الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria

وزارة التعليم العالي و البحث العلمي

Ministry of Higher Education and Scientific Research

جامعة عبد الحميد ابن باديس – مستغانم

Abdelhamid Ibn Badis University of Mostaganem

THESE

Presented at the

FACULTY OF SCIENCE AND TECHNOLOGY
ELECTRICAL ENGINEERING DEPARTMENT
Signals and Systems Laboratory

For graduation

Third-Cycle Degree
DOCTORATE

Option : Electronics of Embedded system

By

BOUSBAI Khalil

Identification of Hand Gestures With Camera Networks

Defended on : 16 /06/2022 before the examination board

Jury President :	YAGOUBI Benabdellah	Pr	Mostaganem University
Examiners :	DAHMANI Mohammed	Pr	USTO University
	HENNI Sid Ahmed	MCA	Mostaganem University
	OULD MAMMAR Madani	MCA	Mostaganem University
Supervisor :	MERAH Mostefa	Pr	Mostaganem University

ABSTRACT

Human-Computer Interaction (HCI) is a broad field involving different types of interactions including gestures. A system may be utilized to identify human gestures to convey information for device control. The aim of gesture recognition is to record gestures that are formed in a certain way and then detected by a device such as a camera. Hand gestures can be used by people who possess different disabilities, including those with hearing impairments, speech impairments, and stroke patients, to communicate and fulfill their basic needs, Hand gestures offer people a convenient way to interact with computers, in addition to giving them the ability to communicate without physical contact and at a distance, which is essential in today's health conditions, especially during an epidemic infectious viruses such as the COVID-19 coronavirus.

Various studies have previously been conducted relating to hand gestures. Some studies proposed different techniques to implement the hand gesture experiments. For image processing, there are multiple tools to extract features of images, as well as Artificial Intelligence which has varied classifiers to classify different types of data. This research discusses this issue using different algorithms. To detect hand gestures, in this work we use Convolutional Neural Networks (CNN) and Capsule Networks (CapsNet) to extract images features. Next, in order to reduce the dimensionality data, we have used the Principal Component Analysis (PCA), finally to classify the American sign language (ASL) gestures we have chosen the Support Vector Machine (SVM) classifier.

Keywords : Hand gesture recognition, American Sign Language, Deep Learning, Capsule Networks, Convolutional Networks.

ملخص

يعد التفاعل بين الإنسان والحاسوب مجالًا واسعًا يتضمن أنواعًا مختلفة من التفاعلات بما في ذلك الإيماءات. يمكن استخدام نظام لتحديد الإيماءات البشرية لنقل المعلومات للتحكم في الجهاز. الهدف من التعرف على الإيماءات هو تسجيل الإيماءات التي يتم تشكيلها بطريقة معينة ثم يتم اكتشافها بواسطة جهاز مثل الكاميرا. يمكن استخدام إيماءات اليد من قبل الأشخاص الذين يعانون من إعاقات مختلفة ، بما في ذلك أولئك الذين يعانون

من ضعف السمع وضعف الكلام ومرضى السكتات الدماغية ، للتواصل وتلبية احتياجاتهم الأساسية ، توفر إيماءات اليد للأشخاص طريقة ملائمة للتفاعل مع أجهزة الكمبيوتر ، بالإضافة إلى منحها لهم القدرة على التواصل دون اتصال جسدي وعن بعد ، وهو أمر ضروري في الظروف الصحية الحالية ، خاصة أثناء انتشار الفيروسات المعدية الوبائية مثل فيروس كورونا ١٩. سبق إجراء دراسات مختلفة تتعلق بحركات اليد. اقترحت بعض الدراسات تقنيات مختلفة لتنفيذ تجارب إيماءات اليد. لمعالجة الصور ، توجد أدوات متعددة لاستخراج ميزات الصور ، بالإضافة إلى الذكاء الاصطناعي الذي يحتوي على مصنفات متنوعة لتصنيف أنواع مختلفة من البيانات. يناقش هذا البحث هذه المسألة باستخدام خوارزميات مختلفة. لاكتشاف إيماءات اليد ، نستخدم في هذا العمل الشبكات العصبية التلافيفية وشبكات الكبسولة لاستخراج ميزات الصور. بعد ذلك ، لتقليل بيانات الأبعاد ، استخدمنا تحليل المكونات الرئيسية ، وأخيراً لتصنيف إيماءات لغة الإشارة الأمريكية اخترنا مصنف (س في م) .

الكلمات الرئيسية: التعرف على إيماءات اليد ، لغة الإشارة الأمريكية ، التعلم العميق ، الشبكات الكبسولة ، الشبكات التلافيفية.

Résumé

L'interaction homme-machine (HCI) est un vaste domaine impliquant différents types d'interactions, y compris les gestes. Un système peut être utilisé pour identifier des gestes humains afin de transmettre des informations pour la commande de dispositif. Le but de la reconnaissance gestuelle est d'enregistrer des gestes formés d'une certaine manière puis détectés par un appareil tel qu'une caméra. Les gestes de la main peuvent être utilisés par des personnes souffrant de différents handicaps, y compris les malentendants, les troubles de la parole et les patients victimes d'AVC, pour communiquer et répondre à leurs besoins de base. Les gestes de la main offrent aux gens un moyen pratique d'interagir avec les ordinateurs, en plus de leur donner la capacité de communiquer sans contact physique et à distance, ce qui est essentiel dans les conditions sanitaires d'aujourd'hui, en particulier lors d'une épidémie de virus infectieux comme le coronavirus COVID-19.

