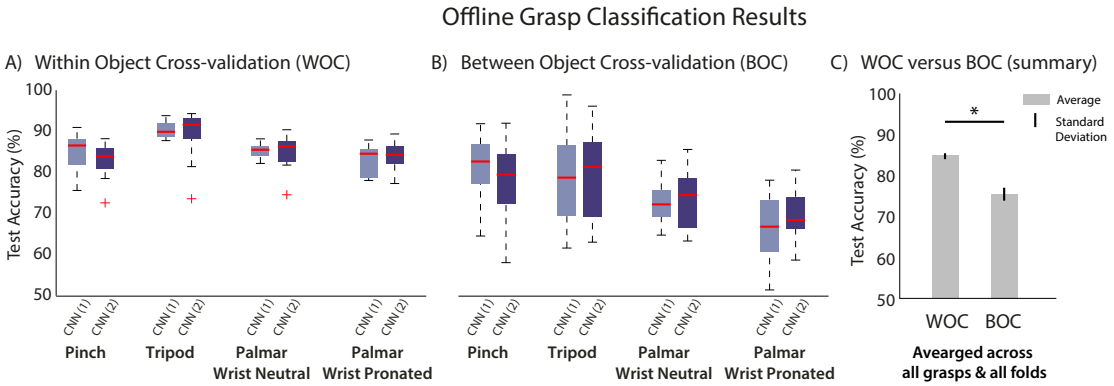


Figure 3.13: Comparison of grasp recognition accuracy of offline experiments. A and B: Grasp classification accuracy for within- (left) and between- (middle) object cross-validations (10 folds). CNN(1) and CNN(2) respectively stand for one-and two-layer CNN structures. Boxplot description: horizontal red lines, medians; solid boxes, interquartile ranges; whiskers, overall ranges of non-outlier data; red crosses (+), outliers. C: The representation of the overall performance in WOC and BOC tests in terms of average classification accuracy together with standard deviations. $*$ denotes statistical significance.

The average grasp identification accuracy in WOC setting was 80.0% and 79.9% for one- and two-layer CNN respectively. The statistical difference between these two results, measured by a repeated measure two-way ANOVA test was negligible ($n = 10, F_{1,9} = 0.001, p = 0.98$). Nonetheless, the WOC and BOC average classification accuracy results (WOC: 85.29% versus BOC: 74.74%) as shown in Figure 3.13C represented significant statistical difference ($n = 10, F_{1,9} = 32.08, p < 10^{-3}$). This difference was expected as the task of generalising to novel objects (BOC) is much more challenging than generalisation to new views of the same object (WOC).

Also in the BOC scheme, two-layer CNN achieved 0.7% higher grasp classification performance than the same structure with one layer (1-layer: 74.38%, 2-layer: 75.10%). The better results of the two-layer CNN however were not of considerable statistical priority (post-hoc analysis with a paired t-test, $t_9 = 0.28, p = 0.78$). Nevertheless, the two-layer CNN was opted to be used in the consequent experiments as the average performance in three out of four grasp classes was prior to the one-layer CNN, Figure 3.13B. As there were 10 training folds for each offline structure, one of these trained CNNs with a reasonable performance in grasp identification of unseen objects was picked for the real-time tests. Consequently, a CNN model which led to $\sim 70\%$ average grasp classification accuracy on novel objects from the range of models

with performances from 64% to 75% was used.

3.5.2 Computer-based Real-time Performance Analysis

After examination of the deep learning-based grasp classification system and gaining a baseline for system's performance, it was implemented for computer-based real-time experiments, in which the images were taken with the webcam.

Performing the real-time computer-based experiment on six distinct objects in each grasp group and seven *random* views of each provided the opportunity of system evaluation in a situation resembling the real scenario independent of user relevant factors.

The grip recognition results of real-time computer-based tests are indicated in Figure 3.14. The figure distinctly demonstrates the suggested grasp by the model for each specific view and object. Ideally, in case of 100% correct grip classification, each bar should be shown in a single solid color. Any inconsistency in the colors composing an object's bar would be counted as a miss-classification of a particular view of that object. For instance, there are 5 mistakes in total 42 trials for *pinch grasp* and all the incorrectly suggested grasps are tripod.

The required time for grasp recognition of a low-resolution snapshot was recorded during the test. The procedure of feature extraction and classification within the CNN takes 78 ± 6ms and 3 ± 0.03ms, respectively.

3.5.3 Real-time Test Platform with a User in the Loop

The last set of results includes the amputees performance within the 6 blocks of object pick and place with the hand prosthetic in the real-time platform. Two trans-radial amputees were involved in these experiments as a proof of principle. Figure 3.15 manifests three trials of experiments with subject M. The recorded EMG signals, the captured images and their relevant time instants and identified grasps are all illustrated in this figure. Moreover, the trials are selected such that almost any kind of observed situation is depicted. That is, the first trial, Figure 3.15A, indicates a full success, in which the object image was fully acquired by the user and the system suggested the correct grasp type. The pick and place duration ($\sim$ 7s) is also counted as an average performance with respect to the time of procedure. In the second trial, shown in Figure 3.15B, although the CNN model suggested an incorrect grasp type, *palmar wrist pronated* instead of a *tripod*, the user accepted the grasp and proceeded with pick and place. Finally, the last trial demonstrated in Figure 3.15C represents a trial in which the error correction feature is enabled. The user initially failed to obtain a reasonable image of target

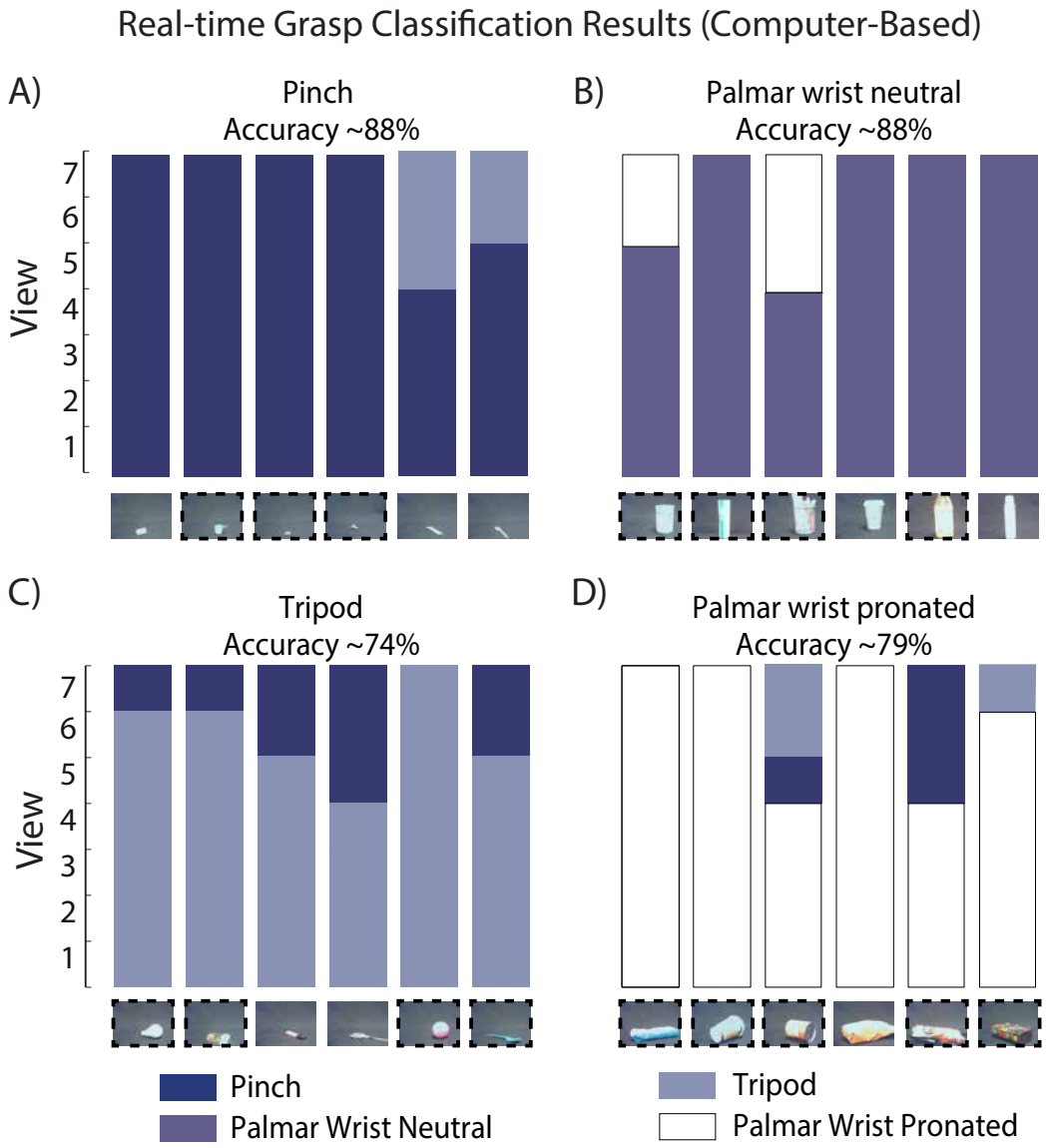


Figure 3.14: Two-layer CNN architecture average classification performance for four grasp types and seven random views of several seen and unseen objects in on-line computer-based test. All images were converted to grey-scale and downsampled before further analysis. Objects shown with dashed black box around them were novel to the classifier. While all other objects were seen by the classifier, they were rotated randomly for this test. In the case of 100% correct classification, each bar would be shown in a single colour. Thus, an inconsistency in each object's bar illustrates a misclassification.

object due to improper orientation of the hand. Repeating the snapshot acquisition procedure three additional times led to a valid image of the object followed by correct grasp recognition by the CNN and further trial accomplishment.

Real-time experiments involving amputee subjects were performed on 8 seen, but randomly-rotated, objects as well as 16 novel objects. This organisation of objects led to investigation of the implemented structure in both within- and between-object generalisation tasks with more attention focused on the latter. A summary of all the achieved results in real-time experiments

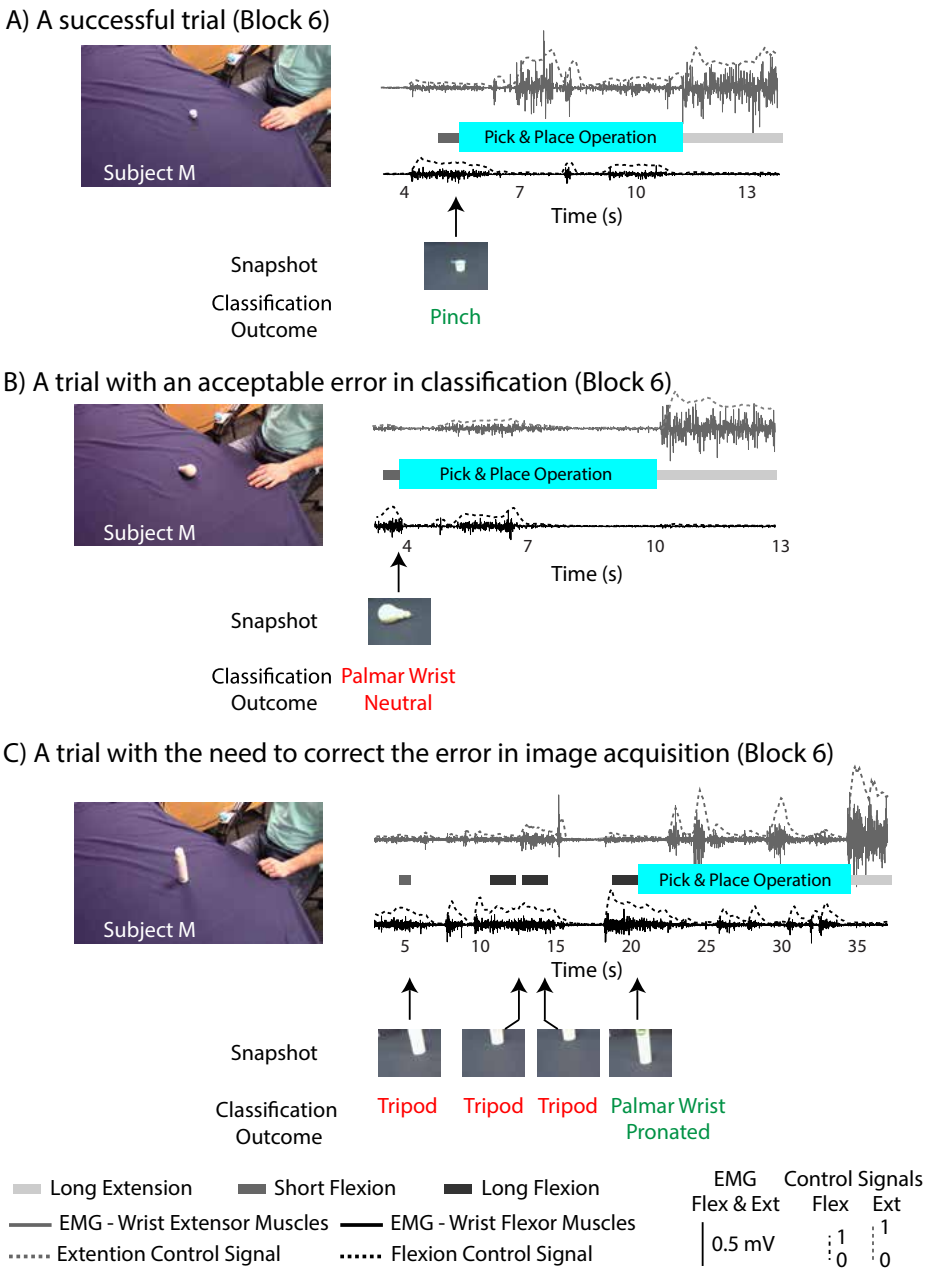


Figure 3.15: Three selected real-time trials accomplished by participant M; A) A successful trial with valid input image and identified grasp; B) A trial representing the trials in which erroneous decision was made by the CNN (*palmar wrist pronated* instead of *tripod* grasp), but the user proceeded with the trial successfully; C) A trial including error correction in which unacceptable classification error (*tripod* instead of *palmar wrist neutral* grasp) was made by the model due to user's arm misalignment. However, repetition of image capturing step led to correct grasp identification.

are indicated separately for subject M (left column) and D (right column) in Figure 3.16. The average grasp identification accuracy within each block are shown in Figure 3.16A for each grasp group.

It is worth highlighting that for blocks 1 to 5, only correct classification of objects, which is when CNN output exactly matches the assigned label of an object, are considered in the

results. The same consideration was applied to quantification of block 6 except that since error correction was enabled, this validation method was only applied to the last attempt within each trial. The overall accuracy of blocks 1 to 6 across all grasp groups are depicted in Figure 3.16B.

Additionally, the number of trials, which included mistaken classification outputs accepted by the subjects and ended up successfully and the ones led to a failure were counted and reported. In case of the latter, the experimenter ended the trial whenever the user failed in trial completion. Thanks to the error correction feature in block 6, the overall success rate and therefore average performance were higher in this block compared to all the other blocks; that is, 79% versus 73% for subject M and 86% versus 73% for subject D. Including the acceptable errors (error subtype 1; as explained in Figure 3.16B) in the overall performance boosted the overall performance within the 6 blocks, which is 88% and 87% average grasp success for subject M and D respectively.

Figure 3.16C indicates the average time each subject spent for trial accomplishment within each block. The first block was the most time consuming one for both participants. The time accomplishment curve represents a decreasing pattern from block 1 to block 6 for both subjects. The significance of this reduction was further investigated and is described in more details. For subject M, the decline in the task accomplishment time (across the 24 trials) in block 6 versus block 1 was only marginally significant (block 1: 21.4 ± 8.1s, block 6: 16.7 ± 9.3s, paired t-test, $n = 24$, $t_{23} = 1.81, p = 0.08$). However, for subject D, this reduction was in fact statistically significant (block 1: 30.7 ± 17.2s, block 6: 19.3 ± 25.7s, paired t-test, $n = 24$, $t_{23} = 2.26, p = 0.03$). Such a decrease in the accomplishment time in spite of the elevation in the task difficulty (removal of available feedback sources) can be interesting.

The time required for grasp identification of a low-resolution input image was recorded during the performed real-time experiments within the implemented graphical user interface. The average time took for pre-processing and classification were respectively ~ 110ms and ~ 40ms using the laptop specified for real-time tests. It is worth to note that snapshot capturing requires the subjects to flex their muscles shortly such that the activity of flexor muscles exceeds its relevant threshold for 300ms, whilst the activity of extensor muscle group maintained below its relevant threshold. As such, the total time required to achieve a correct classification sums up to ~ 450ms. All the mentioned time stamps are included in Figure 3.16D.

As a final step, the capability of the designed system in recognising an appropriate grasp type for novel objects was evaluated. As a result, Table 3.7 includes the achieved results during the real-time experiments with each subject divided based on objects being seen or not by the model. As mentioned previously, 8 out of 24 objects in each block were seen and 16 items were

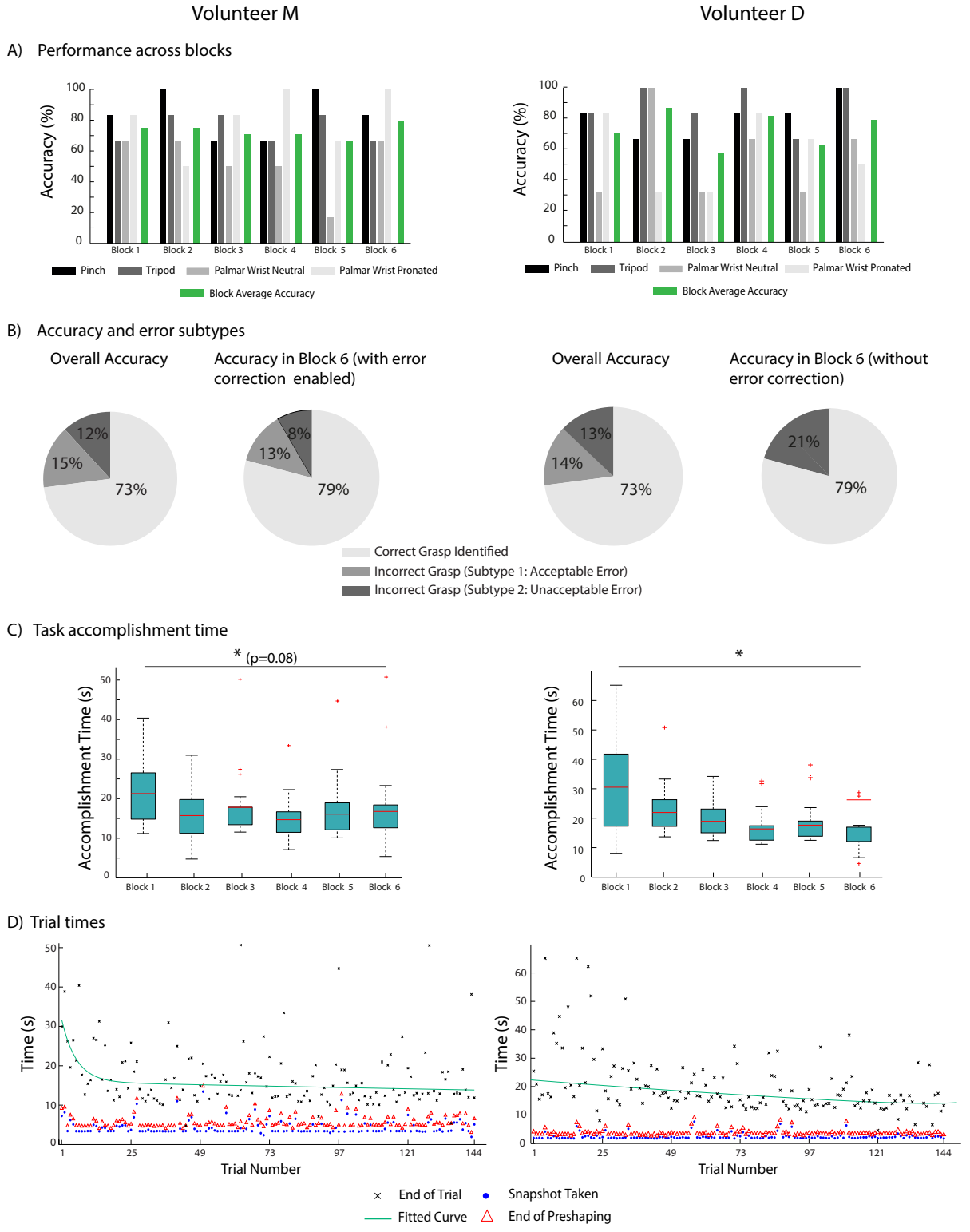


Figure 3.16: Evaluation and quantification of real-time performance of suggested system for both subjects, subject M on the left and subject D on the right: A) Average grasp classification accuracy of each grasp type within each block and the overall performance of each block. B) Overall success rate of the grasp task within all blocks and block 6 considering the error being acceptable or not: error subtypes 1 and 2. C) Task accomplishment time within each block shown in standard boxplots. D) Accomplishment time of each trial shown in details including the snapshot, the preshape and the end of trial times. $*$ denotes statistical significance.

unseen by the trained two-layer CNN. The results indicated that determination of success for seen or unseen objects would be of the same possibility.

Table 3.7: The average success rate of each subject in the real-time experiments with respect to the objects being seen or unseen. Specifically for volunteer M in block 6, which included the correction of errors, the reported performance considers the *first* identified grasp, that is before error correction.

Block \ Volunteer	M		D	
	Seen	Unseen	Seen	Unseen
1	75%	75%	50%	81.2%
2	75%	75%	75%	93.7%
3	62.5%	75%	50%	56.2%
4	37.5%	87.5%	87.5%	81.2%
5	75%	62.5%	62.5%	62.5%
6	63%	75%	87.5%	75%

3.6 Transfer Learning

3.6.1 Architecture

To seek for better and deeper network architecture choices and provide a reasonable comparison with popular off-the-shelf architectures, the implemented model was adapted accordingly. To this end, the two-layer CNN implemented in Section 3.4 was substituted with a deep network (ResNet-50), one of the well-designed deep learning architectures with highly promising results on object recognition tasks [112]. The ResNet architecture is popular for its residual connection in between the layers, which improves the gradient flow, prevents information loss and provides better learning as a result.

To elevate the learning capability of the model and resultant performance, instead of training the new network from the scratch, a pre-trained version of that, trained on images from the ImageNet dataset [147], a huge dataset of object images consisting of 1000 categories, was utilised. It was demonstrated that CNN features learned over the ImageNet are of valuable amount of information and can be exploited for other tasks [113, 148]. It was also shown that the features learned by training a CNN on the ImageNet are so rich that they can substitute random weight initialisation and boost the training performance significantly [113, 148].

Consequently, a pre-trained ResNet-50 was adapted for the task of grasp classification. That is, the last fully-connected layer of the ResNet-50 consisting of 1000 nodes to predict 1000

categories of objects in ImageNet was replaced with a new dense layer to be able to predict 4 grasp classes of *pinch*, *tripod*, *palmar wrist neutral* and *palmar wrist pronated*. Similar to Section 3.4, the augmented ALOI dataset was fed into the pre-trained ResNet-50 for fine-tuning the model. The successive section presents the results achieved with this adapted architecture.

3.6.2 Results and Analysis

A ResNet-50 model previously trained on the ImageNet dataset was fine-tuned using CNTK (Microsoft cognitive toolkit [149]) using an NVIDIA Geforce 960M GPU. As previously mentioned, the last fully connected layer of ResNet-50 responsible for predicting labels was substituted with a new dense layer to adapt the network and existing weights to predict 4 grasp types. The network was then fine-tuned and corresponding weights were optimised during 30 epochs with the parameters present in Table 3.8.

Table 3.8: The parameters set for fine-tuning the pre-trained ResNet-50.

parameter	value
mini-batch size	50
learning rate	0.2
momentum	0.9
Image height	224
Image width	224
number of channels	3

After evaluating the test set, a combination of seen and unseen objects, 3194 out of 4464 predictions were correct, which led to 71.55% average accuracy. The per class average accuracy is reported in Table 3.9.

Table 3.9: Average test accuracy of a fine-tuned ResNet-50 per grasp type.

Grasp	Pinch	Palmar wrist neutral	Tripod	Palmar wrist pronated
Accuracy	71 ± 0.073	86 ± 0.034	69.5 ± 0.082	64 ± 0.084

These results suggest that the implemented structure involving the two-layer CNN with the performance of $\sim 75\%$ (Section 3.4) is superior to the pre-trained network for the task at hand. This outcome further confirms the validity of the designed platform used in Section 3.4.

3.7 Discussion

In this chapter, a commercial prosthetic hand was augmented with a webcam and a deep learning-based structure to improve the grasp ability of the commercial hand prostheses. After comprehensive examinations and gaining certainty over the feasibility of the implemented structure through offline and real-time, but computer-based experiments, two trans-radial amputee subjects tested the functionality and performance of the setting. Only one hour after practicing with the implemented structure, the participants managed to accomplish $\sim 88\%$ of trials successfully.

Currently, the commercial hand prostheses employ a variety of *workaround* solutions to approach the limitations of EMG-based hand prosthetics. That is, they require the user to either learn and perform different co-contractions or follow particular trajectories [150] or have objects with RFID tags in the surrounding environment to be able to switch between different grip patterns. These comparatively simple techniques are preferred to EMG pattern recognition-based methods despite the exceptional performance promised by them as EMG-based pattern recognition methods failed in providing reliable clinical performance and robustness.