Diverses études ont déjà été menées concernant les gestes de la main. Certaines études ont proposé différentes techniques pour mettre en œuvre les expériences de geste de la main. Pour le traitement d'image, il existe plusieurs outils pour extraire les caractéristiques des images, ainsi que l'intelligence artificielle qui dispose de divers clas-

sificateurs pour classer différents types de données. Cette recherche aborde cette question en utilisant différents algorithmes. Pour détecter les gestes de la main, dans ce travail, nous utilisons les réseaux de neurones convolutifs (CNN) et les réseaux de capsules (CapsNet) pour extraire les caractéristiques des images. Ensuite, afin de réduire les données de dimensionnalité, nous avons utilisé l'analyse en composantes principales (ACP), enfin pour classer les gestes de la langue des signes américaine (ASL) nous avons choisi le classificateur Support Vector Machine (SVM).

Mots clés : Reconnaissance des gestes de la main, langue des signes américaine, apprentissage en profondeur, réseaux de capsules, réseaux convolutifs.

ACKNOWLEDGEMENT

We must first, thank Almighty God, for all will, determination and patience. He has given us; enabling us to do this work.

The completion of this work, which took place over several years, is a subject of great satisfaction. It's an opportunity to remember the different pitfalls I had to overcome, especially the people who helped me get there. First of all, I thank my parents who believed in me and gave me all their help when I needed it. This thesis is 200% dedicated to them. I also thank all the members of my family who have contributed directly or indirectly to what I have become. We also thank the president and the members of the jury who did me the honour of studying our work attentively and who had the honour of defending our thesis.

I would like to thank my thesis supervisor, Pr MERAH Mostefa, for his countless advice, patience and support throughout his doctoral journey. I have learned a lot by working directly with him and I receive my sincere thanks from him. His confidence in my abilities allowed me to do a large part of my creative work, and I thank him for all the patience and availability he has shown me. I would also like to thank Dr Abed Mansour, Dr Ould Mammar Madani and Dr Bentoumi Mohamed for their helpful advice and participation in my committees.

This research would not have been possible without the support of Professor Jose´-Luis Sancho-Go´ mez and of Dr Morales-Sanchez Juan of the Polytechnic University of Cartagena, Spain. We warmly thank all the members of the TDAM (Data Processing and Ma- chine Learning) research team and all those who worked with me on the project at the Polytechnic University of Cartagena. I would like to thank all my friends in Cartagena, Spain for their friendship and support during the PhD trip to all my friends Dahmani Abderraouf, Iqbal aityala.

I would also like to thank all my friends and colleagues of the LSS Lab research team in Mostaganem for their help, support and kindness. I would also like to thank them for the confidence they have shown in me by allowing me to participate in the collective life of the team. The human relationships that I have enjoyed having created real bonds of friendship which are invaluable to me. May they all be assured of my deep gratitude. In particular, I would like to express my gratitude to Mohammad chouai and Larbi Arzaki

to my Lord and to Kadda Djebbar for all the help they have given me.

A special dedication to Mohammad Chouai, to tell him that I am very grateful for what you have done for me. Thank you ... it's a very simple word. What I would like to express is above that ... and I cannot thank you enough. I would also like to thank all my wonderful friends whom I had the honour of meeting at the National Polytechnic School in Oran, and who was able to give me confidence and listen to me at all times. Of course this list is not exhaustive and I thank all those who know me and allow me to feel that I exist ... Thank you all.

Place: Mostaganem

Date: 20/06/2022 **Khalil Bousbai**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

List of Acronyms

Acronym : **Stands for**

AI : artificial intelligence

VR : Virtual reality

HGR : Hand Gesture Recognition

HCI : human-computer interface

SLR : Sign Language Recognition

ASL : American Sign Language

DL : Deep Learning

CNN : convolutional neural network

CapsNet : capsule network

RGB : Red Green and Blue

IR : infrared

ToF : Time-of-flight

3D : Three-Dimensional

2D : Two-Dimensional

HMM : Hidden Markov Models

VGA : Visual graphics array

QVGA : Quarter Video Graphics Array

SDK : software development kit

HPR : Hand posture recognition

HGS : hand gesture signature

USB : universal serial bus

RGB-D : simply a combination of a RGB image and its corresponding depth image

ANNs : artificial neural networks

NN : neural networks

TIMIT : is a corpus of phonemically and lexically transcribed speech of American English speaker

ILSVRC : ImageNet Large Scale Visual Recognition Challenge

GPUs : Graphics processing unit

HSV : hue, saturation, lightness and HSV are alternative representations of the RGB

color model

YCbCr : a technique of color spaces used for digital video and photography systems

Camshift : Continuously Adaptive Mean Shift

HoG : Histogram of Gradient

SIFT : Scale Invariant Feature Transform

LoG : Laplace of Gaussian

SURF : Speeded-Up Robust Features

BRIEF : Binary Robust Independent Elementary Feature

ORB : Oriented Fast and Rotated BRIEF

BRISK : Binary Robust Invariant Scalable Keypoints

FREAK : Fast Retina Keypoint

CDP : Continuous Dynamic Programming

DTW : Dynamic Time Warping

MCDNN : Multi column Deep Neural Networks

API : application programming interface

URLs : unique identifier used to locate a resource on the Internet

MLP : multilayer perceptron

RBF : radial basis function

SVM : support vector machine

OCR : Optical Character Recognition

ANPR : Automatic Number Plate Recognition

RNN : Recurrent Neural Network

DNN : Deep Neural Network

ReLU : restricted Linear Unit

MRI : Magnetic Resonance Imaging

CT : Computed Tomography

SPECT : Single-Photon Emission Computed Tomographic

PCA : Principle Component Analysis

EMD : empirical mode decomposition

WT : Wavelet Transformsare

TDNN : Time Delay Neural Network

HDTV : High-Definition Television

MLA : Microlens array

MIT : Massachusetts Institute of Technology

CCD : charge-coupled device

CMOS : is a fabrication technology for semiconductor systems

ADC : analog-to-digital converter

HTS : Hand tracking and segmentation

SOMs : Self Organizing Maps

RBMs : Restricted Boltzmann Machines

CD : Contrastive Divergence

ML : Maximum- Likelihood

UCLA : University of California, Los Angeles

DA : Data Augmentation

Introduction

In recent years, with the development of artificial intelligence (AI), various types of controllers other than a mouse and keyboard have become popular. With the rapid growth of hardware computing power and the precision of machine learning methods, there is an urgent need for more accessible and effective ways to interact with computers, and one form of interaction that has gained popularity is the field of human-machine interactions. Also, called Contact and Collaborate Offline, this contactless cooperation, through a qualified and acceptable HCI interface, will push the limits of HCI, especially in the event of infectious epidemics such as the outbreak of the coronavirus COVID 19. Virtual reality (VR) environments such as web browsers, video games and a variety of tools have benefited from human-computer interaction, and given the richness of gesture expression and the continuous improvement of computer vision technology, hand gesture recognition based on computer vision has practical potential in the following aspects:

1. Home entertainment: TV and game products built into the gesture recognition system can enable users to move, switch, confirm, exit, etc.

2. Intelligent driving: the car's navigation and information system control can reduce driver distraction by introducing the gesture recognition system.

3. Smart wear devices: the natural human-computer interaction with the head-mounted VR screen as a means can enhance the user's immersion in the virtual environment.

4. Sign language recognition: The Interpreter System can recognize and use sign language for communication between the deaf and the outside world.

5. Automated Communication: The robot can understand and respond to the user's actions and communicate with the user [1].

Gesture Recognition is a topic in engineering science and language technology with the goal of interpreting human gestures via mathematical algorithms, and can be seen as a way for computers to understand human body language. These gestures are commonly originated from face or hands motion. The Hand Gesture Recognition (HGR) can

be applied to the Sign Language understanding. Sign languages are expressed through hand articulations in combination with other gestures or movements. Currently many native sign languages coexist worldwide, and everyone has their own grammar and lexicon, whereby they are not mutually intelligible.

Achieving user efficiency and accuracy in the field of human-machine interactions is the goal of this type of research. The natural reaction process requires the use of the user's body without any additional hardware. This is known as a human-computer interface (HCI).

By using sensors and capturing the various interactions of the body, we can recognize commands and perform the required tasks in the system. HCI is the use of body and hand gestures, and thus hand gesture recognition (HGR) is a major component of this type of system. Smartphones and tablets are controlled by touching the device and making gestures. Another method of HCI is to use cameras as sensors to identify different body parts of a user, such as the Kinect sensor for Xbox One. The user's hands can have a set of gestures, so if the gesture recognition is done, this will allow us to control the applications. The segmentation of hands by cameras and the identification of the gesture made by scientists leads to an area in the research community known as: Sign Language Recognition (SLR).

Usually, non-mute people do not learn sign language, and thus, this segment of society has problems of communicating.When we are dealing with someone from this segment, it will be very difficult. For example, when interviewing one of these people, you have to find a translator, and that causes some problems. Here, image recognition technologies play an important role by automating the tag translation process. This paper focuses on this kind of problem: Facilitating communication via SL by automating the transcription of the source language without reference to the human translator.

Although SLR research with machines started a long time ago, but due to the challenges that exist, no automatic sign language transcription system has been found.Dealing with current issues (number of gestures, differences in sizes, hand position ...) is not easy. This is one of the main reasons why leaders of previous studies have focused on limited databases of gestures and users.
The aim of this work is to study the performance of neural networks that recognize and transmit textual SL images (hand gestures). Because of the broad scope of this task, the scope of the study was limited to American Sign Language (ASL) symbols.
In this work, we use Deep Learning (DL) techniques to solve HGR problems. DL can be included in a broader family, specifically the Machine Learning algorithms, but with

a noticeable increase of the layers and capacity in the network. Machine Learning was inspired by the structure and performance of the brain, given place to the so-called artificial neural networks. The ability to handle an outsized number of features makes DL very powerful for representation learning and unstructured data. However, DL algorithms would be overrated to resolve less complex problems, because they require access to massive amounts of information to be effective.

Diversity is a very important concept in machine learning because, in addition to increasing accuracy, it also produces a greater capacity of generalization acting as a regularization element of learning. Diversity can be achieved in several ways. For example, it can be introduced in the training data, such that it provides more discriminative information for the model. Diversity can also be obtained by the combination of the outputs of a model trained several times. A third option for introducing diversity is to combine the outputs of several types of models. This method is called ensemble learning [2]. Ensemble learning was initially used for classification [3] although it can also be used in other fields as regression [4] and feature selection [5] among others. The outputs of models can be combined in different manners as using maxvoting, averaging, weighted averaging or other advanced techniques as stacking, blending, bagging or boosting.

In this research, we demonstrated the efficiency of a specific set of machine learning that combines feature spaces of many deep machines to solve an HGR problem. In particular, the group consists of the distinct spaces of a standard convolutional neural network (CNN) and a capsule network (CapsNet). Diversity is also introduced into the data set by the data augmentation procedure. This regulation technique has been shown to be essential to obtain the best possible result on the MNIST dataset of American Sign Language (ASL).
The rest of this document is organized as follows. In the first chapter, I present studies on previous work in the field of recognizing human body movements and gestures and recognizing methods of self-learning. In the second chapter, I talk about Image Processing and Pattern Recognition and some applications in this field. In Chapter Three, a review of Gesture Recognition and Types of Gesture Recognition with Overview of Types of Cameras. In Chapter Four, a review of the Image Acquisition, Data Preprocessing, segmentation, Machine Learning Approach to form the proposed group to solve the image recognition problem. In Chapter Five is presented some experimental results obtained by several deep tools are compared with those presented by our proposed method. Finally, it closes by providing the conclusions of this work and suggesting some future lines of research.