The aforementioned *workaround* techniques are also not intuitive and flexible and have limited performance capability. As a solution, other sensor modalities namely accelerometry or in general inertial measurements [151–154], RFID tags [155], artificial vision including monocular cameras as well as depth or RGB-D cameras [10, 11, 41, 79, 156, 157] were used to support or substitute the EMG signal information. The multi-modal approaches are mainly pointing out that the incorporation of two or more sources of information for the control of prosthetic limbs leads to less cognitive burden on user and higher performance accuracy.

In the presented setting, vision as an additional modality was exploited. To that end, a CNN architecture was adapted and trained to classify a single low-resolution snapshot of a common object based on its appropriate grip pattern. As a result, the subjects managed to effectively use four different grasp types for picking the target objects in real-time.

3.7.1 Dataset

Deep networks require large amount of data to be trained properly [13]. The more variety and abundance the provided data benefits from, the better performance and robustness can be achieved with a deep network. Therefore, to train a CNN-based architecture to recognise appropriate grip patterns for common objects, sufficient number of object images should be fed to the network. Finding a dataset with adequate amount of data merely focusing on graspable objects

can be challenging as available large datasets such as ImageNet [15] involve several objects that are not graspable.

Initial experiments with a CNN-based architecture for grasp classification was performed with the COIL100 dataset [18] including 100 categories of graspable objects. The best results achieved in the experiments with COIL100 led to 97% and $\sim 55\%$ average classification accuracy in the WOC and BOC settings respectively [157]. The huge drop in the BOC setting is attributed to the lack of generalisability caused by insufficient number of objects in the training data. This problem is specifically more intense in the *palmar wrist pronated* group, where least number of objects are included.

As a solution to this data insufficiency, the ALOI dataset [140] with 1000 object categories was used in the successive experiments. Although not all the 1000 categories were utilised and only ~ 500 of object classes were picked for training and analysis, more variety of objects and more samples were accessible in the ALOI. The objects not used were either repetitive in shape or caused excessive number of objects in one class and therefore neglected. To compensate the class imbalance introduced by less number of object categories in some classes and enable real-time testing where original objects are not accessible, 71 objects were collected such that the class imbalance is alleviated. These objects were photographed at Newcastle University in a similar way as the ALOI objects. The augmented image dataset was then used for further training and analysis and did not show any problems relevant to lack of data or class imbalance.

3.7.2 Object Classification versus Grasp Classification

The task of grasp recognition is considered as a supervised learning task and therefore lacks the ability to provide recognition for the categories not specified before training. Therefore, if a network is trained to do object identification, novel test objects will be assigned to an incorrect object class. This feature however is essential for prosthetic hands as people grasp novel objects every day.

A naiive solution to this problem with supervised grasp classification can be addition of a huge amount of data including every possible object class. Nevertheless, this solution is too costly and not efficient. A better way to tackle this problem is to rely on the capability of deep learning structures in adapting and generalising to new tasks. That is, rather than classifying objects based on their object category, they can be categorised by their suitable grasp type. In this way, the output space is shrunk to a smaller output space limited to the number of usable grip patterns. Additionally, the deep network can be adapted to learn an abstract representation

of each grasp class and therefore generalise to objects novel in shape and appearance.

3.7.3 The Network Design Considerations

In this chapter, two CNN architectures were designed and exploited for grasp classification task. The fundamental difference between the one- and two-layer CNN structures is the extra convolution layer for the latter. The additional layer was utilised for learning more abstraction due to depth improvement. However, this added depth did not lead to any statistical difference in the network performance. Increasing the number of convolution layers more than two also caused a drop in the classification accuracy. This drop can be attributed to the small size of input images and their simple and sparse representation (single object in black background). The size of images were reduced to 36×48 pixels due to the limitations caused by memory and CPU and therefore deeper models were not used further. The two-layer CNN was chosen for the real-time experiments as it provided better classification accuracy in three of four grasp classes compared to the one-layer structure (Figure 3.13B).

A concern about the deeper models is over-fitting phenomena, in which the network parameters are overly tuned for a specific dataset and cannot generalise properly to new data. The deeper the model, the higher the chance of over-fitting caused by more number of parameters. This problem however was prevented by regularising the CNN structure through the Tikhonov regularisation, which penalises the weights matrix $\mathbf{K}_j^l$ during optimisation.

The performance of both CNN structures in real-time was comparable with negligible differences. Despite the long training time of the deep networks, the test time is very fast since only the forward path is followed once. The training and testing time of the two-layer CNN were respectively about 2 hours and ~ 150ms on a CPU (3.4GHz). The test time was in fact dominated by the image pre-processing block taking ~ 110ms to provide the proper input to the CNN as presented in Figure 3.11. All the experiments were carried out using MATLAB via a CPU. The usage of a GPU thanks to the great speed and parallelisation power it provides can boost the real-time implementation speed significantly.

To understand the type of CNN features extracted for grasp recognition, the feature maps achieved from the application of the two-layer CNN are visualised in Figure 3.17. In this figure, for each grip pattern the activation maps of the second convolution layer (outputs of the ReLU layer in Figure 3.8) for two example objects are shown. These activation maps are resulted from extracting learned features in two consecutive convolution layers and therefore indicate the kind of features the network mostly focuses on. Figure 3.17 suggests that the concentration of the

network is mainly the objects' orientation, contour and size.

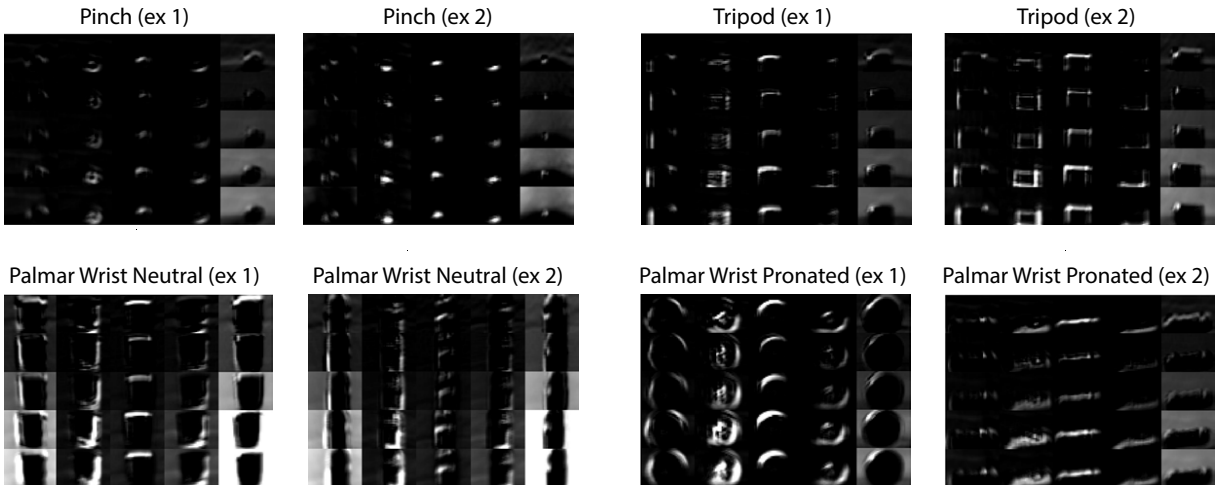


Figure 3.17: Representation of learned features for two sample objects of each grip category. These features can be observed through the 25 resultant maps after the second convolution layer. Probing these feature maps suggests that generalisation may be achieved due to the abstract object features being size and orientation of the objects.

3.7.4 An Alternative Approach for Error Correction

Typically, a CNN architecture ends with one or more fully connected layers for better local connections and weight sharing. The fully connected layer is usually accompanied with a Softmax regression classifier [15, 101, 158]. In this work the same procedure was followed for classification. The benefit of the Softmax to other classifiers such as SVMs is that the former can be trained within the CNN and therefore provides an end-to-end learning procedure, which is preferable.

The outcome of the Softmax is class probilities, such that the class with highest probability is selected as the predicted class. The interesting point here is that the probabilities for other classes are also available. Hence, as an error correction policy, the second most probable class can be picked in cases where the prediction is not desirable. Although this approach offers an automatic error correction mechanism, it was not used in the real-time experiment due to the following reasons:

- Having a vision-based prosthetic hand makes the prosthesis more autonomous [11] and decreases the amount of user supervision. During the familiarisation and initial blocks, it seemed that the subjects were inclined to have more control over the function of system. Therefore, only the manual error-correction approach was experimented. This approach

provided the participants with more supervision in block 6 such that they could even reset the hand prosthetic to the neutral grip and capture a new snapshot.

- Both volunteers were naiive to both the myoelectric prosthesis and its usage and the concept of the experiment. Additionally, the experiment included 6 blocks and took more than 2 hours, which could make them exhausted. Thus, in spite of the attraction of performing an automatic error correction experiment, it was not performed since the results could be degraded due to subjects' fatigue and being unprepared for more complications.

3.7.5 Possibility of more Number of Grip Patterns

As this work mainly presents a proof-of-concept and due to the limitation of the dataset which contains objects belonging to four grip classes, the number of grasp types was limited to four.

With the method presented in this chapter, the number of grasp types can be increased as long as sufficient relevant data is provided. That is, the lateral grip was not considered as a possible output in this work due to lack of relevant objects. The objects belonging to this grasp class represent a particular flat shape, *e. g.* a card or a key, which is distinct from the objects present in the augmented dataset. This distinctive feature can be highly beneficial for grasp recognition and therefore by extending the current dataset with adequate images of objects requiring a lateral grasp, similar recognition performance for 5 grasp types can be achieved.

One controversial aspect about the number of grasp types is that in current commercial hand prostheses the users end up with few patterns as switching between grasp types is exhausting. This problem should not occur in the suggested system as the grasp type selection is performed autonomously.

3.7.6 Performance in the Presence of Clutter

Having a cluttered scene or an arbitrary background can cause complications to the presented system. The real-time tests were all carried out on a single object placed in a scene with simple dark background.

One approach that can eliminate the need of segmentation is to augment the segmented objects on random backgrounds and train the network with such images. This method can lead to an end-to-end grasp classification within a CNN network. The applicability of this method for real-world scenes however requires more investigations.

Another helpful method to deal with clutter and background change can be utilisation of

RGB-D sensor. For instance, previous works on vision-based prosthetic [10, 11] benefited from 3-D pointclouds for scene segmentation. Nonetheless, the ALOI dataset does not contain depth information and therefore this method is not applicable to this dataset. Training a CNN with sufficient RGB-D images of common objects can be an interesting future study.

3.7.7 Real-time Performance: Computer-based versus Human Experiments

Computer-based, real-time experiments were performed to investigate the capability of the system for grasp recognition of objects in a similar to real world scenario. This was independent of user-relevant parameters, namely camera view and distance to the target object. The average accuracy achieved in this test was 84% outperforming that achived in real-time experiments with amputee users in the loop. This superiority in performance is specifically highlighted in initial blocks, where users benefited from two sensory feedback but lacked the adequate experience with the system and thus not reaching the performance of computer-based real-time experiment. Although after being acquainted to the system within initial blocks, the performance of both subjects were boosted considerably, it was still lower than the accuracy in the computer-based test. The higher performance of the computer-based experiment can be attributed to the fixed camera view and distance to the objects. Other intrinsic parameters can also play a role in degrading the performance in the real-time experiments with the amputee users in the loop, namely physical and mental fatigue.

3.7.8 User Training with Full or Partial Visual Feedback

Similar to previous work [41], the camera was mounted on the dorsum of an i-limb hand. This can be slightly unnatural as the user has to point to the target object for performing a grasp. Placing the camera on user's head, as in [11], can be more intuitive. However, that may not provide a pleasant user experience as an additional module should be worn. Hence, keeping everything in one module was preferred in this work. To provide a better user experience and adapt the users to the system function, subjects were trained step-by-step through 6 blocks to be able to localise the target object and fit it in the camera view properly. The most challenging objects were the objects with long vertical dimension belonging to the *palmar wrist neutral* group, Figure 3.15C.

After a short familiarisation block, there were two measurement blocks in which the subjects benefited from the camera and EMG signal feedback. Although it was presumed that users

prefer to keep the visual feedback rather than the EMG signals, they were more comfortable with eliminating the camera feedback. The performance within blocks 3-4 also emphasises on this comfort as there is no drop in the overall classification accuracy.

3.7.9 User Experience

Both participants provided positive feedback on the suggested vision-based prosthetic hand control structure. For instance, subject D said: "*Just getting the routine was difficult at the beginning but once this was established it became much easier. If it would be further refined [in terms of positioning of camera] I would certainly use this and always give feedback*". Subject M tested two prosthetic control systems on the same day, the proposed platform and a novel pattern recognition system. When asked which of the two approaches he would prefer, he responded: "*I'd like the pattern recognition better, when it works perfectly! For the time being, the vision-based system seems to be a good solution. I liked its responsiveness very much*."

3.7.10 Pre-trained CNN v.s. CNN with Randomised Weights

As features learned by training a deep network with ImageNet are rich in information, it is shown that fine-tuninig a pre-trained network on ImageNet is beneficial compared to training from scratch with random weights [113, 148]. This idea was also investigated in this work and a ResNet-50 [112] model previously trained with ImageNet dataset was fine-tuned with the augmented dataset provided in this chapter (ALOI and Newcastle grasp library) for the task of grasp classification. Despite the excessive depth of the architecture, rich initial weights provided by ImageNet and exceptional architecture of ResNet, the overall accuracy did not improve. One reason could be the sparse and simple representation of the training data. Also these deep networks require huge amount of data to tune their large number of parameters and the train data provided was not probably sufficient to tune the network properly. Data augmentation techniques can be helpful in elevating the number of train data. Therefore, it is open to further investigation if additional amount of data and more complex input images can improve the results for transfer learning.

3.7.11 Importance of Generalisation

Object recognition and WOC tests gained high average accuracy. However, all the objects are considered as "seen" in these experiments, which is not practical in daily life. In real life

a large number of unfamiliar objects are needed to be apprehended and failure in this task makes the idea of vision-based prosthetic hand impractical. Although the offline BOC test results are lower than that of the WOC, the average accuracy is still promising as the platform presented reliable performance in real-time tests. Additionally during the real-time tests no specific difference was observed between the classification accuracy for novel and seen objects. Hence, the proposed structure is able to efficiently identify the appropriate grip pattern for any object type independent of its familiarity to the object. This ability to generalise to novel objects makes the proposed structure a promising fit for real-life applications and consequently a good candidate for vision-based prostheses.

3.8 Conclusion

This chapter proposes an efficient vision-based prosthetic control structure. This structure consists of a prosthesis hand augmented with a vision-module responsible for automatic grasp selection for an object of interest. The procedure requires the user to target an object with the prosthesis, such that commands recorded from the amputee's arm trigger the camera to capture a snapshot. The snapshot is then processed to segment the target object properly, which is fed into a CNN for classification of received input into four different grasp types. The user observes this decision as a preshape act in the prosthesis and proceeds to grabbing the object of interest and moving it. If the preshape is not desirable, the user can reset the camera and take another snapshot until the appropriate grasp is provided. The proposed structure provides a fast and efficient automatic grasp recognition setting with a promising performance for unseen objects almost as good as that of seen objects.

Chapter 4

Grasp Map Estimation using Fully Convolutional Residual Networks

In this chapter, a novel grasp estimation method is proposed for detection and estimation of grasp maps for novel objects. This approach accounts for the significant ambiguity involved with the task of grasping and therefore redefines the problem such that this ambiguity is dealt with properly.

4.1 Motivation

In this chapter, the problem of grasp recognition is approached from a robotics perspective by building up on the current advances of robotics and computer vision.

Grasping is a crucial ability for an autonomous agent to have interactions with its ambient. Object grasping and manipulation play an indispensable role in a variety of applications in the field of personal robotics and manufacturing. The grasping performance of robotic systems is however behind the human performance even in simplified environments. Humans can conveniently grab and manipulate various objects while for robots this is still an unresolved problem. Such an issue in the task of grasping is specifically more challenging when the robotic system encounters objects in new positions, orientations or categories. Robotic grasping is therefore a highly challenging task, involving several steps that should all be taken in account, namely perception, planning and control.

Robotic perception as the main concern of robotic vision is commonly involved with the detection of viable grasping locations. Visual recognition from sensors, such as RGB-D cameras, is required to perceive the environment and transfer candidate grasp points from the image

domain to coordinates in the real world. A mandatory first step for successful manipulation through an end effector, such as a robotic hand or a gripper, is the localisation of reliable and effective grasping points on the object surface. The suggested grasping position can later be employed for realisation of the optimal trajectory for grasp execution.

The importance of this visual recognition task was raised significantly during the recent years, with a wealth of solutions proposed in literature [17, 26, 125–134, 136, 137] and the emergence of benchmark datasets, such as the Cornell grasp detection dataset [17], to evaluate the performance of suggested methods for this particular task.

Traditional grasp detection techniques were concerned with grasp point localisation through explicit estimation of the geometry of target object [129, 133]. While reliable grasping points are offered through such procedure, the runtime is extremely slow and the performance drops significantly in presence of complicated or unseen object shapes. The advances of deep learning and its success in a variety of computer vision applications have led to several recent approaches [17, 126, 127, 131, 135–137] to successfully detect grasping points from visual data, typically in the form of *grasping rectangles* [17, 26]. These approaches mostly employ RGB data together with the depth data for grasp detection.

All these solutions have considerably enhanced the performance of robotic systems in grasp detection. Nonetheless, there is still room for improvement especially generalising to novel and complex shapes. Specifically, despite the endeavors of prior works in explicitly improving grasp estimation for unseen objects from RGB-D/depth data [17, 26, 125–127, 129, 134, 136, 137], this aspect is still regarded as an open issue [129].

4.2 Introduction

In this chapter, a novel grasp detection approach from RGB data is proposed. The task of grasping is inherently ambiguous with regards to the best grasp position. The research in either fields of human or robotic grasping mostly overlooked at the ambiguity involved in the task.

There are two techniques that are employed to explicitly model the ambiguity related to the task of robotic grasping. Firstly, a robotic grasp can be redefined such that it accounts for some measure of uncertainty. That is, instead of the conventional grasp representation based on bounding boxes [26] that is widely used, the grasp space can be modeled with 2-D belief maps. This dense belief estimation problem allows the model to predict a grasp distribution with spatial uncertainty and exploits the full potential of CNNs in learning spatial representations. The second solution is concerned with the procedure of grasp detection, in which there are

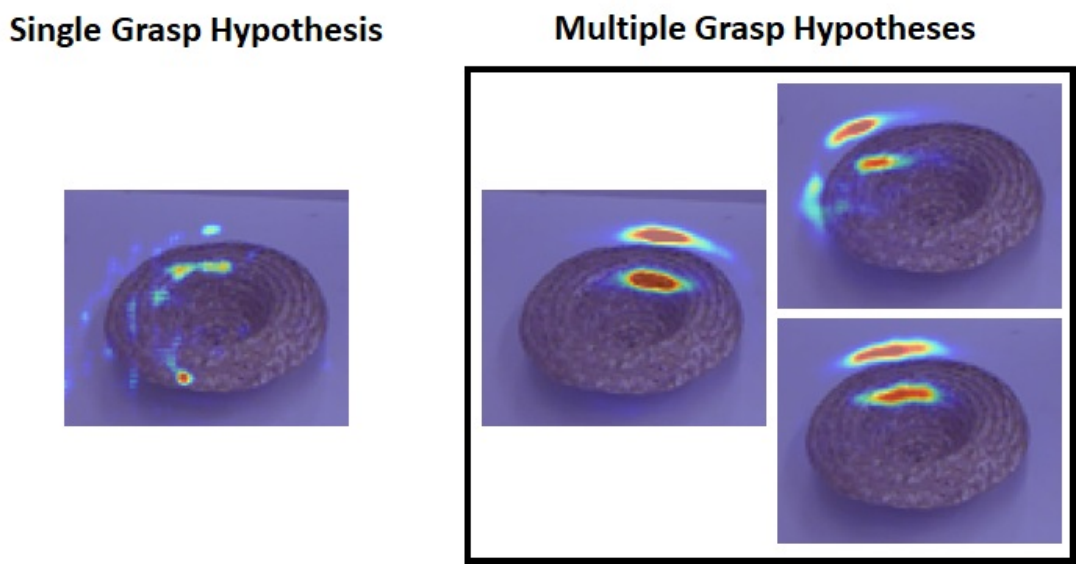


Figure 4.1: A *multi-grasp prediction* framework for regressing multiple grasp hypotheses as 2-D belief maps, which tackles the ambiguity of grasp detection more effectively than a single grasp detection, in particular for completely unseen shapes, as the one depicted here.

several possibilities for each object. To further illustrate, most objects can be gripped in different ways and, although some may be preferable, there is not necessarily a "best" grip. This issue is recently reflected in some benchmarks, which provide multiple viable grasp rectangles as ground truth for each object [17]. Forcing the system to predict a single answer for such an ambiguous problem can harm the performance as the network typically learns the conditional average of all possible outcomes. A solution could be altering the network design such that multiple viable grasp positions for each object can be predicted. The advantage of this approach is better modeling of the output distribution that leads to more precise and robust predictions especially in the case of unseen objects. An illustration of this method in comparison to a conventional single-prediction model is depicted in Figure 4.1.

Having multiple grasp options can possibly deal with the ambiguity in grasp. For an actual grasp act however only one grasp is required to be performed. To this aim, one grasp belief map prediction should be opted out of all the hypotheses. As the definition of belief maps usually incorporates Gaussian distributions [159–161], a Gaussian Mixture Model (GMM) [162] can be fitted to the predicted grasp maps to rank the predictions. This is particularly useful for practical applications of the suggested approach, as well as for the sake of comparing this work with the state of the art. The effectiveness of the proposed approach is investigated by evaluation on a common benchmark [17] against state-of-the-art methods in RGB and RGB-D grasp detection.

4.2.1 Robotic Grasp Detection

As the recent robotic grasping literature is already discussed comprehensively in Chapter 2, this section only focuses on the research works with similar approaches, which also provided results on the Deep Grasping dataset [17].

Representing a robotic grasp with a *grasping rectangle* was firstly suggested by Jiang *et al.* [26]. They designed a two-step SVM-based [163] learning algorithm to predict a grasp rectangle depicting the gripper's location, orientation and opening width. This *grasping rectangle* representation was further utilised in several popular robotic grasping solutions. The robotic grasping system suggested by Lenz *et al.* [17] is probably the most well-known work of the field, in which two techniques are introduced. Firstly, hand-engineered features are substituted with learned features through a deep network for detecting grasp rectangles. Secondly, an RGB-D view of the scene provides extra information over the depth of the objects that greatly enhances grasp detection while utilising group regularisation.

Redmon *et al.* [126] offered both single grasp and multiple grasp detection structures for RGB-D images. The former applies a single-stage regression to grasp coordinates. Their Multi-Grasp platform however employs a YOLO-CNN [139] and generalises previous model by partitioning each image into an $N \times N$ grid. The MutiGrasp approach improved the state-of-the-art accuracy of grasp detection while decreasing the detection time. However, the results were reported only for the best ranked rectangle and the performance of other suggested grasps is not known.

Wang *et al.* [136] came up with a two-stage closed loop estimator similar to [17] for grasping candidates. After suggestion of grasp candidates, a deep CNN is employed for grasp probability estimation of each candidate, which can be used as a means of ranking. Kumra *et al.* [137] suggested a CNN-based grasp estimation approach, in which either RGB-D or RGB data is fed into a pre-trained ResNet-50 architecture [112] to extract features. A successive shallower CNN is then applied to these features for regression of grasp coordinates. Asif *et al.* [134] benefited from hierarchical cascaded forests to predict the object class and grasp poses in a hierarchical point cloud decomposition framework. Finally, Guo *et al.* [127] proposed a hybrid deep network combining both visual and tactile sensing. The multimodal data is fed into a deep visual network based on faster R-CNN [118] and a deep tactile network during training. The features of both networks are concatenated as an intermediate layer to be employed in the deep visual network during test.

4.2.2 Landmark Localisation

In the proposed method, the grasping problem is defined differently. Rather than considering the task as object detection, in which the grasping rectangles are detected, as done in [17, 26, 126, 127, 134, 136, 137], the rectangles are expressed as 2-D belief maps around the grasping location. Such formulation is inspired by the recent techniques in landmark localisation, for example in human pose estimation [159, 164–167], facial keypoint detection [160, 161] and articulated instrument localisation [28, 168]. Benefiting from heat maps as a representation for 2-D joint locations has considerably escalated the performance of state-of-the-art localisation techniques. The training procedure for such models requires the output to match the ground truth heat maps, for example through $\mathcal{L}_2$ regression, and the precise landmark locations can be then calculated as the maxima of the predicted heat maps.

4.2.3 Multiple Hypothesis Learning

To properly model the distribution of a grasp for various objects as well as the uncertainty involved in the grasping task, the grasp belief maps are augmented along the lines of multiple hypothesis learning [29, 169]. Thanks to the introduction of these methods, ambiguous prediction problems can be modeled by production of multiple outcomes for the same input. These techniques however do not provide means of selecting the best hypothesis out of the resultant predictions. Here, this issue is tackled in a task-specific manner, by ranking the predictions based on their alignment with a parametric Gaussian distribution.

4.3 Methods

In this section, the proposed approach is explained in more detail. First, the definition of a grasp is updated by redefining the problem of robotic grasp detection as prediction of 2-D grasp belief maps. In particular, a mapping from a monocular RGB image to grasping confidence maps is learned via CNN regression. A multi-grasp framework is then suggested to simultaneously predict multiple grasp possibilities for dealing with the inherent ambiguity of grasping. Lastly, Gaussian Mixture Models are utlilised to score the predicted grasps so that a top-ranked prediction can be opted.