CHAPTER 1

Review of Literature

1.1 Introduction

Among the most important areas in today's life is communication - reading and writing. The method of communication in which he is involved in any type of body movement is called gestures. Gesture recognition is a mathematical interpretation by a computer. Gestures are expressive and meaningful bodily movements that involve bodily movements of the fingers, hands, arms, head, face, or body. There are many types of human gesture expressions, the most common being the expression of gestures. In other words, a gesture is a silent or non-vocal method of communication that uses hand movement, different positions on the body, and facial expressions. Gesture recognition based on computer vision has gradually become a hot research destination in the field of human-computer interaction. Sign language is the most expressive method for the hearing impaired, the recognition tool must be able to recognize the vocabulary of continuous signs in real time. Direction sensor-based gesture recognition is an emerging area of pattern recognition research. A trial inspection proves the high performance and accuracy of any proposed device. The usual way to use the hand gesture recognition system is when we give some commands to our system by manual gestures, the device first picks up our command as an image and then compares it to the database and if you find any image in the database, the task assigned to that will be executed. Most important in today's life is communication - reading and writing. They feel difficult to communicate because they cannot access the computer. A few papers have focused on human-computer interaction (HCI). They used braille scripts for reading and writing purposes, which cannot be interpreted by existing computers. The six fingers represent the six points in braille. A smart camera can be defined as a vision system that produces a high-level understanding of the captured scene and generates application-specific data for use in the system. Skin tone segmentation techniques based on monovision are used to divide the hand into a complex image sequence.The features of the standard graph are extracted along with the 44 various engineering features.

The main aspects that determine the complexity of the HGR task in various contexts include:

1. Vocabulary structure (vocabulary size, similarity and complexity of gestures, double-hand gestures),

2. Scene settings (lighting conditions, background content, and blockages),

3. Performance limitations (continuous gestures, face / hand overlapping, hand ejecting from the scene, and pause during gesture),

4. Contrast within categories (gesture size, speed, location, and direction). Under different conditions, the task of HGR can be somewhat varied.

In the first half of this chapter, the different definitions of HPR and HGR will be categorized in different contexts. Since this thesis is primarily focused on appearance-based HGR methods, the second half of this chapter is devoted to reviewing the features and other related works in the category of HGR methods based on 2D computer vision.

1.2 Capture

There are diverse methods and sensors for capturing information from a scene: colour cameras (RGB), infrared cameras (IR or thermal cameras), depth sensors, lasers,reflective markers, etc. Gathering depends on the sensor type data and is appropriate for some specific tasks and not for others. As an example, in an assembly line which produces water in metal bottles, infrared cameras can detect fluids inside the bottles while colour cameras cannot, and therefore when testing if the containers are full, infrared cameras will provide the necessary information.
Since this study is focused on Sign Language (SL), the information of the scene is limited to the hands. To capture hand data? a special sensor has been developed to recognize the position of the hand skeleton,[6] controller; compared to other sensors, it is small and portable.It is sometimes attached to a Head Mounted Display to add the user's hand to a virtual world. Because of physical device constraints, the capture area is limited to a modest distance from the sensor; the area of capture restricts the user's movement and it may not be appropriate for sign language as stated in [7]. Example applications include: [8] is an interface for Human-Computer interaction on Windows and Mac devices, it provides shortcuts for switching applications, volume handling, media player actions, etc.; Cyber Science [9] challenges the user to explore, dissect and assemble a skull to learn about human anatomy.
Hand tracking gloves and markers on the hand are other techniques to capture and track finger information for the recognition of hand gestures [10, 11]. Gloves are accurate in determining the exact position of each finger but they are not comfortable for users, they can obstruct fingers' movements and the resulting hand gesture may not be correct.

Markers are not uncomfortable but they need a specific setup step and a constructed environment. Because casual users of the system may not have in possession the required equipment or they may not know the environment arrangement, this approach may not be suitable for the problem.

I focused on capturing images from the scene, the distance from the camera to each.

(a) Leap motion sensor. (b) Hand glove for hand gesture recognition.

Fig. 1.1 Specific sensors for hand gesture

Point of the hand of the user. The camera captures an image of the scene in which each pixel measures the distance to the closest object. Depth provides features imaging cameras cannot offer, for example, finger positions; colour images are 2D images in which each pixel represents the intensity of the light of each colour channel at that point, while depth sensors capture spatial distance points. Falling into the category of depth sensors there are different methods of capturing distance information of a scene. The most common ones are Time-of-Flight sensors [12], Structured light cameras/sensors [13], and Stereoscopic cameras [14].

1.2.1 Time-of-flight (ToF)

ToF cameras (Figure 2.4 [15]) measure the time a light signal spends flying in the space between the camera and the object in front of it. For each pixel of the image, there is a measurement which provides information about the depth of an object. These types

(a) Time of flight. (b) Kinect camera. (c) Stereoscopic camera.

Fig. 1.2 Types of Sensor.

of cameras produce accurate depth images at a high frame rate (150 frames per second) empowering real-time recognition systems, but they are sensitive to lighting conditions and reflective surfaces. The resolution of the ToF device is lower than other sensors [12]. Low-resolution cameras may not fit the requirements of the SL problem because it does not properly detect dissimilarities in hand images or the hand.May not fit within the boundaries of the image. Using more than one ToF sensor to increase the resolution of the image (stitching images together) increases the resolution but it produces noise in the estimation of the depth as a result of the interferences between them. In spite of the low resolution of the cameras, some work has been done using them for hand recognition [16, 17] obtaining accuracy around 90% in real time.

1.2.2 Structured light

The principle of structured light is based on projecting a specific light pattern into a scene. An infrared (IR) light emitter projects a dense structured matrix of dots (IR11 pattern) onto the scene; these projections are captured by an infrared camera. As shown in Figure 1.3, the scene warps the pattern and by triangulation, the position of each dot is calculated. Since the projected pattern is known, the difference between the warped and projected patterns provide information about the scene's depth. Internal processors of the device compare the distortions against the known pattern to estimate the distance value of each pixel and create a 3D point cloud of the scene. These types of sensors do not work well with bright lighting conditions because the overabundant light drowns the projected IR pattern.

Since the light emitter and the camera are different sensors, they are separated by a certain distance which forces the system to set the minimum distance to around 0.6 metres to perform the triangulation. Large distances make the projection sparse and thus the accuracy of the image decreases. These sensors have an optimal capturing range, which depends on the device constraints, but the optimal range of this type of system is approximately 1.2 to 3.5 metres.

Microsoft Kinect, as shown in Figure 2.4, is a camera which captures the depth images of a scene using the Structured Light principle. Since its release in 2010 for the Xbox gaming console, it has been used for many scientific purposes [18, 19, 20, 21]. Regarding hand gesture recognition and Sign Language problems, the Kinect has been used in diverse methods concluding in high accuracy and real-time systems [19, 22, 23, 20].

1.2.3 Stereoscopic

Stereoscopic cameras (Figure 2.4) use two cameras with a certain displacement to get depth information. They simulate the biological eye system to measure depth using a 3D triangulation as shown in Figure 1.4 [15]. These types of cameras have the sensors

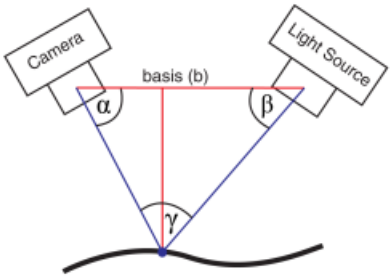

Fig. 1.3 Structured light principle, how to measure the depth of the scene by triangulation, using the distortioned infrared pattern.

embedded in the same device. Since the method of depth estimation is based on the sensor's position and knowing their relative distance, it is not required that the sensors are built-in in the same device. It is viable to get two monoscopic cameras to imitate the stereoscopic ones if the setup of the system is known and appropriate calculations are performed. This method has the same problems as the Structured Light sensors: the field of views of the cameras do not overlap and thus they have a minimum capture distance.

The crucial part of measuring the depth from stereoscopic cameras is to find corresponding points in the two images. Knowing the base length between the two sensors and coordinates of the corresponding points in both images it is possible to compute the depth of it by triangulation. This technique has high computational cost and the resulting depth map has low precision; smooth colour objects in the scene, do not offer good enough features to correspond to pixels in both images, while sharp ones do [24].

Fig. 1.4 3D triangulation of the scene using two common cameras.

Therefore, the accuracy of the depth map is determined by the composition, textures, and colours of the scene.

Even though the computational cost is high and the accuracy is subject to the scene arrangement, stereoscopic cameras have positive aspects: they can be built with normal cameras, thus the resolution and the frame rate is variable; and, they work well with

different lighting conditions.
Diverse publications showed that using these types of cameras, it is plausible to achieve high accuracy for human-computer interaction systems in real time. The author in [25] present a method using Hidden Markov Models (HMM), based on human body images as input to control characters in gaming. Jojic presents [26] a real-time system for detecting pointing gestures.
We chose as the optimal sensor the Microsoft Kinect camera, as it is inexpensive for the resolution and precision, it offers. It has a VGA camera with a 1920x1080 resolution to capture high-resolution colour images. It has a QVGA sensor (Structured Light sensors) with a 512x424 resolution: it captures the depth information of the scene. This sensor has an optimal capture range at 1.5-3.5 meters; objects very close or very far from the sensor are not measured well. The two sensors, VGA and QVGA, work at a frame rate of 30 fps; since the minimum frame rate to get fluid captures is 24 fps, it is sufficient to record videos and also work in real time.
We used the QVGA sensor to capture the images. The frame rate is enough for developing real-time systems, as 30 fps is sufficient to track the bodies of the people and segment the hands from them. One advantage of using the Kinect is the built-in SDK: the requirement to recognize the human body and track the hands is already developed, so there is no need of carrying out with this work again. Approaches for body tracking [27, 20], hand gesture recognition [18, 19, 28] and Sign Language recognition [29, 18, 19, 28], have taken advantage of this camera and there are a lot of existing documentation and forums. The Kinect camera has capturing problems, but defining a specific environment to record users minimizes their impact.

1.3 Categorisation of HGR Methods

To present a systematic review of the works produced by both academic and industrial communities, the existing HGR methods can be categorized according to their purposes and approaches. In Section 1.3.1, HGR methods are introduced under different contexts. A categorization of HGR methods based on the input sensors can be found in Section 1.3.2.

1.3.1 Categorisation by Context

People use hand gestures to convey various messages in communications. For different gesture vocabulary, HGR can be classified into two categories: communicative hand gesture recognition and manipulative hand gesture recognition. Communicative gestures mainly include sign languages. As stand alone languages, sign languages usually consist of 2,500 to 3,500 words [30]. From a pattern recognition point of view, the sheer volume of the vocabulary leads to an enormous number of categories in the fea-

ture space, which is always a difficult classification task. Due to the limitations of the display for the use of hands only, the similarities between the different classes are relatively high. Moreover, to mimic the specific semantic meanings of certain words, exit from the level is often included in gestures (such as the word "bicycle" in American Sign Language [31], as shown in Figure 1.5). Circles are difficult to distinguish from vertical strokes with regular 2D cameras. There are three types of signs in sign languages [32]:

- 1) Verbal signs: Signs are designed based on the semantic meaning of words [33, 32, 34, 35],

- 2) Non-manual signs: they are signs with additional features other than hand movements and postures, Such as head and tongue positions and other positions [36, 37, 38],

- 3) fingerspelling marks: These signs require the performer to spell words by drawing numbers or letters from written languages in the air [39, 40, 41, 42],