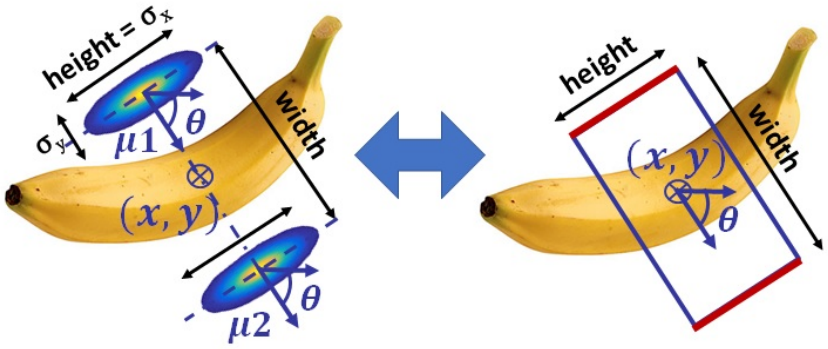


Figure 4.2: The representation of a grasp rectangle and its corresponding grasp belief map. The assigned location for the centers of gripper plates are used as the means of the normal distribution constructing a belief map. While σ_y is a chosen constant, the variance σ_x is proportional to the gripper height.

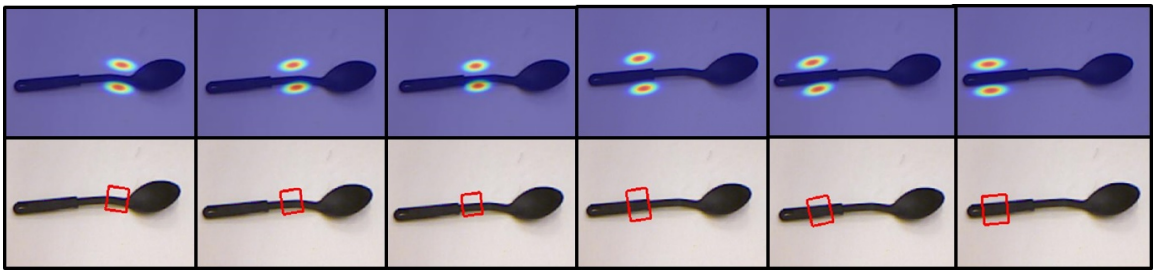


Figure 4.3: Illustration of samples of grasp rectangles and their associated grasp belief maps for the same object.

4.3.1 Grasp Belief Maps

Jiang *et al.* [26] approached the problem of robotic grasp detection by predicting the size and pose of a rectangle. This rectangle representation includes sufficient information for performing a grasp; that is a 5-D grasp configuration denoted by $\{x, y, \theta, h, w\}$, where (x, y) is the center of the rectangle and θ is its orientation relative to the horizontal axis. The parameters w and h denote the width and height of a bounding box respectively. The width and height of a grasp correspondingly refer to the aperture size of the gripper and the length of a grip. This grasp representation has been the employed in prior work [17, 126, 127, 134, 136, 137] as guidance for robotic grippers.

In this work, robotic grasp detection is alternatively modeled by 2-D belief maps manually constructed from grasp rectangles. Having an N-finger robotic gripper, the grasp can be depicted by a mixture model of N bivariate normal distributions fitted around the finger locations.

For a parallel gripper, the common robotic grasp representation involving grasp rectangles [26] can be encoded in belief maps as follows. The centers of the gripper plates in 2-D Cartesian coordinates, correspond to the means $\boldsymbol{\mu}^{(n)} = (\mu_x^{(n)}, \mu_y^{(n)})^T$, with $n \in \{1, 2\}$, around

which the Gaussian distributions are centered. The Euclidean distance between the means $\|\mu^{(1)} - \mu^{(2)}\|_2 = w$ indicates the width of the grasp. Since the Gaussian distributions are elliptical with $\Sigma = \mathrm{diag}(\sigma_x^{(n)}, \sigma_y^{(n)})^2$, the primary axis of the ellipse corresponds to the grasp height h. The gripper angle θ can be utilised for construction of rotation matrix $R(\theta)$, which is further used for adjustment of the orientation of Gaussian kernels with respect to the object. The mixture model can be then defined as

$$G(\mathbf{p}) = \sum_{n=1}^{N} \frac{\exp\Big(-\frac{1}{2}\big(\mathbf{p} - \boldsymbol{\mu}^{(n)}\big)^T R(\theta)\, \Sigma^{-1}\, R(\theta)^T \big(\boldsymbol{p} - \boldsymbol{\mu}^{(n)}\big)\Big)}{\sqrt{2\pi N\, \sigma_x^{(n)} \sigma_y^{(n)}}}, \tag{4.1}$$

where $\mathbf{p}$ denotes a pixel's location inside the belief map. Figure 4.2 illustrates the representation of a grasp rectangle and the adapted grasp belief maps based on that.

While the same amount of information as the grasp rectangles is enclosed within the grasp belief maps, they express a non-parametric encoding of the inherent spatial uncertainty around a grasp location. The suggested grasp representation encourages the encoding of image structures, such that a rich image-dependent spatial model of grasp choices can be learned (σ is pre-defined and encoded in the maps, which are resilient to the exact choice of σ). Furthermore, the amplitude as well as variance of the predicted belief maps can act as a measure of confidence for the exact location and orientation of the grasp. Figure 4.3 manifests all the possible grasp configurations for an object using both representations of traditional bounding boxes and their adapted belief maps. Contrary to the rectangle representation, a model featuring grasp belief maps can express its uncertainty spatially in the map. Directly regressing the 2-D coordinates of a rectangle leaves no room for modeling this uncertainty. Specifically, as shown in human pose estimation literature [159], belief map regression improves accuracy compared to directly regressing the Cartesian coordinates of the points of interest as done by Toshev *et al.* [170]. The proposed mixture models can further be extended to the grasping representations of other types of grippers, such as hand prostheses. They can also be utilised in their present form for formulating grippers with higher dimensions, as in the work by Guo *et al.* [127].

In practice, heat maps are created by constructing Gaussian kernels according to Equation 4.1, guided by the centers and dimensions of the gripper fingers. The centers of the gripper sides are the means of the Gaussian kernels, σ_x is proportional to the gripper height and σ_y is a chosen constant value.

4.3.2 CNN Regression

Inspired by CNN-based object detection methods, many recent approaches in robotic grasp detection focus on predicting candidate bounding boxes in various locations of an image grid. A common design choice among deep learning methods for regressing confidence maps has been fully convolutional networks (FCNs) [171]. There are a wealth of fully convolutional networks offering exceptional performances in a variety of tasks [27, 120, 171–173]. The fully convolutional residual network (FCRN) proposed in [27] was exploited for the purpose of regressing a belief map in this work, since it indicated competitive performance for dense prediction tasks, specifically depth estimation, in real time. The FCRN not only offers state-of-the-art performance on depth map estimation from single RGB images, but also demands less training data and contains less parameters, which is critical for real-time implementations. Benefiting from this architecture, end-to-end model training is feasible, while eliminating subsequent post-processing steps. Thanks to the particular up-convolution layers of the FCRN, a high resolution output image can be generated.

Fully convolutional networks are typically built upon an encoder-decoder architecture. The encoder part of the FCRN is based on the ResNet-50 architecture [112], which embeds the input into a low dimensional latent representation. The decoder features custom residual up-convolutional blocks, which boost the spatial resolution of the latent representation up to half of the input resolution. The architecture of FCRN is manifested in Figure 4.4.

To approach the problem of robotic grasping through grasp belief map estimation, the network is trained to perform a mapping from a monocular RGB input to a single-channel heat map comprised of the Gaussian mixture representing the grasp belief. Selecting a single ground truth grasp creates an ambiguous problem due to the nature of grasping task, in which more than one grasp per object are typically valid. Therefore, in the single-grasp setup, the network is trained with the most stable available grasp, that is the one with the maximum grasping area. During training, the Euclidean norm between the predicted belief map $\tilde{G}$ and the chosen ground truth map G, defined in $\mathbb{R}^2$, is minimised as the objective function:

$$\mathcal{L}(\tilde{G}, G) = \|\tilde{G} - G\|_2^2. \tag{4.2}$$

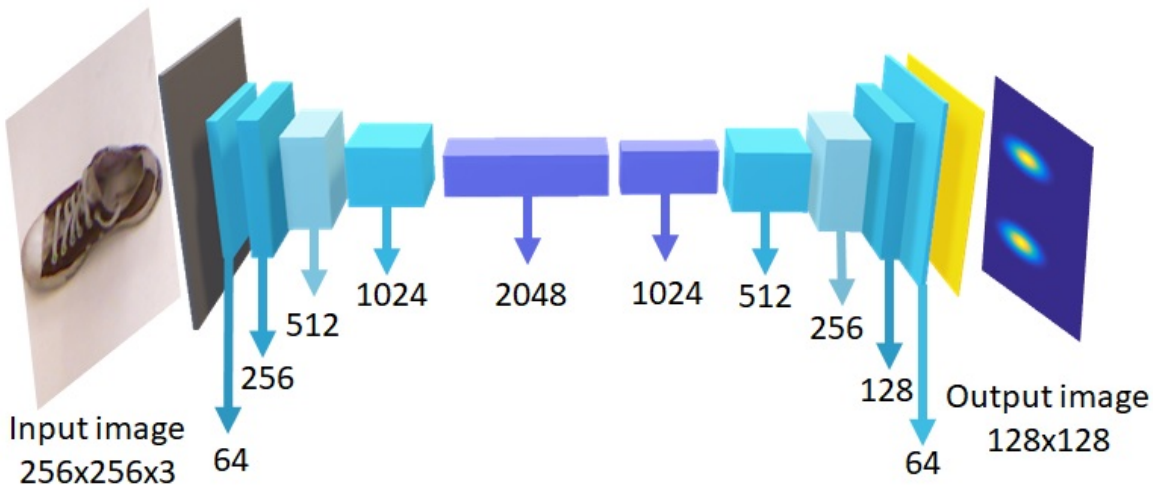


Figure 4.4: The architecture of the fully convolutional residual network used in this chapter.

4.3.3 Multiple Grasp Predictions

Training the model with a single viable grasp is not optimal and could degrade the generalisation capability of the network. That is, the model is penalised for predicting grasps which are potentially valid, but do not exactly match the ground truth. In other words, the samples that the model learns from, do not cover the entire grasp distribution. Thus, in the case of known objects, the model would overfit to the single grasp possibility it has seen, while in the case of previously unseen objects the uncertainty which arises would prevent the model from producing a sharp and reliable belief map.

To overcome this shortcoming, a multi-grasp estimation setup is developed. Rather than being forced to produce merely one output grasp, the model is allowed to produce multiple simultaneous outputs $\tilde{G} = \{\tilde{G}^{(m)}\}$, $m \in \{1, 2, ..., M\}$. In practice, the last layer is replicated M times. The goal is to then train the model such that it approximates the entire distribution of viable grasps. This problem can be formulated as an oracle meta-loss $\mathcal{M}$ that acts on top of the problem-specific objective function $\mathcal{L}$. By denoting the cost value of each grasp output as

$$\mathcal{L}_m = \mathcal{L}(\tilde{G}^{(m)}, G), \tag{4.3}$$

the meta-loss can be defined through the following minimum formulation:

$$\mathcal{M}(\tilde{G}, G) = (1 - \epsilon) \min_{m=1,...,M} \mathcal{L}_m + \frac{\epsilon}{M - 1} \sum_{m' \neq \arg\min_m \mathcal{L}_m} \mathcal{L}_{m'}. \tag{4.4}$$

At each training step, a grasp belief map is chosen randomly as the ground truth label among

all available ground truth possibilities for the given input sample. In this way, the entire grasp distribution for each sample will be seen during training. Since the model cannot know which ground truth belief map will be chosen for a specific image, it will learn to disentangle the possibilities into the M grasping hypotheses. This is achieved by the loss $\mathcal{M}$ in Equation 4.4. This objective is based on the hindsight loss, which only considers the output $\tilde{G}^{(m)}$ which is closest to the given ground truth G. Here it is formulated in a more intuitive way by using a soft approximation in which the oracle selects the best grasp with weight $1 - \epsilon$ and $\frac{\epsilon}{M-1}$ for all the other predictions, where $\epsilon = 0.05$. Thus, reduced gradients can be also returned for all other grasps. This is needed to enable output branches to be trained equally well, especially if they were initially not selected.

4.3.4 Grasp Option Ranking

The previously described model predicts M grasp hypotheses. For this system to be used in practice, a method for assessing the hypotheses quality and making a selection is required. Therefore it is desirable to find a way to rank all candidate grasps and pick one with a high probability of successful grasping. As the model is trained to produce two multivariate normal distributions, one way to rank the predicted belief maps is by fitting a two-component Gaussian mixture model to each output map using finite mixture model estimation [162].

The main parameters of a Gaussian mixture model are the mixture component weights ϕ_k and the component means μ_k and variances/covariances σ_k with K being the number of components. The mathematical description of a GMM distribution over all the components is

$$p(\mathbf{x}) = \sum_{k=1}^{K} \phi_k \mathcal{N}(\mathbf{x} \mid \mu_k, \sigma_k), \tag{4.5}$$

where $\mathcal{N}(\mathbf{x} \mid \mu, \sigma)$ represents a normal distribution with mean μ and variance σ and $\sum_{k=1}^{K} \phi_k = 1$. Mixture models can be typically estimated via the expectation maximisation (EM) algorithm [174], as finding the maximum likelihood analytically is intractable. That is, EM iteratively finds a numerical solution to the maximum likelihood estimation of the GMM. The EM algorithm follows two main steps: (E) computes an expectation of component assignments for each given data point given the current parameters and (M) computes a maximum likelihood estimation and subsequently updates the model parameters. The model iterates over E and M steps until the error is less than a desired threshold.

The same parametric model that was used to create the ground truth belief maps (Equation

Figure 4.5: A representation of a subset of the objects of the Cornell grasp detection dataset [17].

4.1) is fitted. The likelihood of the fit for each of the M predictions was then used for ranking and choosing the best fitted prediction as the system's final output.

4.4 Experiments and results

In this section, the suggested method was evaluated experimentally on a public benchmark dataset and compared to the state of the art. Further, the influence of the number of grasp hypotheses M on the performance of the method was investigated.

4.4.1 Dataset

The proposed approach was evaluated on the Cornell grasp detection dataset [17], which consists of 885 RGB-D images with a size of 640×480 pixels. The images come from 240 graspable objects including several grasping possibilities per object, annotated as rectangles that indicate various ways to grab an object. The dataset is mainly suited for 2-D grippers with parallel plates, but as the grasp size and location are included in the representation, it has the potential to be used also for other types of grippers as it is used in [127] for a 3-finger gripper. There are between 2 to 25 grasp options per object, representing a variety of scales, orientations and locations. Although presenting several "good" grasps per object, these annotated labels are not exhaustive and do not contain every possible grasp. Figure 4.5 shows some cropped samples of the dataset as used in this work. Here we only use the RGB images and disregard the depth maps.

4.4.2 Experimental Setup

In all the experiments the images and annotations were pre-processed as detailed in the following before being fed into the CNN. The images contain a large margin of background around the objects, thus the images and their corresponding grasp maps were cropped semi-automatically to 350×350 pixels and then the images were bilinearly down-sampled to 256×256 and the grasp maps to 128×128. The semi-automatic cropping involved finding a window in which the object was wholly contained.

For training several data augmentation techniques were employed. A random rotation in $[-60°, 60°]$, a translation from $[-20, 20]$ pixels and scaling between 0.9 and 1.1 were sampled. Each image was augmented six times. Thus, the final dataset contained 5310 images after the augmentation procedure. All the images and labels were normalised to a range of $[0, 255]$.

To train the single grasp prediction model, the largest ground truth grasp rectangle was opted as the label since area is a good indicator for probability and stability of the grasp. This selection may be trivial, but training a single grasp prediction model is not feasible without pre-selection of a fixed ground truth among the several annotations provided by the dataset. The reason is that training the single prediction platform with a randomly selected grasp at each iteration would confuse the network such that learning a proper model is not achievable.

On the other hand, the multiple grasp prediction model can deal with a *variable* number of ground truth grasp maps per image. The performance of the implemented multiple grasp prediction framework for different numbers M of grasp hypotheses was investigated and reported. One observation that arises is that the model with $M = 5$ shows significant improvement in performance, compared to the single-grasp model, while the average number of grasps per object in the dataset is also approximately five. To illustrate the effect of the number of grasping options, the system was also evaluated with other M values, which is reported in the following.

Training was performed via an NVIDIA Titan Xp GPU using the $MatConvNet$ [175] toolbox. The learning rate was set to 0.0005 in all experiments. For regularisation weight decay was set to 0.0005 and a dropout layer with rate equal to 0.5 was added. The models were trained using stochastic gradient descent with momentum of 0.9 for 50 epochs and a batch size of 5 and 20 for training multiple and single prediction models respectively.

4.4.3 Cross-validation Sets

In previous works [17, 126, 127, 134, 136, 137] two specific cross-validation splits were evaluated: image-wise and object-wise. The former split involves training of the model with all

objects, while some views of the objects remain unseen. This evaluates the within-object generalisation capability of the network. However, even an over-fitted model could perform relatively well on this split. The object-wise split involves training the network on all available views of the same object and testing it on new objects and thus is suitable for evaluating the network's competence on unseen objects. Interestingly, some novel objects are rather similar to ones used in training. These two splits represent the within-object cross-validation and between-object cross-validation settings exploited in the previous chapter and therefore to be persistent with the terms same naming as previous chapters is followed in the remaining.

It is worth noting that none of the previous work scrutinised the network's potential in detection of novel *shapes*, in spite of the dataset including a variety of similar objects. For example, there are several objects with different colors but of the same shape. Therefore, the BOC setting may not be a good measure for generalisation to novel shapes. To investigate the framework's performance on unseen shapes, an additional *shape-wise* split was created to encourage larger variation in objects between the train and test sets. To follow similar naming as the other cross-validations, it is named between-shape cross-validation (BSC). The train and test folds were picked such that all the objects of similar shapes, for example various kinds of hats, are included in one of the test/train folds only and therefore novel when testing. Both WOC and BOC settings were validated in five folds. Two-fold cross validation for the BSC setting was performed, where the first 20% of objects were used for testing and the remainder were specified for training. The second fold used the same split but with reversed order of objects.

4.4.4 Grasp Detection Metric

To have a fair comparsion, a quantitative performance using the rectangle metric that was suggested in [26] is reported. A grasp is counted as a valid one only when it fulfills two conditions:

- The intersection over union (IoU) score between the ground truth bounding box (B) and the predicted bounding box (B^*) is above 25%, where

$$\text{IoU} = \frac{B \cap B^*}{B \cup B^*}. \tag{4.6}$$

- The grasp orientation of the predicted grasp rectangle is within 30° of that of the ground truth rectangle.

This metric requires the prediction to be in the form of a grasp rectangle, while the designed

network predicts grasp belief maps. To provide comparable representations for prediction and ground truth, the modes μ_1 and μ_2 of each elliptical Gaussian of the belief maps are calculated. The Euclidean distance between these modes should be equal to the width of grasp rectangle (Figure 4.2). The height of the Gaussians is then computed by detecting the range of each belief map's contour. The orientation can also be determined by calculating the angle of the main axis of each belief map. Having these values, a rectangle of the same height between the measured centers can be reconstructed, which is rotated according to the calculated orientation. In case of deformed grasp maps, *e. g.* under high uncertainty, a rectangle cannot be extracted. It was noted that a valid grasp meets the aforementioned conditions with respect to *any* of the ground truth rectangles. The percentage of valid grasps as the *Grasp Estimation Accuracy* can be calculated accordingly.

4.4.5 Evaluation and Comparisons

In the following, the multiple grasp prediction method was compared with the single-grasp baseline and state-of-the-art methods. As there are several ground truth annotations per object, the selected prediction was compared to all the ground truth grasp rectangles to find the closest match. Among the predictions there can be some which are not viable, while others are perfect matches. The selected prediction for each image is the top-ranked prediction after the GMM-based scoring. The computed accuracy in Table 4.1 considers the success of this prediction in providing a valid grasp.

A full comparison of the results is included in Table 4.1, where M indicates the number of hypotheses and consequently choosing $M = 1$ refers to the regression of single belief map and can be seen as a baseline in the following experiments.

Having only an RGB image as the input of the model, the implemented multiple grasp prediction models outperformed all state-of-the-art approaches that use additional depth information, except for Guo *et al.* [127] who also leveraged tactile data. The difference to the implemented single grasp baseline is significant and reveals the potential of modeling ambiguity in robotic grasp detection.

It is worth noting that the comparable performance of the models in the WOC and BOC settings (also in prior work) suggests that task difficulty does not change much between the two scenarios. Through the more challenging BSC scenario a better measure in dealing with novel objects is provided. In this case, the accuracy of the single grasp baseline does indeed deteriorate. The implemented multiple grasp model, however, is still able to handle the in-

Table 4.1: Comparison of the proposed method with the state of the art.

Method	Input	Grasp Estimation Accuracy (%) WOC	BOC	BSC
Jiang [26]	RGB-D	60.5	58.3	-
Lenz [17]	RGB-D	73.9	75.6	-
Wang [136]	RGB-D	85.3	-	-
Redmon [126]	RGB-D	88.0	87.1	-
Asif [134]	RGB-D	88.2	87.5	-
Kumra [137]	RGB-D	89.2	89.0	-
Guo [127]	RGB-D, tactile	**93.2**	89.1	-
Kumra [137]	RGB	88.8	87.7	-
Single-grasp ($M = 1$)	RGB	83.3	81.0	73.7
Proposed ($M = 5$)	RGB	91.1	**90.6**	85.3
Proposed ($M = 10$)	RGB	91.5	90.1	**86.2**

creased difficulty with a large performance boost over the baseline. It can be observed that with an increasing number of grasp hypotheses the performance gap of the multiple-grasp over the single-grasp model becomes the highest for the BSC split, with over 10% increase in accuracy and robustness and generalisability to unseen shapes/objects. Last but not least, both single and multiple grasp models had a faster run-time than the state of the art at 56 milliseconds. Increasing the number of hypotheses did not have a negative effect on speed.

Figure 4.6 illustrates qualitative examples from the multi-grasp framework (with $M = 5$) and a comparison to the single grasp ($M = 1$) model's predictions. The figure indicates the advantage of multiple grasp predictions to a single prediction in terms of both accuracy and variability. It can be observed that for objects that have several distinct grasping options, the adapted multiple prediction framework models the output distribution sufficiently. Object 3 (scissors) is undoubtedly a challenging object with many different grasping poses, which are successfully estimated via multiple predictions.

4.4.6 Evaluating Multiple Grasps

Table 4.2 reports the average grasp detection accuracy of lower and upper limit predictions of the multi-grasp models. To achieve the lower limit of the model's performance, *all* predictions provided by the implemented model were evaluated instead of evaluating only the top-ranked grasp hypothesis. As this evaluation computes the success rate of all hypotheses, including even those with a low probability of being chosen, it is called the lower limit. The comparison

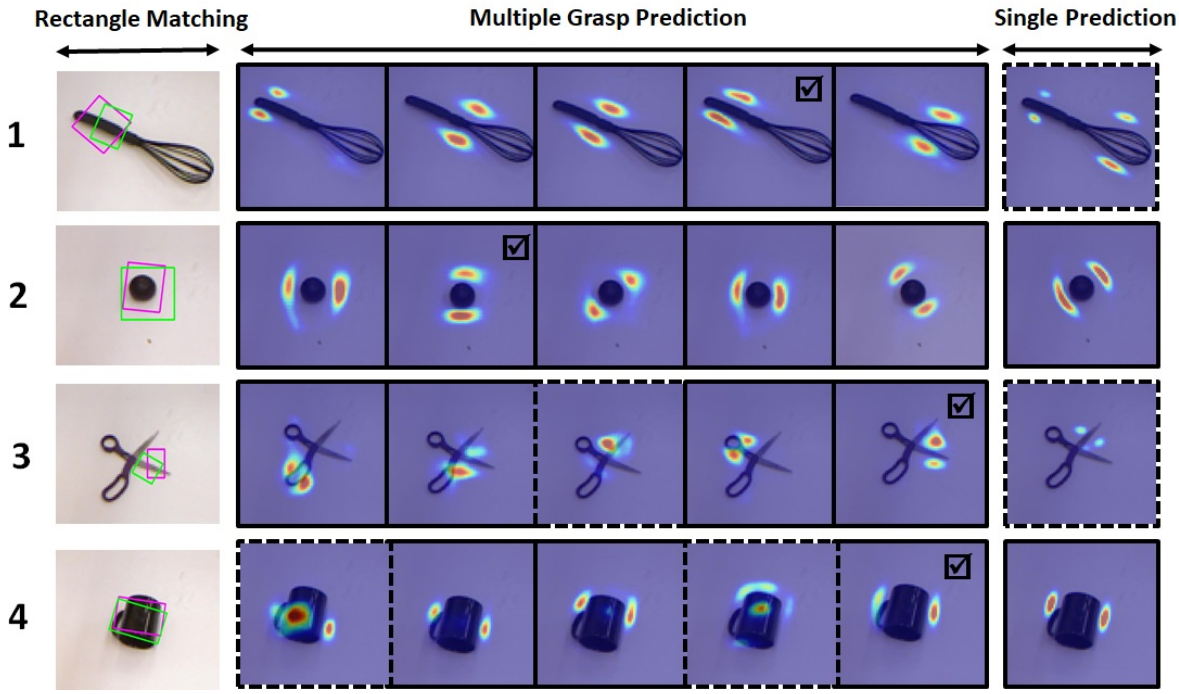


Figure 4.6: Five and single grasp map predictions of sample objects in the dataset. A solid frame around an image is an indicator of grasp detection success, while a dashed line shows an incorrect detection. The images with the ✓ are the top-ranked predictions picked by the designed GMM likelihood estimation module. These predictions are converted back to grasp rectangles (shown in Magenta) and compared with Green rectangles indicating ground truth grasps.

of these results with multiple prediction results in Table 4.1 indicates that the estimated belief maps correspond, in most cases, to valid grasps. This lower bound decreases as M increases, that is it is more likely to have a (noisy) prediction that does not match any of ground truth grasp rectangles with higher M. However, thresholding the "good" matches based on the GMM-fitted ranking can counteract this drop in performance while leaving multiple grasping choices to the robot.