Researchers focus on addressing the sheer volume of vocabulary for HGR communication methods, rather than usability issues from unconstrained environments. As for recognizing manipulative hand gestures, the task is more about sending commands to computers, rather than offering communication purposes. Hence, the gestures are intuitive and simple, like swipes to turn pages. From the point of view of ease of use and user experience, the gestures should be as simple as possible. In this way, performing the gestures may distract the user as much as possible. The size of the vocabulary

Fig. 1.5 Word Bicycle in American Sign Language.

is relatively small. The similarities between the gestures can be low. For real-world applications, manipulative HGR methods are used as interactive user interfaces. A potential market is emerging for HGR's manipulative methods [43, 44, 45]. This thesis focuses on HGR manipulation, and more specifically, the recognition of hand gestures for American signs language. Hand position recognition methods are also widely used

in both communicative hand gesture recognition and manipulative hand gesture recognition. In sign language recognition, many words are designed to have distinct hand shapes. Therefore, hand posture is, of course, a feature of the track. Sometimes the paths of the hand in different marks are the same, and only the positions of the hand are different (Figure 1.6 [46]). In these situations, discovering different hand positions becomes part of the recognition process. Also, for the alphabet of finger spelling in sign languages [47, 48], the only task is to know the position of the hand. For manipulative HGR, most vocabulary in real-life applications is usually formed by fixed hand position [49, 50, 51]. For HPR applications, the camera scope focuses on the hand area with relatively small back areas. This leaves background distractions in a small percentage of the scene to render complex and moving textures. The challenges from

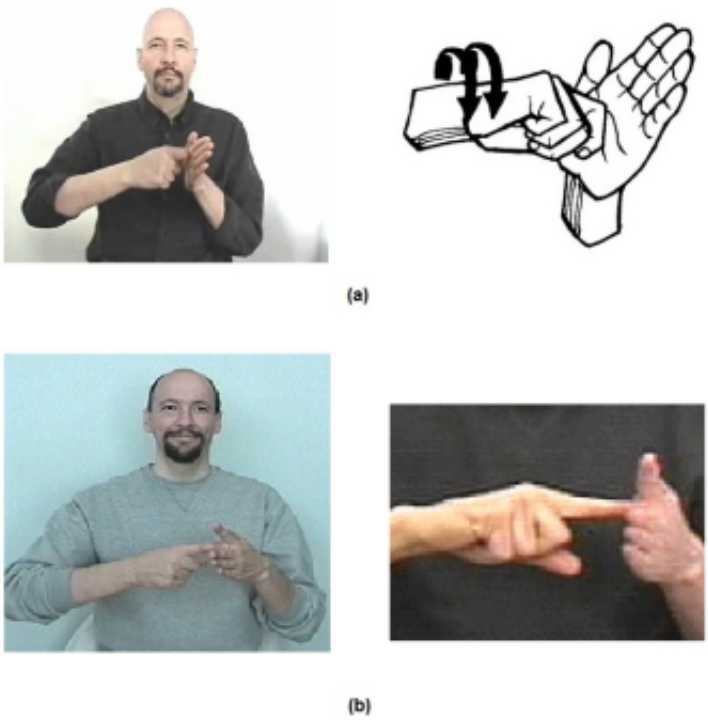

Fig. 1.6 Example of signs that are only different on the hand poses. (a): word Key in the American Sign Language. (b): word start in the American Sign Language.

non-controlled environments that HPR (Hand posture recognition) methods face are much less serious than HGR, which makes real-time response an easier task for HPR methods. HPR methods are also widely used to solve the problem of detecting manual gestures. The simplest solution for HGS (hand gesture signature) is to identify certain hand positions as the starting signal for predefined gestures. This idea has been adopted in both academic research [52, 53] and commercial regulations [49, 43, 50, 54].

1.3.2 Categorisation by Sensor

As a typical pattern recognition problem, the performance of HPR and HGR methods highly depends on the quality of features. Due to the various applications of HPR and HGR methods, different input sensors can provide various imaging methods, such as depth information and infrared data. Hence the classifiers in different applications are facing different types of features. To review HPR and HGR methods, categorization of methods based on input sensors is a vital perspective. In this section, methods with non-vision-based sensors, vision-based methods will be introduced.

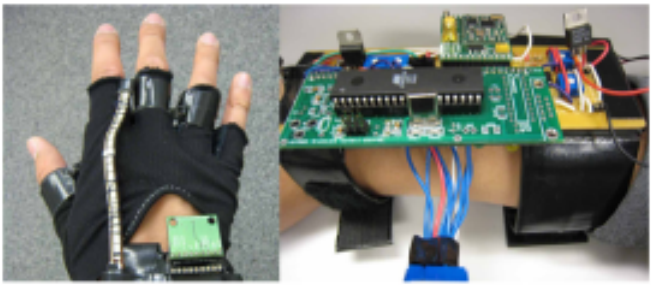

Fig. 1.7 Mister Gloves, Cornell University

1.3.2.1 Non-vision-based HGR

For commercial HGR technologies, stable and robust sensors can provide solutions to challenges from unsupervised environments. Different types of sensors can produce additional information such as pixel depths or hand-held articulated models. Methods using these sensors can identify the target hand without distracting background objects, even front obstacles. Despite the advantages, compared to regular 3D or 2D cameras, the cost of deploying or integrating complex sensors can be relatively high. But this has not prevented non-vision-based methods from marketing in recent years. This section provides a brief review of these methods.

Intuitively, the glove is a natural form of capturing hand gestures and postures. The glove-based method is one of HGR's oldest ideas. Mister Gloves, developed by Cornell University [55], uses an integrated wireless USB transmission on a pair of circuit gloves as an input method. By monitoring hand joint movements to detect different hand positions and sending a wireless USB signal to report hand position, Mister Gloves can accurately recognize dynamic hand gestures and static hand positions. However, the huge hardware costs and the circular gloves precluded the mass production of this method.

Another pioneering glove-based technology is CyberGlove, developed by CyberGlove Systems [56]. The sensor glove is connected to angle gauge sensors and a set of pulse vibration stimuli. Not only can it capture hand movements, but it can also simulate simple sensations on the user's hand. Meaning, the user can "feel," virtual objects through the glove.
Touch screen technologies are widely used today to capture dynamic hand paths. Due to the commercial value, touch screen production has been perfected in the past decade. The touch screen costs are low enough to make it standard compact devices for mobile devices. Touch screen technology is one of the most reliable commercially available HGR methods on the market. MYO [44], is an armband sensor that monitors the user's electrical activity from the arm muscles. The arm bar can predict hand movements and postures using electrical signal patterns. This product expands the limits of usability of HGR technologies. It is portable, wearable, and requires no cameras.

1.3.2.2 2D Vision based HGR

This thesis focuses on 2D vision based HGR, namely appearance based HGR methods. The only sensor used in the methods is a normal 2D camera, without the ability to capture depth information. The reason for this choice is based on the current usability issues with the RGB-D sensors. Currently, RGB-D sensors are not as widely deployed in portable personal electronic devices (laptop, mobile and tablet) as 2D cameras due to the size and extra hardware costs of the RGB-D sensors, despite the fact that the sizes of the RGB-D sensors are getting smaller and the manufacturing cost difference between the RGB-D sensors and the normal 2D cameras is also getting smaller. In other words, from the point of view of real-world application usability, the general population user base of RGB-D sensors is still in a relatively smaller scale than normal 2D cameras. Moreover, no matter how small, the size and price differences of RGB-D sensors and normal 2D cameras remain the major obstacles of the adoption of RGB-D sensors in mainstream consumer electronics. Although the ability of utilizing depth information of the RGB-D sensors is a major advantage over the 2D cameras, the primary motivation of this thesis is to explore the possibility of performing robust HGR with the most widely deployed camera type on the market which currently is the 2D cameras. The appearance based HGR methods normally have 3 main steps, hand segmentation/tracking, feature extraction and gesture classification. The following section provides a detailed review on various techniques used in appearance based HGR.

1.4 Methodologies of Appearance based HGR

In this section, a review of appearance based HGR methods is presented. The review follows the three basic steps of the HGR process. For each basic step, widely used

classic techniques and the state-of-the-art methods will be introduced. Brief analysis on the merits and drawbacks of the techniques can also be found in this section.

1.4.1 Recognition

ANNs are popular in image processing over other classification methods because they can, in theory, be trained to perform any regression or discrimination task [57]. These mathematical models are based on reproducing the behaviour of biological brains by learning from real-world data instead of hand-coding features: in the case of image processing, the features of the images.

The first attempt to imitate a biological brain was the mathematical model creates by Pitts [58], called threshold logic. Based on this idea, an algorithm for pattern recognition [59] was created to perform addition and subtraction operations. These basic neural networks were networks of perceptrons (neurons) organized in connected layers but had limitations in performing some operations until Werbos presented a backpropagation algorithm [60]. The backpropagation allowed the learning of the NN, in adjusting the weights of the layer connections to perform more complex operations. Depending on the problem and the amount of training data, the training step may require high computational cost and time, that is the reason why simpler methods overtook these models in the Machine Learning area.

Nowadays, hardware has more computational power than the early beginning of NNs. Dealing with learning tasks became time-affordable gaining popularity for classification problems. One of the most popular types of NNs for image processing is Convolutional Neural Networks (CNN). They are inspired by the organization of the visual cortex in a biological brain: as shown in Figure 1.8 [61], the perceptrons are not fully connected to the next layer's perceptrons; instead, they are connected only to some of them. This idea was presented in an experiment [62] where it was shown how some specific neurons reacted to particular edge orientations. The global idea of a CNN is having an image as an input to the model, passing it through the different layers and getting an output that can be a single class or a probability of classes. There are three types of operations in a CNN: the convolution step computes a convolution over the input image using the already learned feature maps as filters; the pooling step reduces the dimensionality of the convolved image; and the fully connected layer works as a common neural network, where each neuron is fully connected to the neurons in the next layer.

The CNN and its variations have been used across different areas in diverse problems [63, 64, 65].One of the areas is Natural Language Processing (NLP). In (Col- lobert and Weston 2008) a single CNN architecture is presented to provide information about a sentence: part-of-speech tags, chunks, semantic roles, similar words, etc. Using a CNN, Cicero [64] performed sentiment analysis of short texts using Twitter posts. Text

classification is performed in [66] using a Recurrent CNN, capturing contextual information of the text. Another use of this type of neural network is in the area of speech recognition, due to the spatiotemporal feature of the CNN they are suitable for these kinds of tasks.

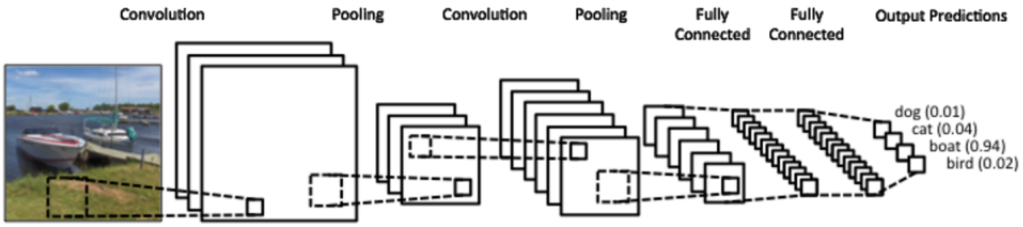

Fig. 1.8 The structure of a CNN

In [63], a hybrid system of a neural network, an HMM, and a CNN is presented for multi-speaker speech recognition. Continuing this research line the same author shows [67] his experiments on how to use a CNN for speech recognition and compares the result against a standard hybrid Deep Neural Network-Hidden Markov Model on the TIMIT recognition tasks.