Another remark is that the top-ranked prediction is not necessarily the best one in terms of grasping performance. This can be observed by investigation of upper limit evaluation, in which if there exists at least one matching grasp detection among all hypotheses, it counts overall as successful. For $M = 10$ the upper limit exceeds 98% accuracy for the BOC split. This implies that there is in almost all cases at least one valid prediction returned by the model, although GMM fitting might not always result in correct ranking. Still, the top-ranked prediction performance in Table 4.1 is closer to the upper rather than the lower limit.

Table 4.2: Average grasp estimation accuracy of all hypotheses (lower limit) and average grasp success (upper limit).

Method	WOC	BOC	BSC
lower limit ($M = 5$)	80.0	77.4	75.0
lower limit ($M = 10$)	76.5	73.3	72.1
upper limit ($M = 5$)	98.0	98.5	96.3
upper limit ($M = 10$)	99.2	98.4	99.1

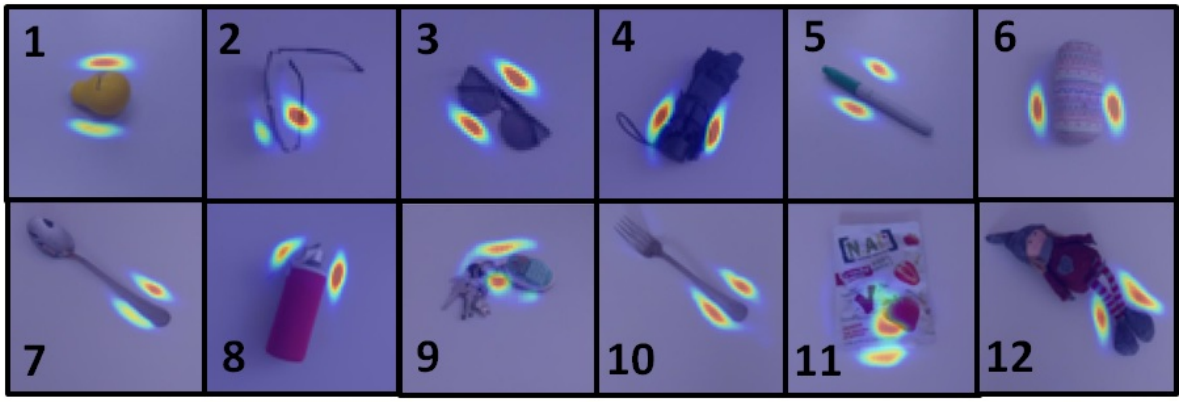


Figure 4.7: The top-ranked grasp map picked by the GMM likelihood estimation module for a $M = 5$ model evaluated on common household objects in real-time. Objects 1-5 have similar shapes to the objects in the Cornell grasp dataset. Objects 6-12, however, represent novel shapes and textures compared to the dataset used for training. Despite variations from the training distribution, the proposed method produces reasonable grasp maps for all tested objects.

4.4.7 Generalisation

Finally, the performance of the proposed model was evaluated in a real-world scenario on several common household objects, such as cutlery, keys and dolls, in an own setup. Contrary to offline test images that come from the same distribution as the training dataset, this setup introduces novel object shapes not included in the Cornell dataset and variations to the test images including camera view and illumination condition changes. Therefore, this setup can provide a measure of the generalisation capability of the proposed model under different conditions and challenging novel shapes and textures. Figure 4.7 depicts the evaluated objects with their top-ranked predicted grasp map that is opted by the GMM likelihood. The suggested model is adequately robust against the introduced variations in this test and generates viable and confident grasping options for all tested objects.

4.5 Discussion

In this section, different aspects of the designed system using the achieved results are investigated. According to Table 4.1, the results show better performance of the developed system compared to other systems focused on robotic grasping. As for the purpose of this thesis, the benefit of the designed system for prosthesis grasp estimation is also observed.

4.5.1 Prediction Considering the Ambiguity of the Task

According to the Section 4.4, estimation of five grasp maps provides 7.8%, 9.6% and 11.6% higher average accuracy for WOC, BOC and BSC splits respectively compared to single grasp prediction case. As already mentioned, this difference gets higher with increase in the task complexity, where the BSC split represents the most challenging case. These observations indicate that in the presence of ambiguity, the multiple grasp map prediction structure can more conveniently deal with this ambiguity of grasp detection task and provide desirable results where the single grasp prediction system fails. Hence, the ambiguity of the task of grasping which was neglected in previous chapters seems to be a crucial factor. It should be noted that boosting the number of possible predictions may not necessarily lead to better performance as there should be a trade-off between the number of hypotheses and accuracy. The desired number of possible hypotheses was examined here by trying two possibilities: $M = 5$ and $M = 10$. The latter however led to negligible difference and therefore $M = 5$ was selected to prevent further complication of the model.

4.5.2 Comparison of Cross-validation Results

The average accuracy within different cross-validation settings indicate reasonable results for each. That is, similar to previous chapters, the WOC split represents the least challenging setting as each object is already seen once by the model. The BOC split includes more novelty in appearance of test set and therefore the model is involved with more complexity. In previous chapters however the difference between the average accuracy of these two tasks was more distinguishable. The reason can be the repetitive objects appeared in the Cornell grasp dataset, in which there are several instances of each object category. For a better illustration, in the ALOI dataset there was hardly any object of distinct categories with similar shapes, e.g. two mugs with different colors but the same shapes. Nonetheless, this is the case for the Cornell dataset, which reduces the distinction between the WOC and BOC splits.

As a solution to the mentioned issue, a shape-wise split was introduced to both distinguish the tasks with higher emphasis and evaluate the system in presence of unknown shapes.

4.5.3 Single Prediction Using Random Ground Truth

In the single-grasp setting, the grasp with largest area was opted as the ground truth. An interesting investigation is to compare the baseline of a single grasp prediction with randomly selected ground truth grasps to selecting the grasp with the largest area. This test reduces prediction accuracy from 82% to 68% in one of the object-wise split folds. Fig. 4.8 shows example visualisations of these results.

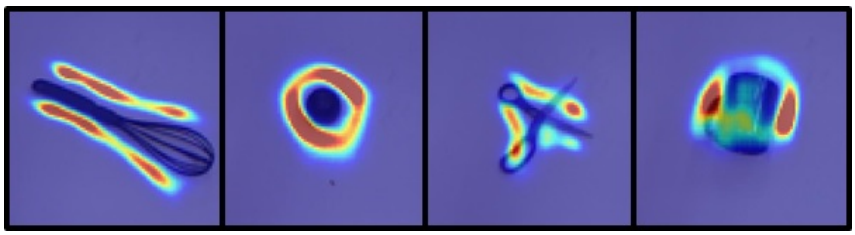


Figure 4.8: Illustration of the random grasp selection baseline, the model blurs all viable grasps to an undefined heat map. These results can be directly compared to Fig. 4.6.

4.5.4 Skip Connections

The usage of long-range skip connections has shown performance improvement for various models including fully convolutional networks [172]. These connections let higher frequency information flow from initial layers to the output layers. The performance of the network therefore was investigated with addition of skip connections. As a result, the grasp detection accuracy dropped by 3%. An explanation for such observation could be the fact that the belief maps do not benefit from finer details in the color image, as it would be the case for example for semantic segmentation.

4.5.5 Utilisation of Grasp Maps for Prosthesis Grasping

Considering the aim of this thesis for prosthesis grasp prediction, it is open to investigation whether the predicted grasp maps can contribute additional information to grip pattern identification task. The Gaussian belief maps utilised for grasp representation contain information about the object height, width, rotation, translation and therefore general object shape and pose. Furthermore, these maps represent the approximate area that the object can be grasped and

therefore can derive the model's attention to the grasp area. These features seem plausible for usage of this system for prosthesis grasping. Yet, this possibility requires further investigation.

4.6 Conclusion

This chapter proposes a novel method for robotic grasp detection by dealing with ambiguity of having multiple viable grasp options. To this end, the representation of a grasp is redefined from an oriented rectangle to a 2-D Gaussian mixture belief map such that the inherent ambiguity of the grasping task is accounted for. A fully convolutional network is trained to predict the grasp representations of a single RGB input image. To better model the high ambiguity of grasping task stemming from the many possible ways to grasp an object, a multiple grasp prediction framework is included in the training procedure. Increasing the number of possible grasp belief map predictions led to a significant improvement in grasp prediction in comparison with a single-grasp baseline. Such enhancement is especially more noticeable in scenarios involving challenging tasks, such as novel objects, shapes and textures.

The algorithm is highly efficient and provides state-of-the-art performance for real-time implementations. A GMM-based ranking approach was also suggested for opting the best grasp prediction through estimating the hypothesis with the highest likelihood. The top-ranked grasp can be utilised in a real-time system.

Since the suggested grasp belief maps contain not only information about the confidence of a potential grasp position, but a measure of grasp size and orientation, the application of this method to a vision-based prosthesis grasp recognition framework can be interesting.

Chapter 5

Grip Pattern Classification for Prosthetic Hands using Estimated Grasp and Depth Maps

This chapter offers several techniques for improvement of grasp recognition methods presented in previous chapters. Firstly, the variation in distance from a monocular camera can degrade the performance of a grasp recognition system. To resolve this issue, a simultaneous system for depth and grasp estimation is suggested to provide more robustness to this distance variation. In addition, a variety of structures are examined to predict a grip pattern based on a single RGB input image and its depth and grasp map estimations. The implemented designs are analysed and the best one is picked for the final implementation of the grasp estimation system.

5.1 Motivation

The main motivation of this chapter is to build upon the proposed structures in chapters 5 and 6 and create a unified platform that performs grasp type estimation effectively. The proposed system in Chapter 5 may suffer from lack of robustness to variety in distance of object from camera. This issue can be overcome by having a depth sensor. Nevertheless, in this thesis the addition of hardware to the prosthesis was avoided. An alternative approach to depth sensing can be depth estimation, especially considering the fact that the Cornell grasp dataset offers RGB-D data, of which the depth information can be used as the ground truth labels. Since an FCRN structure is already implemented in Chapter 6 to estimate grasp belief maps, an efficient solution can be concurrent depth and grasp map prediction by adapting the structure to a multi-

task learning one.

Observing the promising performance of deep networks boosted the expectations from these methods for doing more tasks at the time, *e. g.* in natural language processing [176] or computer vision [122]. Multi-task learning (MTL) can involve several tasks, such as joint learning, learning to learn and learning with auxiliary tasks [177]. In general, having more than one objective in a learning problem refers to MTL [177]. MTL has shown to be beneficial and improve the overall performance for many tasks [177, 178]. Therefore, the possibility of learning depth and grasp maps jointly is investigated in this chapter.

Focusing on the task of grasping for a hand prosthetic, the depth and grasp map estimations of an object's snapshot are still not suitable inputs for a prosthetic hand. That is, a final step is required for predicting a grip pattern based on the available data. Therefore, having an RGB image of a common object, a grasp and depth map can be estimated, which can be further processed to provide a grasp class. The best way to achieve this classification is investigated further during the following sections.

5.2 Methods

5.2.1 Simultaneous Depth and Grasp Map Estimation

The FCRN platform implemented in Chapter 6 features a fully convolutional structure capable of an image-to-image correspondence between an input image and an output map, *e. g.* 2-D grasp belief maps. This structure can be adapted for performing other supervised tasks (classification or regression) at the same time. Specifically, having in mind that the original implementation was designed for accurate depth estimation [27], depth information can simultaneously be regressed with grasp maps using a distinct criteria. To further illustrate the arrangement of such structure for simultaneous prediction of different tasks, the proposed architecture is depicted in Figure 5.1. Both estimations utilise the same weights until the very last branch of fully convolutional architecture in which the prediction happens.

To proceed with such approach, depth images of the RGB input data are required. One attribute of Cornell grasp dataset 4.4.1 is the presence of RGB-D information for every object. Hence, the depth information can be used as the ground truth for the depth estimation branch.

Grasp estimation is performed in a similar procedure as proposed in the previous chapter. A multiple hypothesis prediction criteria optimises regression error of estimated grasp maps using least squares error (LSE) or $\mathcal{L}_2$ loss (Equations 4.3 and 4.4). The depth prediction however

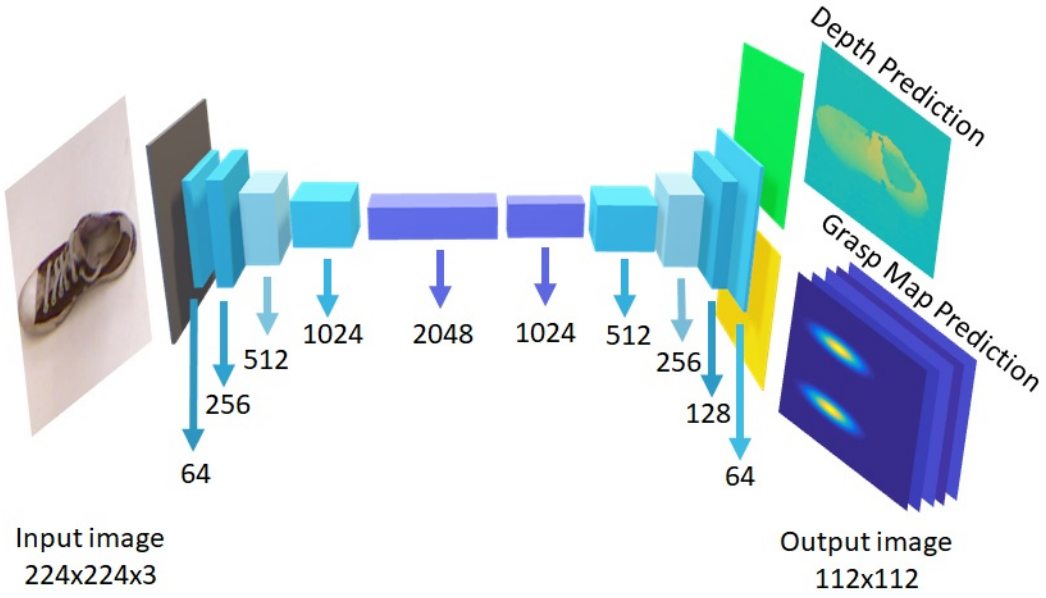


Figure 5.1: The multitask learning architecture implemented for concurrent grasp map and depth prediction. The depth prediction branch includes a single channel including scene depth information. The other prediction channel however produces 5 grasp belief maps.

involves only one possibility and therefore there is no ambiguity to deal with. Accordingly, the Euclidean norm between the estimated depth map and the ground truth depth map is the objective to be minimised according to Equation 4.2.

5.2.2 Grip Pattern Classification

Although grasp maps provide details of a grasp act such as grasp location, size and orientation, they are not indicative of a grasp pattern. Therefore, an approach for converting the estimated grasp belief maps to a grasp is required. Such approach can be employed within a semi-autonomous system presented in Figure 3.12 for grasp suggestion in myoelectric hands.

Numerous methods can be designed to attain a grasp proposal based on the available data: input RGB image, five predicted grasp maps and depth prediction of the scene. In the following different architectures for combining different sources of data are elaborated. Please note that the number of output grasp categories in this chapter are limited to 3 classes, which will be explained further in Section 5.3.1. For a real-time implementation, the image preprocessing and CNN modules presented in Figure 3.12 A can be simply substituted with one of the following architectures.

1. **Simple classification of input image or grasp belief maps** The very first option for grasp classification is to feed the input RGB image directly into a pre-trained ResNet-50 architecture to produce 3 grasp classes. For such task, a similar approach as Section 3.6 can be

performed while the main difference is replacing the 1000 classes specified for ImageNet with 3 classes for the required grip patterns for Cornell grasp dataset. The objective to be optimised here is the *cross-entropy* loss as illustrated previously in Equation A.17. This architecture is depicted in the first path of Figure 5.2.

Another option for direct classification is to directly classify the grasp belief maps. It would be interesting to observe how much the grasp maps can contribute to the classification tasks as their appearance seem to contain less information than the RGB images.

To benefit from the ImageNet rich features rather than using random weights, a pre-trained ResNet-50 can be fine-tuned. Nonetheless, the ImageNet images are RGB and therefore consisting of only 3 channels of data. In order to enable feature extraction out of 5 grasp belief maps predicted by the FCRN, the first layer of the ResNet-50 should be adapted accordingly. Consequently, the first layer of the pre-trained network, a convolution layer of size $7\times7\times3\times64$, is removed and substituted $7\times7\times5\times64$ convolution filters with randomly initialised weights. The rest of the network is designed in the similar way as the previous architecture. Figure 5.2 branch 2 illustrates the details of the explained approach.

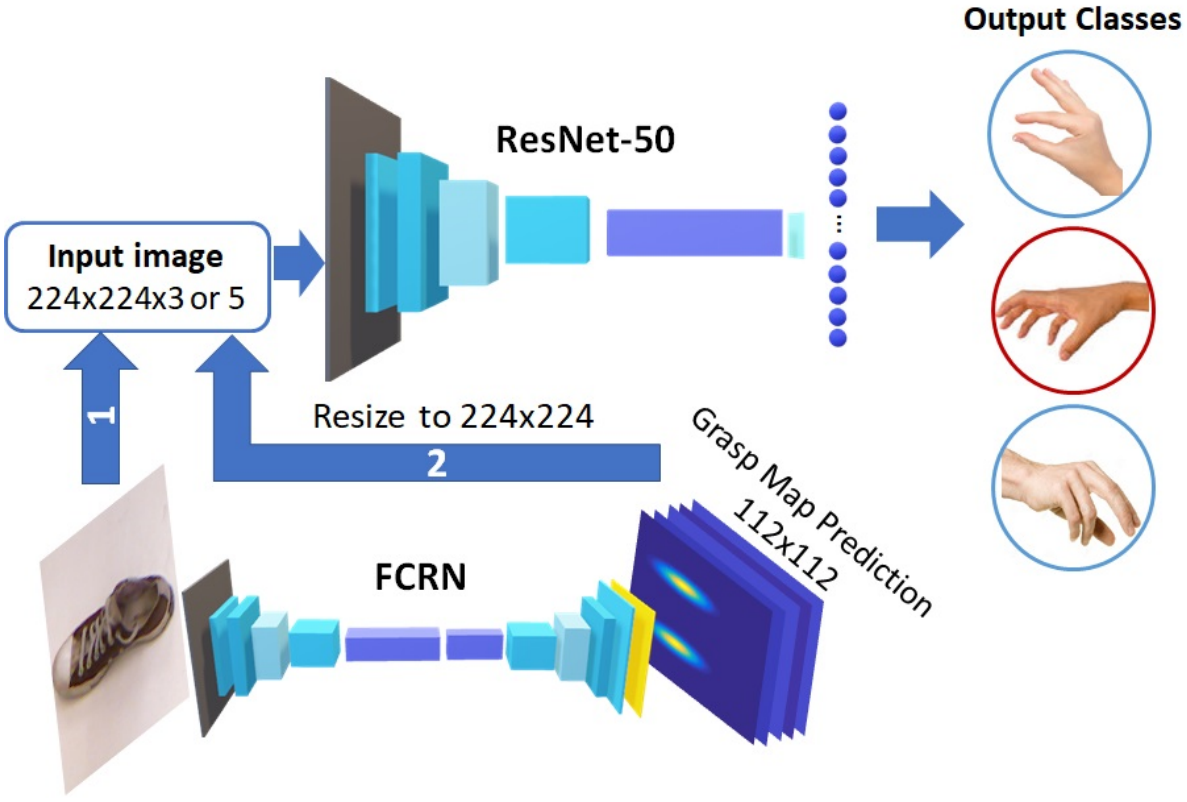


Figure 5.2: The procedure for classification of 1) RGB images (A 1-1) or 2) five grasp maps (A 1-2) into 3 grasp classes. 1 and 2 paths represent two different implementation options and are not present at the same time.

2. **Multi-task network for depth map, grasp map and grasp classification** Another approach to be examined is to investigate the possibility of simultaneous grasp map, depth map and grasp type classification. The first two tasks can be regressed as done in Section 5.2.1. The third task however requires the minimisation of a *cross-entropy* loss over the latent features of the fully connected recurrent network. These features can be produced using a ResNet-50 network and therefore are represented before addition of up-convolution layers.

   Figure 5.3 depicts the implemented architecture for performing the mentioned task through a single network in an end-to-end manner. It is open to investigation whether learning depth and grasp map relevant data elevates the network's performance in grasp classification or not.

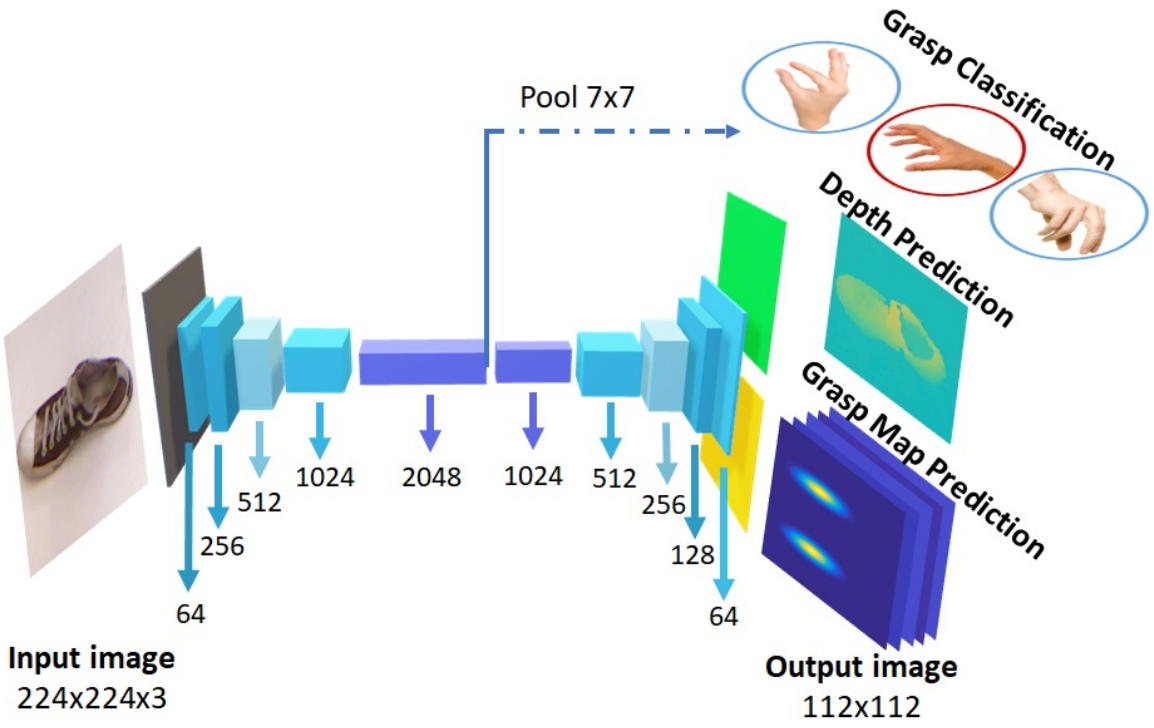


Figure 5.3: The multi-task learning platform for the three task of grasp classification, depth and grasp map estimation (A 2). The grasp classification task requires a pooling layer to convert the latent space of the pre-trained ResNet-50 (2048 features) to 3 classes.

3. **Grasp classification by feeding available data into a single network** A straightforward approach for grasp classification using all the available data is to concatenate and feed them all into the pre-trained ResNet-50. The available data consists of 3 channel RGB image, 5 channels of different grasp heat map predictions and a single channel depth map resulting in 9 channels in total. In order to have a nine-channel input, the first layer of the ResNet-50 should be replaced with $7 \times 7 \times 9 \times 64$ convolutions initialised randomly

similar A 1 branch 2. The network can then be trained the same way as a simple classifier. The details of the adapted network are shown in Figure 5.4.

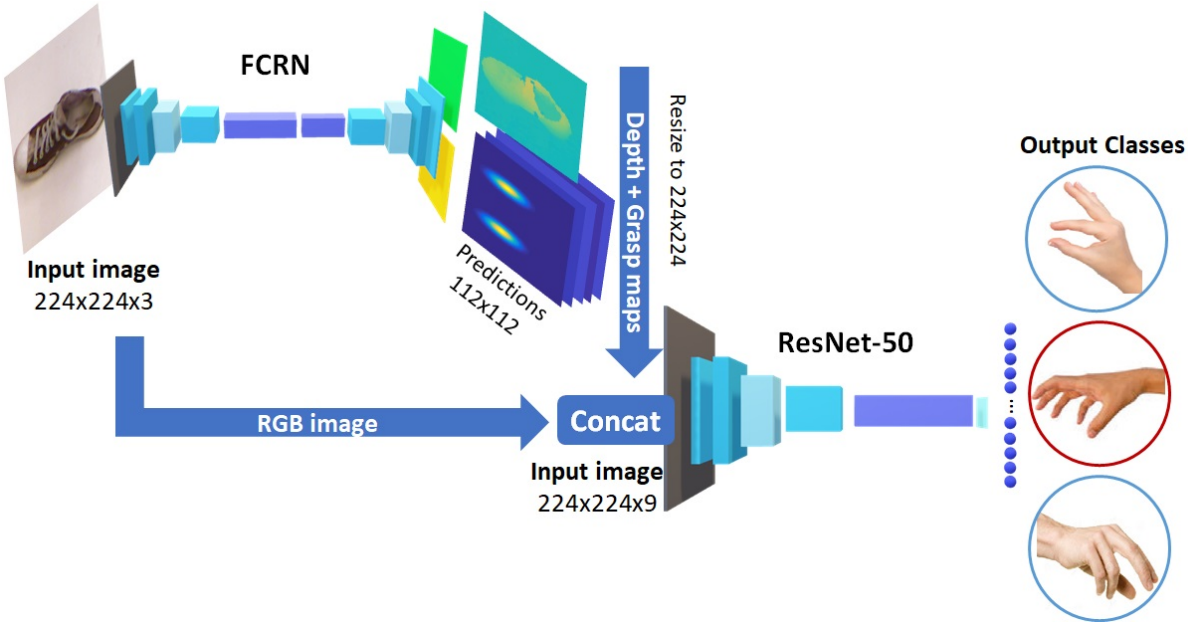


Figure 5.4: The designed platform for grasp classification of images of Cornell dataset based on the estimated grasp and depth maps in combination with the original RGB images (A 3). The estimated maps should be resized to 224×224 to be concatenated along their third channels.

4. **Grasp classification based on a combination map of RGB image and corresponding grasp maps** To benefit from both RGB and grasp belief maps' data while keeping the network as simple as possible, the RGB images can be summed with the grasp belief maps in a weighted manner. Equation 5.1 indicates how the combined map is created from the RGB image (I) and estimated grasp belief maps ($\tilde{G}^{(m)}$) for $m \in \{1, 2, ..., M\}$ where M is the number of grasp map predictions (5) and w is the weight chosen empirically. This equation basically demonstrates a simple method for combining information from two images by summing over the pixel values of each.

$$CombinedMap = w \times \tilde{G}^{(m)} + (1 - w) \times I. \tag{5.1}$$

The architecture for grasp classification based on a combined map is depicted in Figure 5.5.The grasp maps are resized to 224×224 so that they can be added to original RGB images. As RGB images include only 3 channels, 3 heat maps are picked randomly for map construction.

5. **Grasp classification based on log-likelihood estimation**

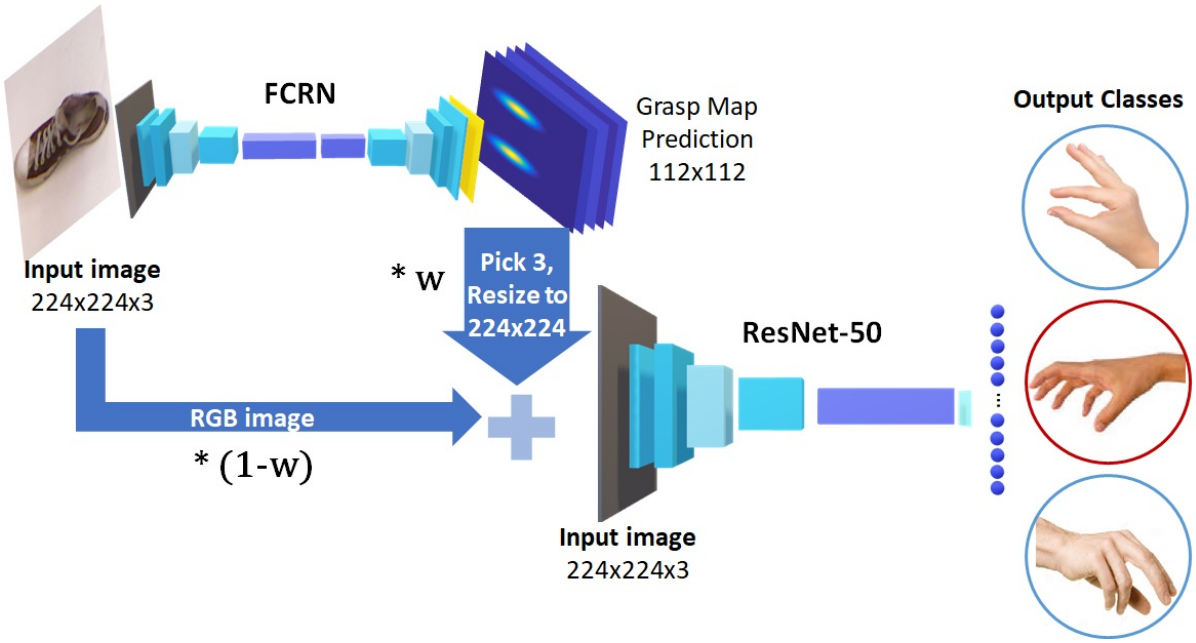


Figure 5.5: The network design for producing a combined map of RGB and grasp map data and feeding that to a pre-trained ResNet-50 for grasp suggestion (A 4).

The output of a ResNet after the last pooling layer are values that can be considered as class likelihoods given the input image. In the previous chapter, Gaussian mixture models were exploited to calculate the log-likelihood of estimated grasp maps. The log-likelihood of the grasp maps can be estimated and concatenated with the likelihood values of each grasp class given the RGB data to be fed into a *voter* network for further decision making based on all the values. The concatenated values should be adapted to be in the same range. Further details of the proposed method can be found in Figure 5.6.

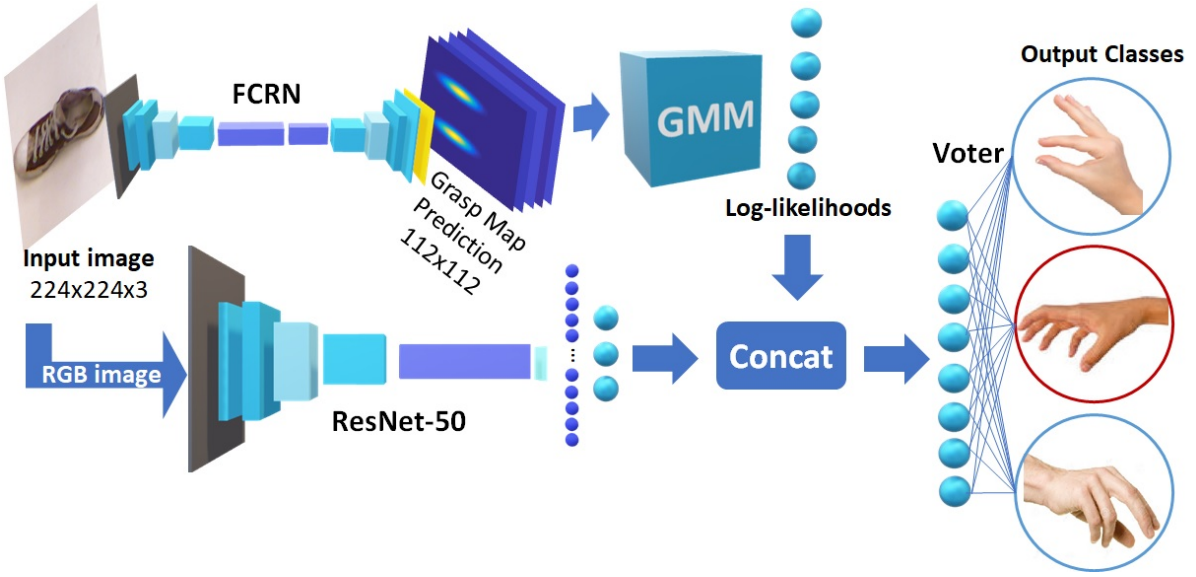


Figure 5.6: The designed architecture for production of log-likelihood values from the grasp maps fed into a voter net together with image features based on grasp classes extracted by a ResNet-50 (A 5).

6. **Grasp classification through parallel feature extraction** Finally, an approach to keep features extracted from the RGB image and its corresponding grasp maps separated while predicting grasp classes jointly is to exploit a parallel architecture. That is, an RGB image can be given to a pre-trained ResNet-50, while 3 or more grasp maps can be fed into an identical ResNet-50 for classifying 3 grasp types jointly. The weights of each ResNet are updated separately and after training, predictions of each one are given to a shallow voter net for decision making. A detailed overview of the architecture is demonstrated in Figure 5.7. The difference between this architecture and a siamese [179] one is that weight updating is done separately for each branch as the inputs are from different domains in contrast to a siamese net. Another difference is the loss, which is the contrastive loss based on distance of output maps [179]. Here, on the other hand, a normal cross-entropy loss is used for classification of 3 grasp types.

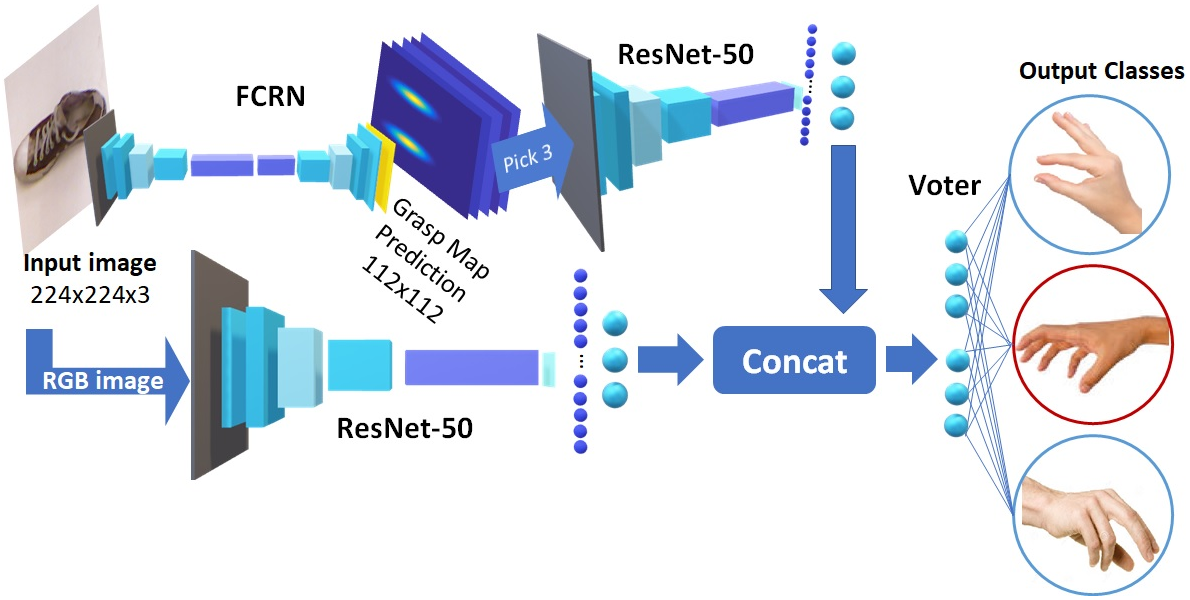


Figure 5.7: The parallel network design for grasp classification from multiple resources of information (A 6). The produced grasp maps from the FCRN are frozen to be fed simultaneously with RGB data into the parallel net.

In the next section, these architectures are evaluated and compared based on their performance, complexity and processing time.

5.3 Experiments and Results

5.3.1 Dataset

The experiments in this chapter were carried out on Cornell grasp dataset (Section 4.4.1) similar to the previous chapter. The main reasons for this decision are that the dataset covers a variety of common objects (885 images in total), includes robotic grasp annotations required for grasp belief map generation and includes RGB-D information. Another merit of the Cornell grasp dataset is that the objects are already annotated for the human grasping task by DeGol *et al.* [180]. The objects were annotated based on 5 different grip patterns, namely, *Key, Pinch, Power, Three Jaw Chunk* and *Tool*.

An issue about these annotations can be the high data imbalance in grasp classes. For instance, there is no objects of *Key* grasp in this dataset. Furthermore, the distinction between the *Tool* grasp, which contributes to only $\sim 3\%$ of objects in the dataset, and *Power* grasp is not highlighted. To resolve these issues, *Key* grasp was removed from the possible grasp groups and the objects annotated as the *Tool* grasp were labeled as *Power*. In this way, there are 3 main grasp classes for this dataset: *Pinch, Power, Three Jaw Chunk* or *Tripod*.

5.3.2 Cross-validation Sets

The validation setting in this chapter follows similar procedure as Chapter 5 including the same splits of BOC and WOC. The reason behind this decision is that in Chapter 5 there was a grasp classification framework in which only BOC and WOC settings were used. In this way, the achieved results are comparable to the ones achieved in that chapter.

It is worth noting that although WOC and BOC settings were already used for Cornell dataset in Chapter 6, this chapter utilises different sets of random objects for these settings, while the overall procedure of each validation setting remains exactly the same.

5.3.3 Implementation Details

All the experiments in this chapter were performed via an NVIDIA Titan Xp GPU. The $MatConvNet$ [175] deep learning toolbox was utilised to implement all architectures.

5.3.4 Simultaneous Depth and Grasp Map Estimation

Experimental Setup

For the concurrent depth and grasp estimation platform, similar data processing steps as Chapter 6 were followed. That is, input RGB images, grasp and depth maps were centrally cropped by a 320×320 pixels window. The input RGB images were resized to 224×224 pixels and the depth and grasp maps were resized to 112×112 pixels.

Similar to previous chapter, a variety of augmentation techniques were applied to images: random rotations in range of $[-60°, 60°]$, translations from $[-20, 20]$ pixels, scaling between 0.9 and 1.1 and illumination modification. Each image was augmented 5 times leading to a sum of 4250 images, which were all normalised to a range of $[0, 255]$.

The multi-task network was trained with Adam optimiser with the learning rate of 0.001. Overfitting was attempted to be prevented using regularisation weight decay of 0.0005 and a dropout layer with 0.5 rate. In the multi-task setting the model was trained for 60 epochs rather than 50 epochs used in previous chapter to make sure the network is properly trained as the task seem to be more challenging. A batch size of 5 was used for training.

The objectives for the two tasks of multiple grasp and depth estimation belong to different distributions and therefore present different output ranges. To balance the overall objectives, a weighted combination of both should be found such that both tasks are trained properly. To do so, the weight of 1 and 0.5 were respectively chosen for the multiple grasp map and depth losses. The weight specification was done by trial and error and observation of gradients relevant to each objective to analyse the contribution of each task to the overall loss.

Evaluation Metric

The evaluation of multiple grasp maps are done the same way as the previous chapter, namely IoU and orientation measurement. For depth estimation however two main metrics are used for error calculation: root mean squared error (RMSE) and mean absolute relative error (MARE). The former produces the error of depth estimation in the same unit as the depth values (m or cm) as shown in Equation 5.2.

$$M = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tilde{D}_i - D_i)^2} \tag{5.2}$$

The MARE contrarily measures the relative error (Equation 5.3), which is between 0 and 1 and therefore shows error percentage when multiplied by 100. In Equations 5.2 and 5.3, $\tilde{D}$

represents the predicted depth map and D represents the ground truth depth map. The parameter n is the total number of pixels in the depth map D.

$$M = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{D_i - \tilde{D}_i}{D_i}\right| \tag{5.3}$$

There are several shortcomings for the MARE metric. For instance, the MARE cannot be used when there are zero values in the ground truth. To solve this issue, a mask indicating the zero values of depth map was created. The MARE was then calculated only over the valid pixels of ground truth. The MARE also emphasises more on negative errors where $D_i < \tilde{D}_i$ [181]. Therefore, other measures are usually provided for error measurement besides MARE to provide unbiased reports for depth prediction evaluation.

Evaluation Results

The comprehensive results achieved by simultaneous regression of grasp and depth maps are shown in Table 5.1. There is a slight drop of $\sim 2\%$ in overall accuracy performance of grasp estimation for the BOC setting compared to previous chapter indicating that depth features do not contribute to prediction of grasp maps. A factor which can be attributed to this drop is the increased hardship of learning two tasks together. It is also worth highlighting that the concurrent prediction platform requires almost twice the training time as the single branch prediction one. The overall time however is still acceptable for real-time implementation of such setting.

Figure 5.8 illustrates the algorithm performance over some unseen objects. The selected grasp maps by the GMM ranking were coincidentally the first one in these items. For this figure it should be noted that the depth values are not the actual values but the normalised values when used for training. That is why the same table, on which the object is located is shown with different colors (depth values) depending on the cropped window. For error calculation, the actual depth values were investigated.

In general, multi-task learning is beneficial when the two tasks are correlating to each other properly. In such scenario, each task contributes to the other one and both tasks learn better while richer features are extracted [178]. This however is not the case when two tasks are relatively distinct, which is the case in this part. Grasp belief maps represent different information from the depth maps and therefore learning each task is harder as a trade-off between the features learned for each task should be held. This issue can be observed in the learning curves for similar grasp belief map learning settings implemented for single-task prediction in previous chapter and two-task prediction here in the BOC setting (Figure 5.9). The curves clearly

Table 5.1: The overall performance for concurrent grasp and depth map estimation. The grasp maps are evaluate with the metric suggested in Section 4.4.4, measuring both IoU and difference in angle of rotation. The depth maps are compared using two measures of MARE and RMSE.

Prediction	Method	WOC	BOC	Time (s)	Architecture
Grasp($M = 5$)	IoU + angle	87.2	88.6	0.13	Concurrent
	IoU + angle	91.1	90.6	0.056	Single
Depth	MARE	0.21	0.25	0.13	Concurrent
	RMSE	16.7	22.82	0.13	Concurrent

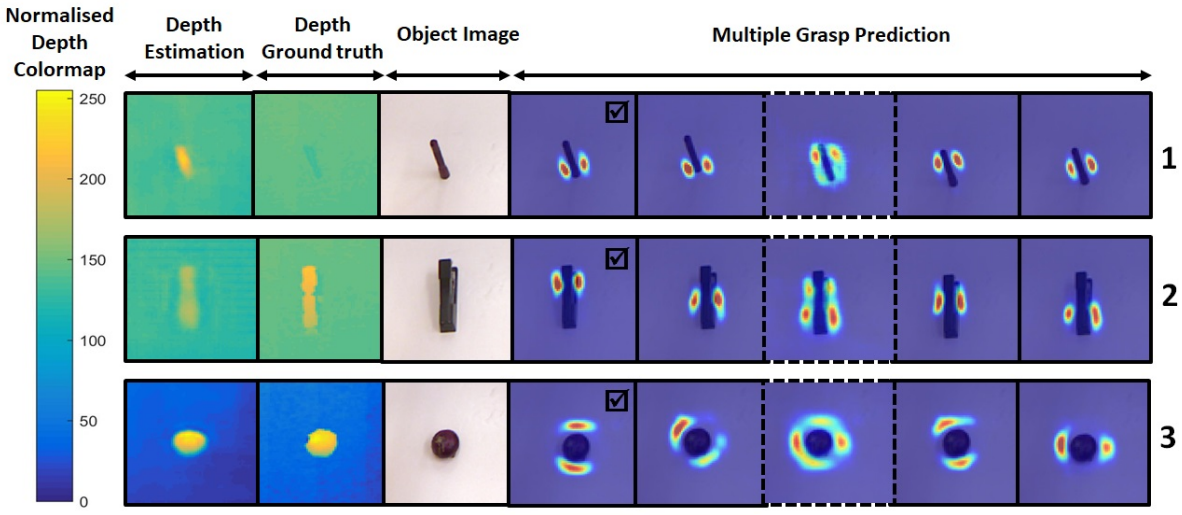


Figure 5.8: The results of simultaneous grasp and depth map estimation for unseen objects. A solid frame around an image is an indicator of grasp detection success, while a dashed line shows an incorrect detection. The images with the ✓ are the top-ranked predictions picked by the designed GMM likelihood estimation module.

indicate that the objective converges faster for the single-task setting during the same number of epochs. Additionally, Figure 5.8 indicates that the third hypothesis is not learned well for all the objects. That can be due to effect of having multiple tasks and therefore a more challenging learning procedure. These incorrect hypotheses are however neglected when using the GMM ranking and selected as the last possible option among the hypotheses.

The results represented in Table 5.1 and Figure 5.8 suggest promising performance for the depth prediction task. Specifically, when an object is shiny, transparent or tiny, the common depth cameras are not sufficiently sensitive to capture them. Accurate depth estimation from an RGB image however can correctly predict the depth for such items as shown in Figure 5.10. Interestingly for object 1, the depth prediction considered the object as a vertical one and gradually increased the depth while getting closer to the camera, whereas the ground truth failed to represent this change. Such cases can be a cause of error when comparing the predicted depth

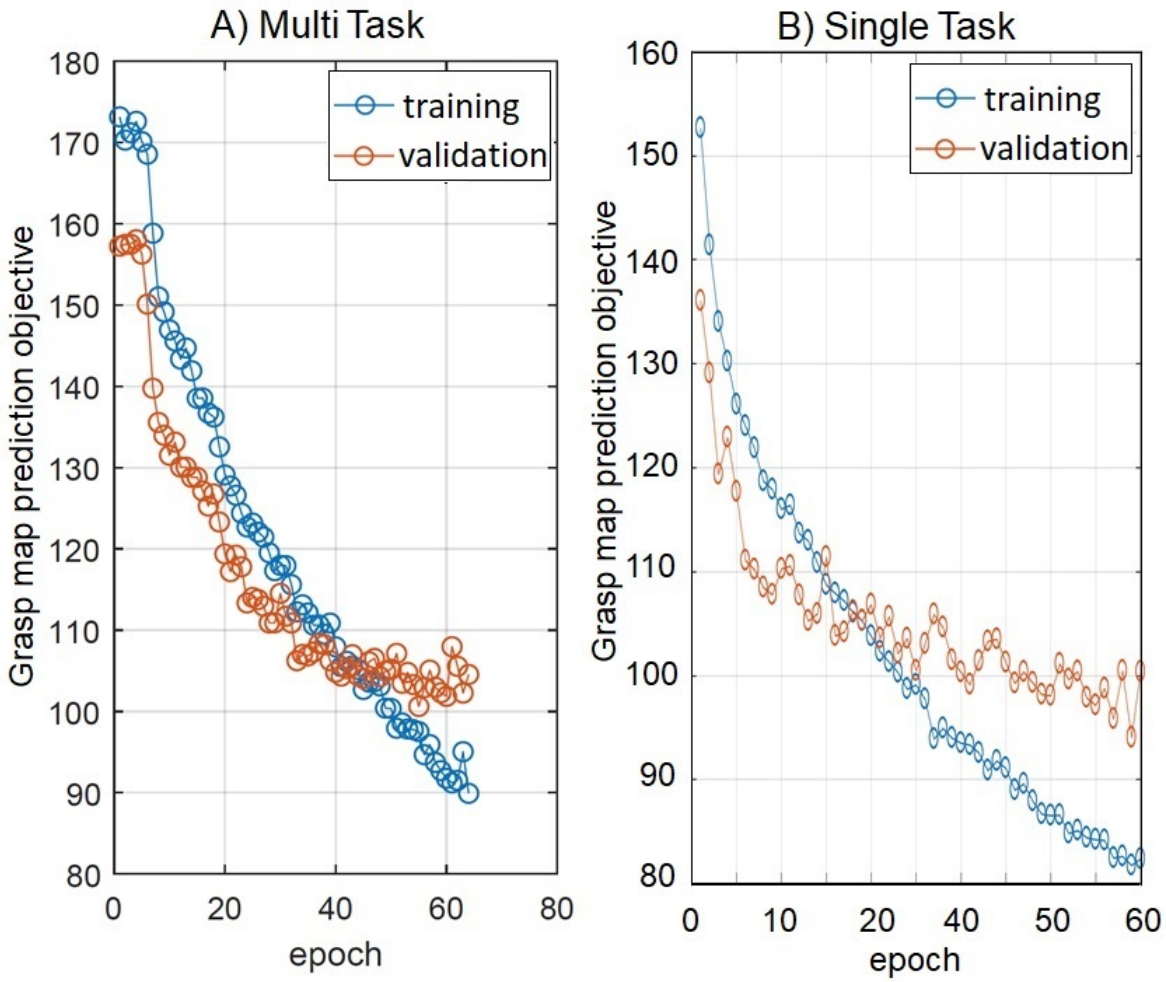


Figure 5.9: A comparison between the training curves for multi-task and single-task learning of grasp belief maps. A) Training grasp belief maps together with depth maps B) Training grasp belief maps only.

with the ground truth one and reduce the actual performance of system.

Although depth prediction brings about some limitations to the developed system, it can play an important role when distance causes detrimental effects on the system's classification performance, which will be discussed further in the next section. The evaluation of depth estimation indicates a MARE error of 25% and RMSE of ~ 23 showing that there is approximately 23 cm error in depth prediction of an image in BOC setting. This 23 cm seem acceptable as such change in distance cannot potentially cause a misclassification in grasp types. The more important aspect to be investigated is the algorithm performance for real objects. Although there is negligible difference in the overall accuracy for multiple grasp prediction in WOC and BOC settings, the depth prediction seem to be easier to be done in the WOC setting. This conclusion can be drawn by comparison of MARE and RMSE values for both settings, in which the WOC setting shows $\sim 5\%$ and 6 cm less error than that of BOC for these measures respectively.

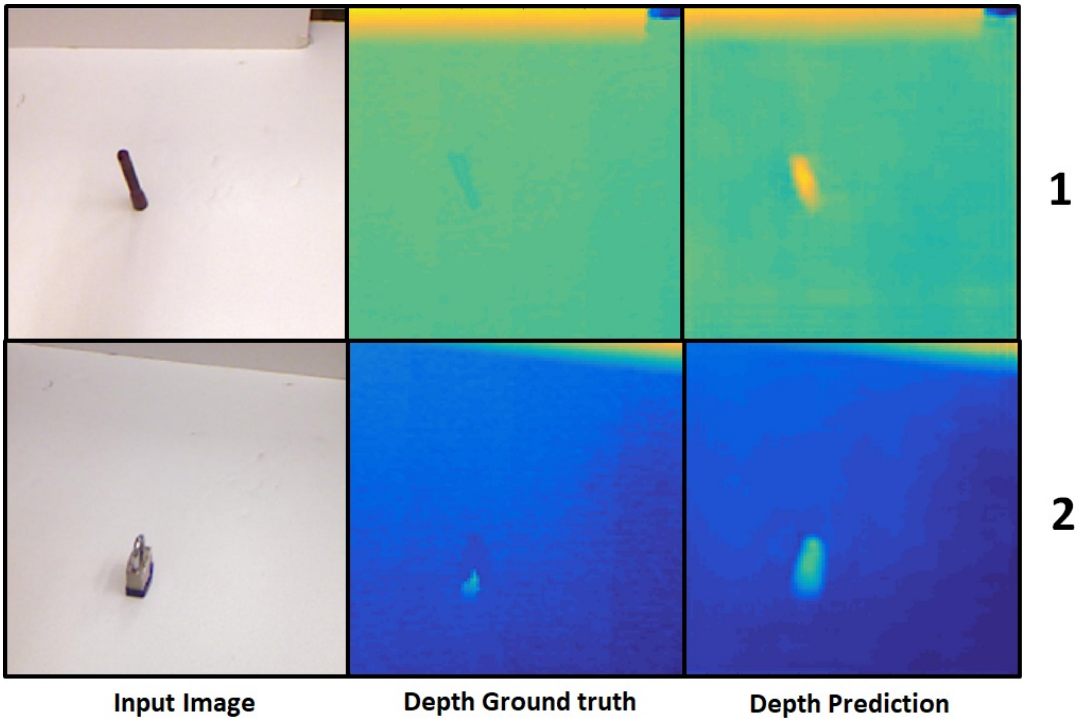


Figure 5.10: Highlighting two cases in which the depth ground truth is not accurate and almost missing an object, while the depth prediction provides a more precise estimation.

5.3.5 Grip Pattern Classification

Experimental Setup

The image sizes and specific settings of each architecture are described thoroughly in their relevant architectural illustrations (Figures 5.2 to 5.7) and Table 5.2. All the input RGB images are of size 224×224 pixels and when fed to the FCRN, the outputs size is always 112×112 pixels. The input to ResNet-50 is also of size 224×224.

The exact same augmentations as the Section 5.3.4 were applied here leading to 4250 images. Accordingly, all the images were nomalised to the range of $[0, 255]$. Mini-batch stochastic gradient descent optimiser with momentum of 0.9 and weight decay of 10^{-3} was used for training. The learning rate was set to 10^{-5} and a dropout layer with rate of 0.8 was used. The models were trained for 40 epochs with batch size of 10. Finally, the weight value chosen for architecture A 4 was set to 0.5 after carrying out trial and error steps.

Evaluation Metric

The evaluation metric for grasp classification is the average accuracy regarding the top-1 class that is the class, which gained the highest probability among all.

Evaluation Results

The performance of all the implemented architectures in the two validation settings of between- and within-object cross-validation are included in Table 5.2. According to this table, the best grasp classification accuracy for seen and unseen objects is achieved with the A 5 and A 6 architectures.

Table 5.2: Average grasp classification accuracy of different architectures (A 1-A 6) designed and illustrated in Section 5.2.2 and the implementation from the set-up in Chapter 5 in different validation settings. GHM stands for grasp heat maps and FCRN stands for fully convolutional residual network used for grasp and depth map estimation. The term frozen means that the frozen network is not trained and only used for production of input to another part.

Architecture	WOC	BOC	Time (s)	Information source	training
A 1-1	83%	78%	0.07	RGB	end-to-end
A 1-2	67%	63%	0.14	RGB + GHM	frozen FCRN
A 2	69%	64%	0.13	RGBD + GHM	end-to-end
A 3	75%	71%	0.21	RGBD + GHM	frozen FCRN
A 4	83%	75%	0.13	RGB + GHM	frozen FCRN
A 5	89%	**81%**	0.21	RGB + GHM	frozen FCRN
A 6	**90%**	80%	0.21	RGB + GHM	frozen FCRN
Ch 5 [157, 182]	85%	75%	0.15	RGB	2-layer CNN

As Table 5.2 illustrates, the forward path for all the implemented architectures takes no more than 210 milliseconds on the GPU, which in applicable to real-time implementations. All the architectures yield to a better average accuracy for the WOC setting compared with the BOC one. Such observation is predictable as the WOC setting includes the objects that are already seen and therefore the decision making for the algorithm is less challenging for this task. The results are consistent in the performance for both settings. That is, the architectures A 5 and A 6 offer the highest grasp classification accuracy in both WOC and BOC settings compared to the rest of architectures. Moreover, the grasp classification performance ranking remains consistent within the architectures.

The architectures A 5, which achieved the best grasp classification accuracy for unseen objects was evaluated for grasp classification of several objects not included in the Cornell grasp dataset. Figure 5.11 depicts these objects and the corresponding grasp class suggested by the A 5 model. It can be seen that the algorithm is not sensitive to variations in background, illumination and camera view. This real-time test is consistent with the offline results in terms of accuracy leading to $\sim 83\%$ grasp recognition accuracy.

Figure 5.11: Grip pattern classification of several unseen household objects. Each object is placed in front of its predicted grasp pattern. The dashed frame is an indicator of an incorrect grasp.

5.4 Discussion

5.4.1 Simultaneous Depth and Grasp Map Estimation

A setting was provided for concurrent depth and grasp map estimation of objects which led to real-time performance of both tasks. Considering the inherent diversity between the two tasks, the performance can be deteriorated in an MTL platform. The grasp estimation performance however dropped only 2%. This drop could probably be avoided if the network is trained for a longer time, since the training curve convergence was significantly slower for this MTL platform.

The depth prediction also provides reliable results within the range of 20 cm. The depth estimation can potentially be improved by training the network on more samples including a variety of scenes. Here, the effort was to obtain results with the least amount of data and stick to the current available dataset. The network was trained on roughly 200 objects only with 2-8 views for each one. Having such an accuracy with this limited number of objects and views indicates the potential of the designed structure in presenting even better performance when trained with a more comprehensive dataset including more instances and object, scene and viewing point varieties.

5.4.2 Grip Pattern Classification

To find the best structure for using the predicted grasp belief and depth maps, a variety of architectures were implemented to employ the available sources of data. Among those imple-

mentations, two architectures (A 5 and 6) indicated the best grasp recognition performances. The grasp recognition of the architecture A 5 was evaluated for some real objects unseen to the trained network.

One challenge in performance analysis for real objects is that the actual annotations for these objects are not known. Also grasping preference varies among people. To avoid any bias in annotations, the ground truth labels provided by Degol *et al.* [180] were utilised for the Cornell grasp dataset. These annotations however are not consistent with the annotating approach of Chapter 5. Additionally, there are some inconsistencies in the offered ground truths. That is, sometimes objects belonging to the *tripod* grasp group are labeled as *pinch*, which can degrade the performance by confusing the network.

Considering the small size of dataset and the issues with annotations, the designated architectures still provide better grasp classification accuracy in both of the WOC and BOC settings compared to both Chapter 5 [157, 182] and Degol *et al.* [180]. The latter offered a 20% accuracy when classifying unseen objects trained on the Cornell grasp dataset. Regarding the work done in Chapter 5, it should be noted that the combined dataset of the ALOI and Newcastle objects included 482 categories of objects with 72 views per category. Contrarily, the Cornell grasp dataset [17] used in this chapter consists of only 235 object categories with 2-8 views for each one. As already mentioned, data availability plays an indispensable role in training a deep network [13, 14]. With such a smaller amount of data, the designed architectures, A 5 and 6, still outperform all the previous state-of-the-art vision-based grasp recognition structures [157, 180, 182].

It is worth to note that, both A 5 and A 6 architectures with the top performances among all the architectures do not employ the depth data. In the initial efforts for using all the data sources, the depth data did not contribute much to the task of classification and therefore was set aside. This source of information however can be used for ensuring that the object is within a distance range that can be grasped with convenience. In real-time applications, when a snapshot is taken from a further distance than *e. g.* 30 cm from an object, a range which provides convenient grasping, the image can be re-scaled accordingly in case grasp identification is affected by the change in distance.

The results also indicate that benefiting from grasp belief maps can boost the overall performance at least $2-3\%$. Another merit of using the grasp belief maps is the orientation prediction, which can aid in posing the hand wrist in the appropriate orientation for grabbing an object.

5.4.3 Conclusion

In this chapter, a platform for concurrent depth and grasp belief map estimation for common objects included in the Cornell grasp dataset was suggested. These sources of data were subsequently used in combination with the input RGB data to create an accurate grasp classification framework. A handful of architectures were designed and evaluated for grasp recognition of common objects. The architecture presenting the most accurate grasp choices among all for novel object categories was tested on real objects. The designed platforms presented state-of-the-art grasp classification performance for unseen objects with the possibility of real-time implementation.

Chapter 6

Conclusions and Discussions

In this chapter a summary of the implemented structures is presented and the limitations and capabilities of each system are further highlighted.

6.1 Overview and Contributions

In this thesis several deep learning architectures are developed in order to improve the performance of a current commercial artificial hand in grasping objects. To achieve this goal, the deep learning architecture should meet three objectives: 1) efficient grip pattern recognition, 2) grasp detection and recognition for novel objects and 3) provision of a user-friendly structure which minimises the burden of the task on user.

In the following, a summary of the developed systems with respect to the aim and objectives is provided.

6.2 Deep Learning-based Artificial Vision for Grasp Classification in Myoelectric Hands

This contribution is a proof of concept for a vision-based prosthesis controller. That is, it was proved that the augmentation of a hand prosthesis with a camera and a deep learning module can provide reliable grasp input for the hand prosthesis such that the user skips the burdensome procedure of grasp selection via the EMG signals. The deep learning module in this part consists of a two-layer CNN that classifies objects based on four grip patterns. The input to this module is a processed RGB image captured by an inexpensive camera, which acts by a command recorded from the muscle activity of the user.

The developed system yielded reliable performance in real-time experiments with two amputee participants. Having observed a myoelectric pattern recognition experiment on the same day with the same subjects made us realise the considerable decrease in the difficulty of grasping task via the vision-based system. By proving the concept of vision-based control of artificial hands and offering promising performance, further investigations were carried out to improve the deep learning module.

The deep learning structure developed in this system is comparatively simple and requires a preprocessing step in which the object background is removed. Although rarely being a source of error, this simple structure limits the usable input size to 36×48 as larger sizes require more processing capacity as well as deeper architecture. The overall offline grasp recognition accuracy with the implemented architecture reached to 75% for novel objects. This issue was further investigated by employing transfer learning (Section 3.6), such that a much deeper architecture (ResNet-50) was fine-tuned for the task of grasping with 224×224 images. Such setting however led to 71.5% accuracy. It was concluded that the choice of network did not play an essential role in the task of grasp classification. The impact of using datasets with more variety of objects and data augmentation is however open to further discussion.

Another improvement to the proposed CNN structure can be incorporating depth data together with the RGB information for more accurate object detection and distance estimation as well as dealing with the variable distance of the hand from object during snapshot capturing. To this end, RGB-D images can be converted to point cloud data to be employed within a model such as PointNet [183] to provide grasp classes [184]. This solution can provide better classification accuracy as the grasp type choice relies considerably on the object sizes as well as objects' 3-D shape. This approach however requires depth sensors that are sufficiently accurate for granular and close objects.

Considering that neither the architecture nor the image size or quality are the major constraining factors in grasp recognition performance of the suggested structure, a change of perspective can be the key to further performance improvement. To do so, more attention was drawn to robotic grasping solutions in which advanced deep learning structures led to promising performances, as after all a prosthetic hand is a robotic gripper.

6.3 Accurate Object Localization and Grasp Map Estimation in Presence of Ambiguity

This contribution was developed with more concern about the task of grasping. That is, efforts were made to estimate the position, orientation and size of a grasp for novel objects. To do so, inspired by the recent trends in robotic grasping [17, 126, 127, 131, 135–137] the grasping task was defined as prediction of grasp belief maps through which the desired grasp information can be represented. These belief maps not only indicate the size, orientation and position of a grasp, but also include some uncertainty in the grasp through their Gaussian mixture representation. This uncertainty can be beneficial in modeling the grasp considering the high ambiguity of the task.

An FCRN architecture was implemented to predict the grasp belief maps for a target object. The investigations indicated that the grasp definition is not sufficient for accounting for the large amount of uncertainty in the task. The reason is the possibility of several grasps per object. Ignoring this fact and assuming only one grasp for each object can confuse the algorithm. To tackle this issue, the single prediction platform was converted to a multiple prediction one to be able to estimate several grasps for a single object. The multiple grasp prediction structure indicated state-of-the-art performance for grasp estimation of novel objects.

This contribution led to a high accuracy in grasp pose, size and orientation estimation. These parameters however cannot be fed into a prosthesis hand as a grasp. Therefore, a method for employing these parameters for precise grasp identification is required.

Moreover, the system is designed to estimate grasps for non-cluttered scenes. This however is not the case for real world applications. Hence, further adjustments can be done to improve system performance for cluttered scenes. One solution could be employing object detection architectures such as mask R-CNN [122] or regressing a bounding box for each object in the scene together with the grasp belief maps for that object in a multi-task learning framework. These tasks however require an appropriate dataset including cluttered scenes or augmentation of object images artificially into cluttered scenes.

It is worth to note the presence of depth data for all the images in the Cornell grasping dataset, which can be used for developing a system with the capability of concurrent depth and grasp map prediction. If accurate depth estimation is achieved, an additional source of information is provided for object detection and grasp classification.

6.4 Grip Pattern Classification for Prosthetic Hands using Estimated Grasp and Depth Maps

This contribution involves the design of an architecture for grasp classification using the grasp maps produced in the second contribution. To increase the amount of available data, depth information of the scene was also predicted. Depth information can be crucial whenever the algorithm indicates sensitivity to distance variation. This was the case in the system developed for the first contribution. When the distance of the target from the camera exceeded from a specific amount, the grasp prediction performance deteriorated.

Having several sources of information including grasp belief maps, depth and input RGB image creates a variety of scenarios for grasp classification. Several architectures were designed using these resources and their performances were compared. Opting the architecture with the best performance, the average accuracy of 81% was achieved for novel objects, which is $\sim 6\%$ better than that of the first contribution. The algorithm is efficient and implementable in real-life systems as it works in real-time. Besides, GPUs are available in a variety of sizes and usable in almost every system.

This system offers an accurate grasp classification framework. Unlike the second contribution, the grasp can be used in a prosthetic hand similar to the first part. The suggested structure offers state-of-the-art performance for vision-based grasp recognition. Having a pattern recognition-based hand such as the COAPT system [36] commercially available, provides the opportunity of improving both systems even further by using two resources of information (vision and EMG) for decision making.

A future enhancement to this grasp identification scheme is to extend it to more number of grasp types. This idea requires a proper grasping dataset, including grasp rectangles and ideally RGB-D data of different views of a large variety of common objects with sufficient samples for each grasp category.

As adequate training images are required for an accurate prediction within deep networks while abundant amount of unlabeled images of objects are available online, semi-supervised learning [185] can be exploited to benefit from this massive source of unlabeled data for the purpose of grasp classification. There are however numerous methods for semi-supervised learning. One example can be the work done by Mobahi *et al.* for object classification using coherence in different views of same object. Exploiting the notation of temporal coherence [186, 187] and adapting it to suit the grasp recognition problem, one can treat successive

images as a video stream such that a scale- and pose-invariant representation of images can be learned. To illustrate, successive images include large amount of data that is similar to each other and this data can be used for better training of a grasp classiying network. By applying a regularisation factor through a similarity function, the similarity in similar views of an object may be reinforced. In addition, representations of non-consecutive frames of the same object may be pushed apart. Hence, the algorithm can intelligently learn to treat similar views of the same object alike. To expand this idea, such similarity can be applied to grasp groups as well. An alternative to this method can also be canonical correlation analysis [188], in which multi-view learning is used to provide a predictor through a semi-supervised algorithm. Implementing either of these ideas may boost the robustness against scale and pose for the grasp recognition task.

Another future step for improvment of the current structure is to include users' preference in grasp labeling of objects. To both achieve this purpose and boost the performance, a reinforcement learning structure [189–191] can be designed and implemented. Reinforcement learning achieved great success in robotic grasping [128]. Moreover, it has also been used to improve the performance of hand prosthesis using EMG data [192, 193]. To have such a system, through real-time training of a deep learning structure, not only the grasp recognition accuracy elevates, but also the grasps will be adjusted to the majority of subjects' preferences. That is, learning with a pre-trained deep network, the weights are updated in real-time based on user decisions or a reward function quantifying the goodness of the identified grasp. After performing the task by several users for several objects and trials, the algorithm can gain more robustness through real-time training. Moreover, the performance can get gradually better as the weights are updated and adjusted based on the user's preference or grasp quality.

6.5 Conclusion

This thesis provided three main contributions presented chronologically. The first contribution demonstrates a proof of concept for a vision-based artificial hand controller featuring a deep learning module. The second contribution designs an accurate grasp regression platform. The third contribution benefits from both works to propose an accurate grasp classification structure which can be utilised in a commercial hand prosthesis.

Appendix A

CNN Building Blocks

This appendix provides mathematical background for the CNNs. First a detailed explanation over the building components of CNNs, such as convolution and pooling, are presented. The contributing factors to the performance of CNNs such as activation functions, pooling methods and optimisation and regularisation techniques for exploiting the full potential of network are elaborated in this appendix.

A.1 Mathematical Description of Convolutional Neural Networks[1]

Assume that there are m^l input maps of size $R^l \times U^l$ in each of the CNN layers l, such that m^0 represents the number of images in the $0-$th layer (input images). At each layer, features are extracted by convolving each input map with k^l kernels of size $C^l \times D^l$ ($C^l < R^l, D^l < U^l$) as shown in equation A.1.

$$\begin{gathered} \mathbf{Z}_{ij}^l = (\mathbf{X}_{ij}^{l-1} * \mathbf{K}_j^l + \mathbf{b}_j^l) \\ \mathbf{X}_{ij}^l = a(\mathbf{Z}_{ij}^l) \\ i = 1, 2, \ldots, m^l, \quad j = 1, 2, \ldots, k^l, \quad l = 0, 1, 2, \ldots, L \end{gathered} \tag{A.1}$$

where, $\mathbf{Z}_{ij}^l$ is a $(R^l - C^l + 1) \times (U^l - D^l + 1)$ matrix of convolved features (feature maps) resulted from convolving the i-th input map from the $(l-1)$-th layer ($\mathbf{X}_{ij}^{l-1}$) and the j-th kernel in the l-th layer ($\mathbf{K}_j^l$) and adding the bias $\mathbf{b}_j^l$. By element-wise application of a non-linearity through the activation function $a(\cdot)$ to the resultant feature map, the output of layer l is achieved. In equation A.1, the sign $*$ refers to a *valid* convolution known as a convolution performed inside

[1] All the equations in this chapter are written in the vectorised format.

the image borders.

The next layer in CNN is normally pooling, in which a region size is selected, *e. g.* $S_1 \times S_2$, where $(R^l - C^l + 1)/S_1$ and $(U^l - D^l + 1)/S_2$ should be an integer value. The convolved features are partitioned into $S_1 \times S_2$ sub-regions. In each sub-region, the maximum or average is selected as explained previously. The result can be used for classification through a fully connected layer [194] or followed by several other layers of convolution and pooling and then classified. Figure A.1 depicts the first layer of a CNN structure including extraction of feature maps and the pooling mechanism.

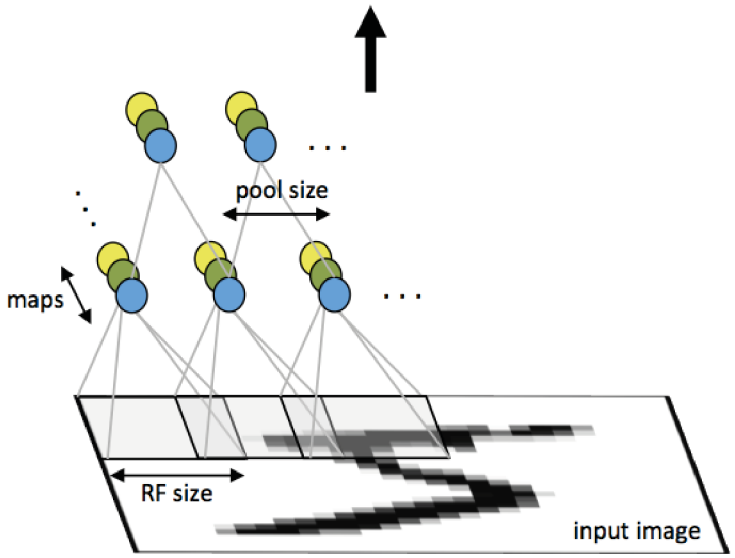


Figure A.1: First layer of a convolutional neural network indicating a convolution layer followed by pooling. Units with the same colour share weights (taken from [19]).

A.1.1 Back Propagation

Back propagation is a training technique in neural networks, in which the errors are propagated backwards through the last layer of the network (output layer). Through back propagation the classification/regression error calculated at the last layer is used for updating the weights of filters/kernels within each layer. The procedure of error propagation when layer l is densely connected to layer $l+1$ is described in Equation A.2.

$$\mathbf{\Delta}_{ij}^{l} = ((\mathbf{K}_{j}^{l+1})^{T}\mathbf{\Delta}_{ij}^{l+1}) \bullet a'(\mathbf{Z}_{ij}^{l}) \tag{A.2}$$

where $\mathbf{\Delta}_{ij}^{l}$ denotes the error matrix for the i-th input map and j-th kernel in l-th layer. The training pairs of $\{\mathbf{X_{ij}^{0}}, y_i\}$ include input images and their specified labels respectively. $a'(\mathbf{Z}_{ij}^{l})$

indicates the derivative of the activation function. The sign "•" is an indicator of element-by-element multiplication. The corresponding gradients to this layer for weights and biases can be calculated accordingly:

$$\bigtriangledown_{\mathbf{K}_j^l} J(\mathbf{K}, \mathbf{b}; \mathbf{X}, \mathbf{y}) = \boldsymbol{\Delta}_{ij}^l (\mathbf{X}_i^{l-1})^T, \quad \bigtriangledown_{\mathbf{b}_j^l} J(\mathbf{K}, \mathbf{b}; \mathbf{X}, \mathbf{y}) = \boldsymbol{\Delta}_{ij}^l \tag{A.3}$$

where $J(\mathbf{K}, \mathbf{b}; \mathbf{X}, \mathbf{y})$ represents the cost function for the training set $\{\mathbf{X}, y\}$ and network with kernels ($\mathbf{K}$) and biases ($\mathbf{b}$).

Usually there is a pooling layer right before a fully connected layer. Additionally, pooling layers normally follow convolution layers. Hence, considering the l-th layer as a convolutional layer followed by a pooling layer (sub-sampling) , the error in l-th layer is propagated through as:

$$\boldsymbol{\Delta}_{ij}^l = \text{upsample}((\mathbf{K}_j^{l+1})^T \boldsymbol{\Delta}_{ij}^{l+1}) \bullet a'(\mathbf{Z}_{ij}^l) \tag{A.4}$$

Please note that the up-sampling operation depends on the pooling technique. If max pooling is used, the unit chosen as the maximum receives all the error, while having average-pooling, the up-sample function uniformly distributes the error among the units.

Now the error is propagated through the convolution layer. The gradients with respect to feature maps are calculated as:

$$\bigtriangledown_{\mathbf{K}_j^l} J(\mathbf{K}, \mathbf{b}; \mathbf{X}, \mathbf{y}) = (\mathbf{X}_i^{l-1}) * (\boldsymbol{\Delta}_{ij}^l), \quad \bigtriangledown_{\mathbf{b}_j^l} J(\mathbf{K}, \mathbf{b}; \mathbf{X}, \mathbf{y}) = \sum_{a,b} (\boldsymbol{\Delta}_{ij}^l)_{a,b} \tag{A.5}$$

which follows the same steps as general back propagation steps (Equation A.3) [19, 195]. In Equation A.5, a 180° rotation is applied to the error image to perform a cross-correlation instead of convolution to rotate the output back. In this way, in the feed-forward path, the kernel has its expected orientation.

Having more than one convolutional and pooling layer leads to deep CNNs, in which more complex features can be extracted and higher performance may be achieved. The back propagation procedure follows Equations A.4 and A.5 repeatedly for each additional layer.

There are a variety of enhancements for the above mentioned general method, such as local contrast or batch normalisation techniques. There are a variety of choices for the activation function, which can have a significant impact on learning capability of the network. A common issue in deep networks is the over-fitting, in which the network parameters are overly tuned for

the given data and as a result the network cannot generalise to new data properly. As a solution to this problem, there are various regularisation techniques from which dropout [20] indicated promising results. Batch normalisation [196] is a more recent technique for accelerating and improving network's learning ability, which also contributes to network's regularisation.

In supervised learning, the last layer of a network is a fully connected layer, comprising a neural network or any other kind of classifier such as SVMs [163] so that groundtruth labels can be compared with the predicted ones. According to the best practices in CNN [15,101,158,197], utilising a Softmax regression, which is a linear classifier that uses log probability distribution, works well for classification in CNNs. Another reason for using Softmax is that the output of Softmax provides the probability of each class and facilitates examining the performance of algorithm.

It is worth mentioning that the depth of the network is an important factor to achieve a better performance, as potentially more abstraction can be learned through a deeper network. However, there is a trade-off among the level of abstractness, performance, training time and computations.

In the subsequent sections more detailed explanation over each component of CNN and available techniques can be found.

A.1.2 Preprocessing

Each image is usually normalised before being fed into a CNN so that pixel values are in an acceptable range and the image distribution is centralised. Equation A.6 represents zero mean and unit variance normalisation (Z-scores).

$$I_{normalised} = \frac{(I - \mu_I)}{\sigma_I} \tag{A.6}$$

where

$$\mu_I = \frac{1}{N+M} \sum_{n=1}^{N} \sum_{m=1}^{M} I_{n,m} \tag{A.7}$$

and

$$\sigma_I = \sqrt{\frac{1}{N+M} \sum_{n=1}^{N} \sum_{m=1}^{M} (I_{n,m} - \mu_I)^2}. \tag{A.8}$$

where I is an input image matrix of size $N \times M$ pixels and $I_{n,m}$ denotes the intensity for pixel

(n, m). The preprocessing steps should be applied to both train and test images.

A.1.3 Convolution

One way to automatically extract features from an input image is to connect all the pixels of the image to all the hidden units of a neural network and try to learn appropriate weights. This procedure however is computationally expensive and the amount of computations and number of parameters to be learned raise considerably with increasing the input image size. As a solution, the connections between the hidden and input units can be restricted to only a small contiguous region of pixels in the input image. This kind of problem modeling closely resembles how neurons in the visual cortex have localised receptive fields.

Due to the stationary nature of images, features learned at a specific patch over an image can be helpful for detecting features at other image patches. Therefore, a $C \times D$ filter can be convolved through all the patches of the same size within an image and extract desired features.

There are two other parameters to be considered in a convolution: stride (s) and padding (P). The former specifies the step size of convolution window. Thus, when less overlapping among receptive fields or smaller output spatial dimension is desirable, the stride will be elevated. The padding as its name represents defines a pad around the border such that convolution does not cause information loss and input size shrinkage. Accordingly, the output size can be calculated using Equation A.9, where (O_x, O_y), (R, U), (C, D), P, s define the output map, input map, filter, padding and stride sizes respectively. Figure A.2 a indicates how convolution is performed in details.

$$\begin{aligned} O_x &= \frac{(R - C + 2P)}{s} + 1 \\ O_y &= \frac{(U - D + 2P)}{s} + 1 \end{aligned} \tag{A.9}$$

A.1.4 Activation Function

After each convolution (and addition of a bias term to the result), a nonlinear function is applied to convolved features to add non-linearity to the network. The reason is that convolution and bias addition procedure are linear and to model nonlinear functions some non-linearity within network is required. Consequently, each feature map undergoes an activation function, which can conventionally be of three types:

Logistic function (Sigmoid)

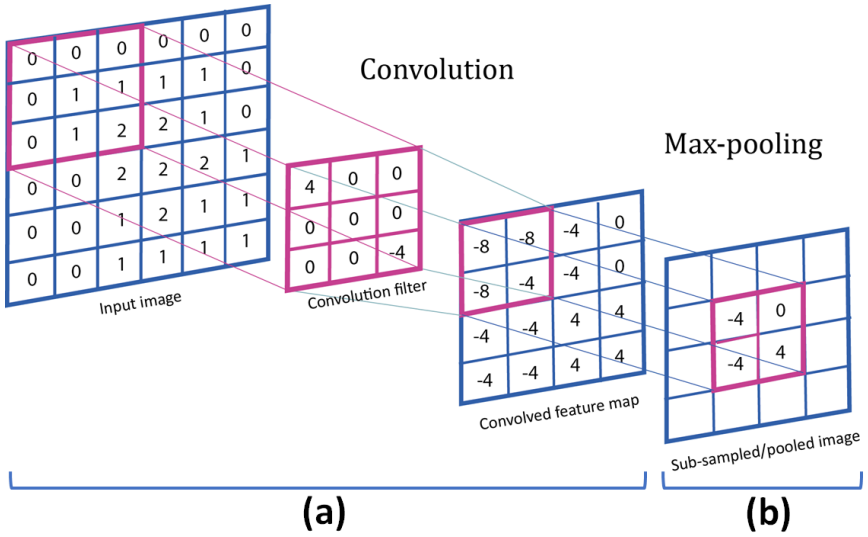


Figure A.2: A convolution operation (a) followed by a max pooling (b).

The output varies between 0 and 1, the same as a step function, but with the addition of a region of uncertainty. That is why the sigmoid function can be considered as having similar functionality as input-output relationships of biological neurons.

$$a(t) = \frac{1}{1 + e^{-\alpha t}} \tag{A.10}$$

$$\frac{da(t)}{dt} = (\alpha)a(t)(1 - a(t)), \tag{A.11}$$

where α is the slope parameter and $a(.)$ indicates the activation function. According to Equation(A.11), which indicates the derivative of a sigmoid function, the bigger the α parameter, the greater the slope and the sigmoid is more similar to a threshold (step) function.

Hyperbolic tangent (tanh)

This function provides similar output as sigmoid function, while the output range is between -1 and 1:

$$a(t) = \frac{1 - e^{-2t}}{1 + e^{-2t}}. \tag{A.12}$$

Rectified linear unit (ReLU)

The rectified linear unit more closely resembles biological activations of neurons compared to sigmoid activation function [197, 198]. This function is bounded by a minimum

value, typically zero and unbounded by maximum value, which makes it able to represent any non-negative real value. Having a real zero activation value, the function also benefits from good sparsity properties. These properties cause this function to suffer less from diminished gradient flow.

Alleviating the vanishing gradient problem is probably the main reason for using ReLU over the other activation functions. That is, both sigmoid and tanh functions bound the output in a small range with values smaller than 1. When using gradient-based methods, in which the effect of a small change in a parameter's value on the output value is observed, the change of parameters will be diminished through the layers. The deeper the network the more intense this gradient vanishing is, as multiplication of small values leads to even smaller values. Hence, the change of parameters will not be visible in the output and as a consequence, the network cannot learn properly. This problem can be overcome by the use of ReLU in which the output range is not bounded.

Equation A.13 depicts the ReLU activation function. ReLU is widely utilised in successful deep network arcitectures [15, 158]. ReLU is also computationally more efficient than the mentioned activation functions. Finally, ReLU as a non-saturating non-linearity is several times faster than sigmoid and tanh, which are saturating non-linearities in terms of training time with gradient descent optimisation.

$$a(t) = \max(0, t). \tag{A.13}$$

Figure A.3 indicates sigmoid, tanh and ReLU activation functions and their range compared to each other.

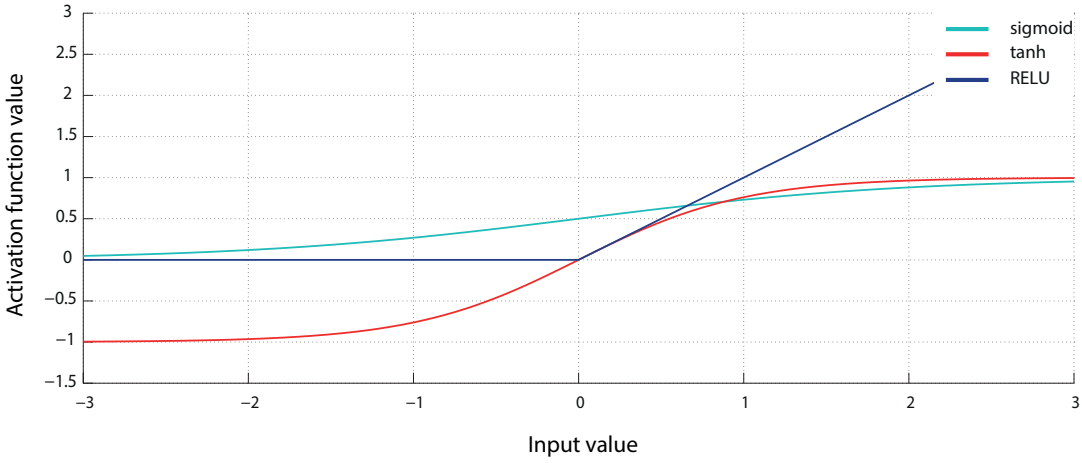


Figure A.3: Sigmoid, tanh and ReLU activation functions output according to same input values

A.1.5 Pooling

During pooling, the size of the feature maps, which are the outcome of activation functions, are decreased, bringing about feature maps with reduced spatial dimension. This step reduces spectral variance in the input features.

Pooling brings about two major benefits: reduction in the number of parameters and thus computation cost and control of overfitting. The latter could be due to production of relatively invariant features when aggregating multiple low-level features over a small neighborhood.

Non-overlapping pooling is more commonly utilised as overlapping pooling does not provide any extra information compared to non-overlapping pooling nor yield any improvements in recognition rates [158, 199]. The most prevalent and effective pooling functions are average pooling, max pooling and stochastic pooling.

Average Pooling

Average pooling takes the mean value for each pooling region R_j. During back propagation, average pooling requires up-sampling the error matrices [195]. The major drawback of average pooling is low or negative activations downplaying a higher activation value, which causes a near-zero activation function; whereas the high activation of one of the feature detectors may be the most important information. Considering sparsity in the activation function, average pooling can be less effective by giving low activation values.

Max pooling

Max pooling follows the same steps as average pooling except that it takes the maximum value in each pooling region R_j (Equation(A.14)).

$$s_j = \max_{i \in R_j} a_i \tag{A.14}$$

where a_i is each element of the resultant feature map matrix in the pooling region R_i. During back propagation, the error is only back propagated through the maximum values used previously. Figure A.2 b shows how max pooling is carried out in details. In [15, 97, 199], it is suggested that max pooling leads to better performance compared to average-pooling. The main disadvantage with max pooling could be neglecting other values in a pooling region, which may cause over-fitting problems. Figure A.4 indicates an example for the non-overlapping max pooling function.

Stochastic pooling

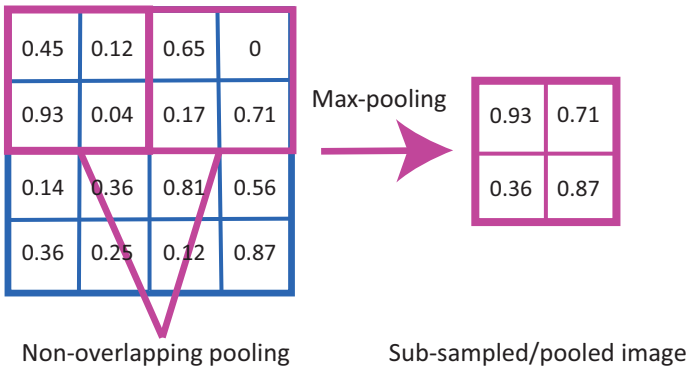


Figure A.4: An illustration of non-overlapping max pooling.

Several works [158, 197, 200] highlight that max pooling over-fits the training data rather quickly and may not generalise it to test data. Their alternative is using stochastic pooling, which is designed to work with the ReLU activation function. Equation A.15 describes stochastic pooling for pooling region R_j. The probability of each non-zero activation value is calculated and one of the non-zero activations is chosen randomly based on a multinomial probability distribution (location l). Experimental results in [200] have shown that stochastic pooling provides a longer training period without over-fitting the model.

$$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}, \qquad s_j = a_l \quad \text{where} \quad l \sim P(p_1, p_2, ...p_{|R_j|}) \tag{A.15}$$

In spite of the presence of several pooling methods, max pooling is still the most popular one [15, 110, 111]. Recent studies have shown that max pooling can be substituted with a convolution with increased stride leading to comparable performance [201]. Since then, network architectures such as ResNet [112] are benefiting from this approach for simplifying the network architecture.

A.1.6 Fully Connected Layers

Similar to convolutional layers, fully connected layers are used for feature extraction. They are usually used as the last layers before the classification to aid the classification by reduction in output nodes or providing more abstraction [12, 15, 110–112].

A.1.7 Softmax Regression

The features attained by pooling or a fully-connected layer can be classified in K classes through Softmax regression. Softmax regression or multinomial logistic regression [52] is a logistic regression generalised for multiple classes. That is, instead of binary labels $y^{(i)} \in 0, 1$, there are K labels (classes) $y^{(i)} \in 1, ..., K$. Having m labeled examples for the training set $(\mathbf{x}^{(1)}, y^{(1)}), ..., (\mathbf{x}^{(m)}, y^{(m)})$, the hypothesis estimates the probability $P(y = k|\mathbf{X})$ for each value of $k = 1, ...K$ (Equation(A.16)).

$$\mathbf{H}_{\mathbf{\Theta}}(\mathbf{X}) = \begin{bmatrix} P(y = 1|\mathbf{X}; \mathbf{\Theta}) \\ P(y = 2|\mathbf{X}; \mathbf{\Theta}) \\ \vdots \\ P(y = K|\mathbf{X}; \mathbf{\Theta}) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)^T}\mathbf{X})} \begin{bmatrix} \exp(\theta^{(1)^T}\mathbf{X}) \\ \exp(\theta^{(2)^T}\mathbf{X}) \\ \vdots \\ \exp(\theta^{(K)^T}\mathbf{X}) \end{bmatrix} \tag{A.16}$$

where $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(K)} \in \mathbb{R}^n$ are the model parameters, which are tuned during training. Equation(A.17) indicates the *cross-entropy* loss or cost function usually utilised with Softmax regression;

$$J(\mathbf{\Theta}) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\{y^{(i)} = k\} log \frac{\exp(\theta^{(k)^T}\mathbf{x}^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)^T}\mathbf{x}^{(i)})}] \tag{A.17}$$

where the function $1\{.\}$ produces a ground-truth vector [202]. $\mathbf{X}$ is the input matrix, in which each row belongs to one example, so $\mathbf{x}^{(i)}$ denotes a vector specific to the i-th example.

The Softmax function has a set of "redundant" parameters, in a way that for any hypothesis, there are multiple parameter settings that lead to the same hypothesis. Interestingly, the minimiser of cost function $J(\mathbf{\Theta})$ is not unique while it is still convex; so it can be optimised using gradient descent.

For classification tasks, Softmax is one of the most common options as the training can happen end-to-end, while Softmax parameters are trained together with the CNN ones [12, 15, 110–112].

A.1.8 Optimisation Techniques

Gradient Descent Gradient descent and its variations, a method first suggested by Cauchy for optmisation [203] is widely used in deep learning since Lecun figured out stochastic gradient descent (SGD)'s capability in training neural networks [104, 204].

Mini-batch gradient descent (GD) is nowadays the most popular method used as the optimi-

sation algorithm for a cost function. In this method, a mini-batch size is chosen (m) and in each iteration, weights (θ) are updated by SDG optimisation for m examples; while in batch gradient descent, in each iteration weights are updated for all the training examples and in stochastic gradient descent they are updated by training only one example in each iteration. Mini-batch gradient descent is computationally faster, although it requires more iterations than batch gradient descent. In mini-batch GD compared to the SGD samples are less noisy as the noises are averaged. That is why, choosing a suitable mini-batch size can be effective in algorithm's learning performance. Equation A.18 demonstrates the update procedure for weights in SGD; α is learning rate, which is another consequential factor in training a CNN and should be chosen carefully.

$$\theta_{i+1} := \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta) \tag{A.18}$$

The higher the α value the faster the training, while high values of α can cause oscillation and divergence of leaning curve. Hence, α should be selected such that a balance between speed and smoothness is maintained. To provide a smoother gradient optimisation, a momentum parameter is used, which takes care of the information learned during previous steps of training [205]. The momentum parameter is applied during the weight updating procedure and provides better convergence and reduces the risk of getting stuck in local minima [205]. Equation A.18 can be modified accordingly for using a momentum parameter (β) as shown in Equation A.19.

$$v_{i+1} := \beta v_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta), \qquad \theta_{i+1} := \theta_i + v_{i+1} \tag{A.19}$$

Local Response Normalisation (LRN) Normalisation can be applied after each convolution to restrict the output values of feature maps or intensify specific features. A simple solution for normalisation is the Z-score normalisation (Equation A.6), which was introduced as a preprocessing step. Local response normalisation (LRN) is a more advanced normalisation method applied to the output of ReLU in AlexNet [15] and can be helpful in generalisation (Equation A.20). Similar to local contrast normalisation [97], LRN brings about competitions between neuron outputs, while this competition is specific to outputs of neighboring kernels at the same layer. To illustrate, output of neurons with large activations are amplified, while uniform responses are dampened [15, 97].

$$\hat{a}^i_{x,y} = \left(\frac{{a^i}_{x,y}}{k + \alpha \sum_{j=\max(0,i-n/2)}^{min(N-1,i+n/2)} ({a^j}_{x,y})^2}\right)^\beta \tag{A.20}$$

where $\hat{a}^i_{x,y}$ is the response-normalised version for ${a^i}_{x,y}$ activity resulting from applying kernel i to the input images at position (x, y). While N is the entire number of kernels and therefore feature maps in the layer, summing happens over n adjacent feature maps. Constants k, n, α and β are hyper-parameters.

Batch Normalisation (BN) Firstly introduced by Ioffe *et al.* [196], batch normalisation (BN) is applied after a convolution layer and before an activation function as a learnable normalisation step. It provides zero mean and unit variance to boost the performance of CNNs [196]. Equation A.21 illustrates how each scalar feature is normalised to zero mean and variance of 1, where $\hat{\boldsymbol{a}}$ represents the normalised d dimensional activation $\hat{\boldsymbol{a}} = (a^{(1)} \cdots a^{(d)})$. $E(.)$ and $Var(.)$ indicate *Expectation* and *Variance* operators over the training set respectively.

$$\hat{a}^{(k)} = \frac{a^{(k)} - E[a^{(k)}]}{\sqrt{Var[a^{(k)}]]}}. \tag{A.21}$$

Batch normalisation is designed to overcome the phenomenon of *internal covariate shift* caused by change of the distribution of layer's inputs while training. This solution integrates normalisation to the network architecture and updates the normalisation parameters while training. As the name batch normalisation suggests, updating parameters happens within each mini-batch. As a results of using BN, the network can be trained faster and better while it also acts as a regulariser such that in some cases Dropout (Section A.1.9) is not needed [196].

A.1.9 Regularisation Techniques

There are a variety of regularisation techniques to prevent overfitting, a phenomenon in which the model is fitted to the training data points but not generalising well to new data. Expanding the training set can be a helpful approach for overcoming this problem. Achieving new data however is a challenging task in most cases. Thus, other solutions are offered to hinder overfitting.

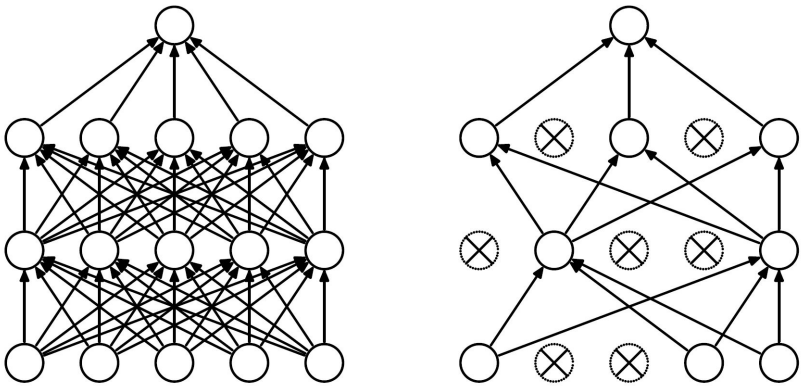
Enforcing Sparsity (Tikhonov Regularisation) Sparsity can be imposed on the distribution of weights by adding a regularisation penalty to the cost function. As a result, large values of the parameters are penalised through the weight decay and some weights go to zero, so only a

few input maps have an effective role in leading to a given output map. Adding a regularisation penalty, the cost function will change according to Equation A.22.

$$J_{new}(\theta) = J(\theta) + \frac{\lambda}{2}\sum_{i=1}^{K}\sum_{j=0}^{n}\theta_{ij}^2. \tag{A.22}$$

Regularisation causes the cost function to be strictly convex. Consequently, obtaining a unique solution is now guaranteed. Moreover, its relevant gradient term should be updated accordingly.

Dropout Suggested by Srivastava *et al.* [20], Dropout as its name conveys refer to randomly dropping a portion of neurons during training to provide better learning and generalisation capability. As an illustration, at each training iteration, with a probability of p single nodes are maintained while the rest of the nodes are dropped with probability $1-p$. This structure prevents the units from excessive co-adaptation and therefore leads to better model generalisation [20]. Figure A.5 illustrates the procedure of Dropout.



(a) Standard Neural Net (b) After applying dropout.

Figure A.5: The Dropout procedure: a) a standard NN b) the resultant reduced network by applying Dropout. Image taken from [20].

A.2 Training Meta Parameters and Further Considerations

There are a variety of possible improvements for training a CNN and therefore several parameters should be taken care of. The depth of network and size of filters can be varied depending on the task, the required amount of complexity and number of output classes.

Regarding the optimisation part, there are other techniques such as Adagrad [206], RMSProp [207], AdaDelta [208] and Adam [209], which can provide faster convergence. Still,

mini-batch gradient descent is being used widely in deep architectures due to its simplicity.

Finally, the choice of the parameters is also crucial. The best choice of parameters however is usually done by trial and error.

In this thesis mainly max pooling and ReLU are implemented as the pooling method and activation function respectively. The Tikhonov regularisation was always used, while dropout and batch normalisation are implemented after Chapter 3 only.

Appendix B

Cross-validation Techniques

In this appendix, two types of cross-validations are offered as a rational way to evaluate and report results, which were used throughout the thesis.

B.1 Cross-validation

Cross-validation is a technique in which available data is divided to train and test sets. The train set is used for training the network and updating its weights while the latter is used for evaluation. Test set can also be used for validation after each epoch of training to monitor whether overfitting is occurring.

When having a k-fold cross-validation, data is randomly partitioned into k sets including equal amount of data samples called folds. Commonly one fold is specified to evaluation set and the $k-1$ folds are used for training. The cross-validation occurs k times such that all the data is used merely once for validation. The k different results are usually averaged leading to a final performance measure. The benefit of cross-validation is that the results are independent of the data arrangements.

As in this thesis there is an interest over the objects to be categorised for grasping, two different validation methods were carried out for verifying the network's robustness and generalisability: Within-object cross-validation (WOC) and between-object cross-validation (BOC).

B.1.1 Within-Object Cross-validation (WOC)

In this cross-validation, the samples included in the test set are images of objects presenting new views of the same objects contained in train set. That is, a portion of available views of an object are *seen* by model when trained on training set and the model is evaluated on the novel

views/orientations of the same object. The results of this validation provides the network's capability in grasp estimating for previously *seen* objects.

B.1.2 Between-Object Cross-validation (BOC)

Contrary to WOC, this type of validation is concerned with the network's potential in recognising *novel* objects. Thus, having n object categories, $n \times \frac{k-1}{k}$ of all the categories are used for training, while the remaining classes $n \times \frac{1}{k}$ are used for model evaluation. In this way, the object classes used for testing are all *unseen* by the model. As in real life, humans can grasp any objects independent of their category and familiarity, this validation provides performance on real world scenarios and therefore is of higher importance.

References

[1] I. S. D. N. Scotland, "The Amputee Statistical Database for the United Kingdom," National Amputee Statistical Database(NASDAB), Tech. Rep., 2005.

[2] J. N. Billock, "Upper limb prosthetic terminal devices: Hands versus hooks," *Clinical Prosthetics and Orthotics*, vol. 10, no. 2, pp. 57–65, 1986.

[3] T. A. Kuiken, L. A. Miller, R. D. Lipschutz, B. A. Lock, K. Stubblefield, P. D. Marasco, P. Zhou, and G. A. Dumanian, "Targeted reinnervation for enhanced prosthetic arm function in a woman with a proximal amputation: a case study," *The Lancet*, vol. 369, no. 9559, pp. 371–380, 2007.

[4] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O. C. Aszmann, "The extraction of neural information from the surface EMG for the control of upper-limb prostheses: emerging avenues and challenges," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 797–809, 2014.

[5] "bebionic features," http://bebionic.com/the_hand/features, accessed: 2015-1-05.

[6] "i-limb ultra," http://www.touchbionics.com/products/active-prostheses/i-limb-ultra, accessed: 2015-12-28.

[7] "Vincent evolution 2: The touch sensing hand prosthesis of next generation." [Online]. Available: https://vincentsystems.de/en/prosthetics/vincent-evolution-2/

[8] "Michelangelo hand: Therapist product brochure." [Online]. Available: http://www.living-withmichelangelo.com/fileadmin/downloads/therapeuten/english/therapist_product_brochure.pdf

[9] M. A. Oskoei and H. Hu, "Myoelectric control systemsa survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.

[10] S. Došen, C. Cipriani, M. Kostić, M. Controzzi, M. C. Carrozza, and D. B. Popović, "Cognitive vision system for control of dexterous prosthetic hands: experimental evaluation," *Journal of Neuroengineering and Rehabilitation*, vol. 7, no. 1, p. 42, 2010.

[11] M. Markovic, S. Dosen, C. Cipriani, D. Popović, and D. Farina, "Stereovision and augmented reality for closed-loop control of grasping in hand prostheses," *Journal of Neural Engineering*, vol. 11, no. 4, p. 046001, 2014.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[13] A. Ng, "Machine learning yearning," *Draft-Version 0.5 ed*, 2016.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[17] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[18] S. Nayar, S. A. Nene, and H. Murase, "Columbia object image library (COIL 100)," *Department of Computer Science, Columbia University, Technical Report. CUCS-006-96*, 1996.

[19] "Supervised Convolutional Neural Network," http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork/, accessed: 2015-01-30.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] A. Ninu, S. Dosen, S. Muceli, F. Rattay, H. Dietl, and D. Farina, "Closed-loop control of grasping with a myoelectric hand prosthesis: Which are the relevant feedback variables

for force control?" *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 5, pp. 1041–1052, 2014.

[22] I. Saunders and S. Vijayakumar, "The role of feed-forward and feedback processes for closed-loop prosthesis control," *Journal of Neuroengineering and Rehabilitation*, vol. 8, no. 60, pp. 1–12, 2011.

[23] C. Light, P. Chappell, B. Hudgins, and K. Engelhart, "Intelligent multifunction myoelectric control of hand prostheses," *Journal of Medical Engineering and Technology*, vol. 26, no. 4, pp. 139–146, 2002.

[24] H. H. Sears and J. Shaperman, "Proportional myoelectric hand control: an evaluation." *American Journal of Physical Medicine and Rehabilitation*, vol. 70, no. 1, pp. 20–28, 1991.

[25] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The Handbook of Brain Theory and Neural Networks*, vol. 3361, p. 310, 1995.

[26] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 3304–3311.

[27] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *2016 Fourth International Conference on 3D Vision (3DV)*. IEEE, 2016, pp. 239–248.

[28] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaíno, A. Eslami, F. Tombari, and N. Navab, "Concurrent segmentation and localization for tracking of surgical instruments," *arXiv preprint arXiv:1703.10701*, 2017.

[29] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager, "Learning in an uncertain world: Representing ambiguity through multiple hypotheses," in *International Conference on Computer Vision (ICCV)*, 2017.

[30] L. V. McFarland, S. L. H. Winkler, A. W. Heinemann, M. Jones, and A. Esquenazi, "Unilateral upper-limb loss: satisfaction and prosthetic-device use in veterans and servicemembers from vietnam and oif/oef conflicts," *Journal of Rehabilitation Research and Development*, vol. 47, no. 4, p. 299, 2010.

[31] K. J. Zuo and J. L. Olson, "The evolution of functional hand replacement: From iron prostheses to hand transplantation," *Plastic Surgery*, vol. 22, no. 1, pp. 44–51, 2014.

[32] A. E. Schultz and T. A. Kuiken, "Neural interfaces for control of upper limb prostheses: the state of the art and future possibilities," *Physical Medicine and Rehabilitation*, vol. 3, no. 1, pp. 55–67, 2011.

[33] P. Geethanjali, "Myoelectric control of prosthetic hands: state-of-the-art review," *Medical Devices (Auckland, NZ)*, vol. 9, p. 247, 2016.

[34] M. Dyson, J. Barnes, and K. Nazarpour, "Myoelectric control with abstract decoders," *Journal of Neural Engineering*, vol. 15, no. 5, 2018.

[35] K. Li, Y. Fang, Y. Zhou, and H. Liu, "Non-invasive stimulation-based tactile sensation for upper-extremity prosthesis: a review," *IEEE Sensors Journal*, vol. 17, no. 9, pp. 2625–2635, 2017.

[36] "The COAPT system," https://www.coaptengineering.com/, accessed: 2018-08-24.

[37] M. Atzori and H. Müller, "Control capabilities of myoelectric robotic prostheses by hand amputees: a scientific research and market overview," *Frontiers in Systems Neuroscience*, vol. 9, p. 162, 2015.

[38] E. A. Biddiss and T. T. Chau, "Upper limb prosthesis use and abandonment: a survey of the last 25 years," *Prosthetics and Orthotics International*, vol. 31, no. 3, pp. 236–257, 2007.

[39] N. Jiang, L. Tian, P. Fang, Y. Dai, and G. Li, "Motion recognition for simultaneous control of multifunctional transradial prostheses," in *Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE*. IEEE, 2013, pp. 1603–1606.

[40] A. J. Young, L. H. Smith, E. J. Rouse, and L. J. Hargrove, "Classification of simultaneous movements using surface EMG pattern recognition," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 5, pp. 1250–1258, 2013.

[41] S. Došen and D. B. Popović, "Transradial prosthesis: artificial vision for control of prehension," *Artificial Organs*, vol. 35, no. 1, pp. 37–48, 2011.

[42] M. Štrbac and M. Marković, "Stereovision system for estimation of the grasp type for electrotherapy," *Serbian Journal of Electrical Engineering*, vol. 8, no. 1, pp. 17–25, 2011.

[43] M. S. Trachtenberg, G. Singhal, R. Kaliki, R. J. Smith, and N. V. Thakor, "Radio frequency identificationan innovative solution to guide dexterous prosthetic hands," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3511–3514.

[44] E. Cho, R. Chen, L.-K. Merhi, Z. Xiao, B. Pousett, and C. Menon, "Force myography to control robotic upper extremity prostheses: a feasibility study," *Frontiers in Bioengineering and Biotechnology*, vol. 4, p. 18, 2016.

[45] V. Ortenzi, S. Tarantino, C. Castellini, and C. Cipriani, "Ultrasound imaging for hand prosthesis control: a comparative study of features and classification methods," in *The 2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2015, pp. 1–6.

[46] W. Guo, X. Sheng, H. Liu, and X. Zhu, "Toward an enhanced human-machine interface for upper-limb prosthesis control with combined EMG and nirs signals," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 4, pp. 564–575, 2017.

[47] A. Fougner, E. Scheme, A. D. Chan, K. Englehart, and Ø. Stavdahl, "A multi-modal approach for hand motion classification using surface EMG and accelerometers," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 4247–4250.

[48] Y. Hao, M. Controzzi, C. Cipriani, D. B. Popovic, X. Yang, W. Chen, X. Zheng, and M. C. Carrozza, "Controlling hand-assistive devices: utilizing electrooculography as a substitute for vision," *IEEE Robotics and Automation Magazine*, vol. 20, no. 1, pp. 40–52, 2013.

[49] A. Gigli, A. Gijsberts, V. Gregori, M. Cognolato, M. Atzori, and B. Caputo, "Visual cues to improve myoelectric control of upper limb prostheses," *arXiv preprint arXiv:1709.02236*, 2017.

[50] F. Giordaniello, M. Cognolato, M. Graziani, A. Gijsberts, V. Gregori, G. Saetta, A.-G. M. Hager, C. Tiengo, F. Bassetto, P. Brugger *et al.*, "Megane pro: myo-electricity, visual and gaze tracking data acquisitions to improve hand prosthetics," in *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2017, pp. 1148–1153.

[51] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific Data*, vol. 1, p. 140053, 2014.

[52] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006. [Online]. Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:No+Title# 0http://soic.iupui.edu/syllabi/semesters/4142/INFO_B529_Liu_s.pdf

[53] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*. Springer Science and Business Media, 2013.

[54] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[55] D. Sueaseenak, S. Wibirama, T. Chanwimalueang, C. Pintavirooj, and M. Sangworasil, "Comparison study of muscular-contraction classification between independent component analysis and artificial neural network," in *Communications and Information Technologies, 2008. ISCIT 2008. International Symposium on*. IEEE, 2008, pp. 468–472.

[56] N. F. Güler and S. Koçer, "Classification of EMG signals using pca and fft," *Journal of Medical Systems*, vol. 29, no. 3, pp. 241–250, 2005.

[57] A. D. Chan and K. B. Englehart, "Continuous myoelectric control for powered prostheses using hidden markov models," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 1, pp. 121–124, 2005.

[58] R. J. Smith, F. Tenore, D. Huberdeau, R. Etienne-Cummings, and N. V. Thakor, "Continuous decoding of finger position from surface EMG signals for the control of powered prostheses," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*. IEEE, 2008, pp. 197–200.

[59] T. Oyama, Y. Mitsukura, S. G. Karungaru, S. Tsuge, and M. Fukumi, "Wrist EMG signals identification using neural network," in *Industrial Electronics, 2009. IECON'09. 35th Annual Conference of IEEE*. IEEE, 2009, pp. 4286–4290.

[60] M. Lei, Z.-Z. Wang, L.-Y. Cai, H.-H. Zhang, and H. Cai, "An EMG classifying method based on bayes' criterion," in *Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE*, vol. 5. IEEE, 1998, pp. 2625–2626.

[61] A. B. Ajiboye and R. F. Weir, "A heuristic fuzzy logic approach to EMG pattern recognition for multifunctional prosthesis control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 280–291, 2005.

[62] J. Peng, D. R. Heisterkamp, and H. Dai, "LDA/SVM driven nearest neighbor classification," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, vol. 1. IEEE, 2001, pp. I–I.

[63] J. Kwon, D. Lee, S. Lee, N. Kim, and S. Hong, "EMG signals recognition for continuous prosthetic arm control purpose," in *1996 IEEE Asia Pacific Conference on Circuits and Systems*. IEEE, 1996, pp. 365–368.

[64] F. Sebelius, L. Eriksson, C. Balkenius, and T. Laurell, "Myoelectric control of a computer animated hand: A new concept based on the combined use of a tree-structured artificial neural network and a data glove," *Journal of Medical Engineering and Technology*, vol. 30, no. 1, pp. 2–10, 2006.

[65] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer learning for sEMG hand gestures recognition using convolutional neural networks," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2017, pp. 1663–1668.

[66] U. Côté-Allard, C. L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette, and B. Gosselin, "Deep learning for electromyographic hand gesture signal classification by leveraging transfer learning," *arXiv preprint arXiv:1801.07756*, 2018.

[67] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers in Neuroscience*, vol. 11, p. 379, 2017.

[68] Y. Yamanoi and R. Kato, "Control method for myoelectric hand using convolutional neural network to simplify learning of EMG signals," in *2017 IEEE International Conference on Cyborg and Bionic Systems (CBS)*. IEEE, 2017, pp. 114–118.

[69] P. Xia, J. Hu, and Y. Peng, "EMG-based estimation of limb movement using deep learning with recurrent convolutional neural networks," *Artificial Organs*, vol. 42, no. 5, pp. E67–E77, 2018.

[70] M. Z. ur Rehman, A. Waris, S. O. Gilani, M. Jochumsen, I. K. Niazi, M. Jamil, D. Farina, and E. N. Kamavuako, "Multiday EMG-based classification of hand motions with deep learning techniques," *Sensors*, vol. 18, p. 2497, 2018.

[71] L. J. Hargrove, E. J. Scheme, K. B. Englehart, and B. S. Hudgins, "Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 18, no. 1, pp. 49–57, 2010.

[72] N. Jiang, S. Dosen, K.-R. Muller, and D. Farina, "Myoelectric control of artificial limbsis there a need to change focus?[in the spotlight]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 152–150, 2012.

[73] N. Jiang, S. Muceli, B. Graimann, and D. Farina, "Effect of arm position on the prediction of kinematics from EMG in amputees," *Medical and Biological Engineering and Computing*, vol. 51, no. 1-2, pp. 143–151, 2013.

[74] C. Cipriani, F. Zaccone, S. Micera, and M. C. Carrozza, "On the shared control of an EMG-controlled prosthetic hand: analysis of user–prosthesis interaction," *IEEE Transactions on Robotics*, vol. 24, no. 1, pp. 170–184, 2008.

[75] R. Tomovic and G. Boni, "An adaptive artificial hand," *IRE Transactions on Automatic Control*, vol. 7, no. 3, pp. 3–10, 1962.

[76] J. Nightingale, "Microprocessor control of an artificial arm," *Journal of Microcomputer Applications*, vol. 8, no. 2, pp. 167–173, 1985.

[77] A. Krasoulis, I. Kyranou, M. S. Erden, K. Nazarpour, and S. Vijayakumar, "Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements," *Journal of Neuroengineering and Rehabilitation*, vol. 14, no. 1, p. 71, 2017.

[78] I. Kyranou, A. Krasoulis, M. S. Erden, K. Nazarpour, and S. Vijayakumar, "Real-time classification of multi-modal sensory data for prosthetic hand control," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 536–541.

[79] M. Markovic, S. Došen, D. Popovic, B. Graimann, and D. Farina, "Sensor fusion and computer vision for context-aware control of a multi degree-of-freedom prosthesis," *Journal of Neural Engineering*, vol. 12, no. 6, p. 066022, 2015.

[80] Đ. Klisić, M. Kostić, S. Došen, and D. B. Popović, "Control of prehension for the transradial prosthesis: natural-like image recognition system," *Journal of Automatic Control*, vol. 19, no. 1, pp. 27–31, 2009.

[81] M. C. Carrozza, G. Cappiello, S. Micera, B. B. Edin, L. Beccai, and C. Cipriani, "Design of a cybernetic hand for perception and action," *Biological Cybernetics*, vol. 95, no. 6, p. 629, 2006.

[82] M. Štrbac and M. Marković, "Stereovision system for estimation of the grasp type for electrotherapy," *Serbian Journal of Electrical Engineering*, vol. 8, no. 1, pp. 17–25, 2011.

[83] D. Forsyth and J. Ponce, *Computer vision: a modern approach*, illustrate ed., Jean Ponce, Ed. Prentice Hall, 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=580035

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

[85] S. Haykin, *Neural networks and learning machines*. Prentice Hall, 2009. [Online]. Available: https://cise.ufl.edu/class/cap6615sp12/syllabus.pdf

[86] M. S. Nixon and A. S. Aguado, *Feature extraction and image processing for computer vision*. Academic Press, 2012.

[87] P. M. Roth and M. Winter, "Survey of appearance-based methods for object recognition," *Inst. for Computer Graphics and Vision, Graz University of Technology, Austria, Technical Report ICGTR0108 (ICG-TR-01/08)*, 2008.

[88] C. Harris and M. Stephens, "A combined corner and edge detector." in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[89] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–534, 1997.

[90] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[91] D. G. Lowe, "Object recognition from local scale-invariant features," in *The Proceedings of the 1999 Seventh IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 1999, pp. 1150–1157.

[92] ——, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[93] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*. Springer, 2006, pp. 404–417.

[94] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2005, pp. 886–893.

[95] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Advances in Neural Information Processing Systems*, 2012, pp. 2222–2230.

[96] V. Nair and G. E. Hinton, "3D object recognition with deep belief nets," in *Advances in Neural Information Processing Systems*, 2009, pp. 1339–1347.

[97] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 2146–2153.

[98] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, "Learning convolutional feature hierarchies for visual recognition," in *Advances in Neural Information Processing Systems*, 2010, pp. 1090–1098.

[99] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8595–8598.

[100] Y. LeCun, "Learning invariant feature hierarchies," in *Workshops and Demonstrations in 2012 European Conference on Computer Vision (ECCV)*. Springer, 2012, pp. 496–505.

[101] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, p. e2, 2014.

[102] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and Cooperation in Neural Nets*. Springer, 1982, pp. 267–285.

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California University of San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[104] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[105] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[106] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man and Cybernetics*, no. 5, pp. 826–834, 1983.

[107] G. Orchard, J. G. Martin, R. J. Vogelstein, and R. Etienne-Cummings, "Fast neuromimetic object recognition using FPGA outperforms GPU implementations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1239–1252, 2013.

[108] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[109] "The mnist database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[110] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[111] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[112] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[113] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2014, pp. 512–519.

[114] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[115] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2013, pp. 3626–3633.

[116] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[117] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[118] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[119] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2017.

[120] Y. Li, K. He, J. Sun *et al.*, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[121] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2015.

[122] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 2980–2988.

[123] G. Kootstra, M. Popović, J. A. Jørgensen, K. Kuklinski, K. Miatliuk, D. Kragic, and N. Krüger, "Enabling grasping of unknown objects through a synergistic use of edge and surface information," *The International Journal of Robotics Research*, vol. 31, no. 10, pp. 1190–1213, 2012.

[124] M. Kopicki, R. Detry, M. Adjigble, R. Stolkin, A. Leonardis, and J. L. Wyatt, "One-shot learning and generation of dexterous grasps for novel objects," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 959–976, 2016.

[125] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[126] J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1316–1322.

[127] D. Guo, F. Sun, H. Liu, T. Kong, B. Fang, and N. Xi, "A hybrid deep architecture for robotic grasp detection," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017.

[128] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *arXiv preprint arXiv:1603.02199*, 2016.

[129] B. S. Zapata-Impata, "Using geometry to detect grasping points on 3d unknown point cloud," in *International Conference on Informatics in Control, Automation and Robotics*, 2017, pp. 154–161.

[130] B. Kehoe, S. Patil, P. Abbeel, and K. Goldberg, "A survey of research on cloud robotics and automation," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398–409, 2015.

[131] U. Viereck, A. Pas, K. Saenko, and R. Platt, "Learning a visuomotor controller for real world robotic grasping using simulated depth images," in *Conference on Robot Learning*, 2017, pp. 291–300.

[132] J. Varley, C. DeChant, A. Richardson, A. Nair, J. Ruales, and P. Allen, "Shape completion enabled robotic grasping," *arXiv preprint arXiv:1609.08546*, 2016.

[133] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics and Automation Magazine*, 2004.

[134] U. Asif, M. Bennamoun, and F. A. Sohel, "Rgb-d object recognition and grasp detection using hierarchical cascaded forests," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 547–564, 2017.

[135] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[136] Z. Wang, Z. Li, B. Wang, and H. Liu, "Robot grasp detection using multimodal deep convolutional neural networks," *Advances in Mechanical Engineering*, vol. 8, no. 9, p. 1687814016668077, 2016.

[137] S. Kumra and C. Kanan, "Robotic grasp detection using deep convolutional neural networks," *arXiv preprint arXiv:1611.08036*, 2016.

[138] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings of 2000 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2000, pp. 348–353.

[139] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[140] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders, "The amsterdam library of object images," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 103–112, 2005.

[141] F. J. Huang, Y.-L. Boureau, Y. LeCun *et al.*, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *The 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[142] K. Nazarpour, A. Barnard, and A. Jackson, "Flexible cortical control of task-specific muscle synergies," *Journal of Neuroscience*, vol. 32, no. 36, pp. 12 349–12 360, 2012.

[143] T. Pistohl, C. Cipriani, A. Jackson, and K. Nazarpour, "Abstract and proportional myoelectric control for multi-fingered hand prostheses," *Ann. Biomed. Eng.*, vol. 41, pp. 2687–2698, 2013.

[144] S. Graziadio, K. Nazarpour, S. Gretenkord, A. Jackson, and J. A. Eyre, "Greater intermanual transfer in the elderly suggests age-related bilateral motor cortex activation is compensatory," *Journal of Motor Behavior*, vol. 47, no. 1, pp. 47–55, 2015.

[145] T. Pistohl, D. Joshi, G. Ganesh, A. Jackson, and K. Nazarpour, "Artificial proprioceptive feedback for myoelectric control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 3, pp. 498–507, 2015.

[146] C. M. Harris and D. M. Wolpert, "Signal-dependent noise determines motor planning," *Nature*, vol. 20, pp. 780–784, 1998.

[147] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[148] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems*, 2014, pp. 3320–3328.

[149] F. Seide and A. Agarwal, "Cntk: Microsoft's open-source deep-learning toolkit," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, pp. 2135–2135.

[150] Touch bionics: i-mo. [Online]. Available: http://www.touchbionics.com

[151] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific Data*, vol. 1, p. 140053, 2014.

[152] R. N. Khushaba, A. Al-Timemy, S. Kodagoda, and K. Nazarpour, "Combined influence of forearm orientation and muscular contraction on EMG pattern recognition," *Expert Systems with Applications*, vol. 61, pp. 154–161, 2016.

[153] A. Krasoulis, S. Vijayakumar, and K. Nazarpour, "Evaluation of regression methods for the continuous decoding of finger movement from surface EMG and accelerometry," in *2015 7th International IEEE/EMBS Conference on Neural Engineering (NER)*. IEEE, 2015, pp. 631–634.

[154] I. Kyranou, A. Krasoulis, M. S. Erden, K. Nazarpour, and S. Vijayakumar, "Real-time classification of multi-modal sensory data for prosthetic hand control," in *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE, 2016, pp. 536–541.

[155] M. S. Trachtenberg, G. Singhal, R. Kaliki, R. J. Smith, and N. V. Thakor, "Radio frequency identificationAn innovative solution to guide dexterous prosthetic hands," in *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. IEEE, 2011, pp. 3511–3514.

[156] N. E. Krausz, T. Lenzi, and L. J. Hargrove, "Depth sensing for improved control of lower limb prostheses," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 11, pp. 2576–2587, 2015.

[157] G. Ghazaei, A. Alameer, P. Degenaar, G. Morgan, and K. Nazarpour, "An exploratory study on the use of convolutional neural networks for object grasp classification," in *2nd IET International Conference on Intelligent Signal Processing*, 2015, pp. 1–5.

[158] T. N. Sainath, B. Kingsbury, A. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, "Improvements to deep convolutional neural networks for LVCSR," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 315–320.

[159] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4724–4732.

[160] A. Bulat and G. Tzimiropoulos, "Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[161] D. Merget, M. Rock, and G. Rigoll, "Robust facial landmark detection via a fully-convolutional local-global context network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[162] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[163] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[164] V. Belagiannis and A. Zisserman, "Recurrent human pose estimation," *arXiv preprint arXiv:1605.02914*, 2016.

[165] A. Bulat and G. Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 717–732.

[166] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[167] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[168] X. Du, T. Kurmann, P.-L. Chang, M. Allan, S. Ourselin, R. Sznitman, J. D. Kelly, and D. Stoyanov, "Articulated multi-instrument 2d pose estimation using fully convolutional networks," *IEEE Transactions on Medical Imaging*, 2018.

[169] S. Lee, S. P. S. Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra, "Stochastic multiple choice learning for training diverse deep ensembles," in *Advances in Neural Information Processing Systems*, 2016, pp. 2119–2127.

[170] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[171] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3431–3440.

[172] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-assisted Intervention*. Springer, 2015, pp. 234–241.

[173] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017, pp. 1175–1183.

[174] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[175] A. Vedaldi and K. Lenc, "Matconvnet – convolutional neural networks for matlab," in *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

[176] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*. ACM, 2008, pp. 160–167.

[177] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[178] Y. Zhang and Q. Yang, "A survey on multi-task learning," *arXiv preprint arXiv:1707.08114*, 2017.

[179] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *International Conference on Machine Learning (ICML) Deep Learning Workshop*, vol. 2, 2015.

[180] J. DeGol, A. Akhtar, B. Manja, and T. Bretl, "Automatic grasp selection using a camera in a hand prosthesis," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*. IEEE, 2016, pp. 431–434.

[181] S. Makridakis, "Accuracy measures: theoretical and practical concerns," *International Journal of Forecasting*, vol. 9, no. 4, pp. 527–529, 1993.

[182] G. Ghazaei, A. Alameer, P. Degenaar, G. Morgan, and K. Nazarpour, "Deep learning-based artificial vision for grasp classification in myoelectric hands," *Journal of Neural Engineering*, vol. 14, no. 3, p. 036025, 2017.

[183] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *arXiv preprint arXiv:1612.00593*, 2016.

[184] G. Ghazaei, F. Tombari, and K. Nazarpour, "Grasp type estimation for myoelectric prostheses using point cloud feature learning," *Workshop on Human Aiding Robotics, IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[185] X. Zhu, "Semi-supervised learning literature survey," *Computer Science, University of Wisconsin-Madison*, vol. 2, no. 3, p. 4, 2006.

[186] H. Mobahi, R. Collobert, and J. Weston, "Deep learning from temporal coherence in video," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 737–744.

[187] W. Y. Zou, A. Y. Ng, and K. Yu, "Unsupervised learning of visual invariance with temporal coherence," in *2011 Workshop on Deep Learning and Unsupervised Feature Learning in Neural Information Processing Systems (NIPS)*, 2011.

[188] S. M. Kakade and D. P. Foster, "Multi-view regression via canonical correlation analysis," in *Learning Theory*. Springer, 2007, pp. 82–96.

[189] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," in *Machine Learning*, 1992, pp. 229–256.

[190] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 1998.

[191] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of Artificial Intelligence Research*, pp. 237–285, 1996.

[192] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton, "Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning," in *2011 IEEE International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2011, pp. 1–7.

[193] A. L. Edwards, M. R. Dawson, J. S. Hebert, R. S. Sutton, K. M. Chan, and P. M. Pilarski, "Adaptive switching in practice: Improving myoelectric prosthesis performance through reinforcement learning," *In Proceedings of the Myoelectric Controls Symposium*, vol. 14, pp. 18–22, 2014.

[194] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[195] J. Bouvrie, "Notes on convolutional neural networks," 2006, technical report. [Online]. Available: http://cogprints.org/5869/

[196] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[197] J. van Doorn, "Analysis of deep convolutional neural network architectures," 2014.

[198] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, vol. 15, 2011, pp. 315–323.

[199] D. Scherer, A. Müller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Artificial Neural Networks–ICANN 2010*. Springer, 2010, pp. 92–101.

[200] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *arXiv preprint arXiv:1301.3557*, 2013.

[201] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[202] "Softmax regression," http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/, accessed: 2015-01-29.

[203] A. Cauchy, "Méthode générale pour la résolution des systemes déquations simultanées," *Comptes Rendus Seances Academy of Science Paris*, vol. 25, no. 1847, pp. 536–538, 1847.

[204] H. Robbins and S. Monro, "A stochastic approximation method," *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

[205] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, 1999.

[206] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[207] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[208] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[209] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.