Among the various areas a CNN can be used in, the most popular one is image processing. In this case, the CNN learns from a training image set and creates feature maps which, after a convolution, detect local features. These features are kept for the next layers of the process to recognize more abstract features by combining simpler ones. This process reduces the dimensionality of the image, and thus the efficiency of the neural network is increased. In [65] diverse methods for handwriting recognition are presented and the use of a CNN is studied; these studies show how CNN eliminates the need for hand-crafted features. The results reported in [64] demonstrates the robustness of a system for detecting handwritten letters on bank checks, integrating it across several banks in the US. Some work in breast cancer detection using mammography images has been done in [68, 69, 70]. Magnetic Resonance Imaging (MRI) images have been processed with a CNN [71, 72, 73] to detect the gray mass in the brain and identify neurological diseases. Challenges in image recognition [74] taking advantage of NNs was raised due to the actual computational power; given a dataset with millions of images challenge participant teams had to recognize objects in the images. The datasets of the challenges are public and scientist used CNNs to create recognition systems [75, 74]. There was an inflection point in 2012 when CNN presented in the work [76] won the challenge ILSVRC 2012 [74] reducing the error rate from 26.2% to 15.3%. With the power of two GPUs, they designed and trained a CNN called AlexNet [76] to deal with the 15 million labelled images.

Other recognition models have been used for image processing and automatic human

body detection which are the main features of various applications: virtual reality, surveillance, person identification, fall detection for elderly people, or gender classification [77, 78, 79]. These applications segment human bodies from environments with varying conditions such as lighting, weather or a number of objects. Once the system is capable of detecting bodies, the next step is tracking them to capture the behaviour of people. As an example, video monitoring with human body tracking can be found in London [77], where the use of RGB cameras in a network all over the city permits the tracking of people.
Another use of body segmentation is body joint estimation to recognize the skeleton and body pose. The detection of different parts of the body empowers Natural User Interfaces for Human-Computer Interaction (HCI) applications. Since the release of the Microsoft Kinect camera, the use of bodies for controlling characters in video games has risen [80].

1.4.2 Hand Segmentation and Tracking

To record the hand trajectory for HGR or extract the hand area for HPR, the target hand area needs to be segmented from the background and tracked throughout the video stream. In this thesis, to distinguish the features of Hand Segmentation and Tracking from the features of Gesture Classification, the features of the former are called spacial tracking features, while the features of the latter are called temporal trajectory features. The most distinctive characteristic of the target hand region against the background is the skin colour. Hence, skin colour detection is widely used in HGR community, which is also one of the active research fields in the computer vision community [81, 82, 83, 84, 85, 86] . Human skin colour has a relatively distinct distribution in the colour spaces [84]. The skin colour tones from different races share similar hue value, but the saturation are different [87], especially in the HSV colour space. That makes the HSV colour space the most widely used colour space to perform skin detection [86, 88]. It is obvious that a model of skin colour can be trained to perform skin detection. [81] trained a Mixture of Gaussian Model through a large database of skin and non-skin images. The detection time is unacceptable for time-sensitive applications. There are also some methods focus on other colour spaces. [89], proposed a Gaussian model for skin detection in the YCbCr colour space. Unsupervised learning methods are also used in skin detection. [83] proposed a clustering method for skin detection to overcome the changing illumination. For real-time applications, the common choice of skin detection method is simple thresholding with carefully chosen thresholds.
For HGR methods that only consider the controlled background without any moving distractions, the target hand can be easily segmented by using scene depth information. Some HPR methods adopt the depth information to facilitate the hand segmentation

through stereo cameras. [90] proposed a Latent Tree Model that is capable of presenting the hierarchical topology of the hand postures, with depth information. [91] proposed a method for learning a compact and efficient model of the surface deformation of human hands from depth information.

For HPR methods, there is no need to perform hand tracking given that the samples are normally still images. For HGR methods, after detecting the target hand, the hand region needs to be tracked. Hand tracking is no different from other standard tracking tasks such as object tracking. Hence, various traditional tracking methods have been utilized for hand tracking. Mean Shift as one of the traditional methods is firstly proposed by [92] or feature space analysis. Mean Shift has been used for tracking in many works [93, 94]. The basic idea is firstly to train a model of the target represented by texture, contour or colour features. Then similar to the optimization methods, Mean Shift takes a density function of the target region model and searches for the optimized matching area in the frames. Continuously Adaptive Mean Shift (Camshift) [95] is a variation of Mean Shift which changes the window size in Mean Shift when the search is near the convergence point. It has also been used for hand tracking [96, 97, 98].

Particle filter methods are essentially grid-based iterative search methods. With the extracted features from the given target region, the methods search for similar regions in the whole frame iteratively. In each iteration, the search biases to the matching result from the last iteration until the search converges in a region. Particle filter methods are widely used for tracking [99, 100, 101] .[102] proposed a hand tracker that combining, the Particle filter with Mean Shift. Another well-known hand tracker proposed by [103], trains a dynamic model to guide the particle filter search. [103] proposed a method using Hierarchical Bayesian filters for model-based hand tracking.

Optical flow [104, 105] is a velocity field that shows the movement of pixels. Optical flow can be used to extract trajectories of pixels in video streams. However, it works under two assumptions. The first one is that the target feature point has constant texture and brightness during the motion. The second one is that all pixels within the neighbouring region of the target feature point are moving towards the same direction as the target point.[106], proposed an optical flow hand tracking method that can recover the full hand articulation model with 27 degrees of freedom from the gray level images. [107] proposed a fusion model that combines optical flow with other features to perform hand tracking.

1.4.3 Feature Extraction

For Hand Posture Recognition, the feature of posture classification is the data representation of the hand shapes. There are two main types of features for HPR, contour and texture features. Contour features are descriptors that represent the exterior contour of

the target hand region. The Fourier descriptor is one of the most widely used contour features. The basic idea is making the contour a one-dimensional vector, similar to the chain code. Then Fourier Transform is applied to the vector. The Fourier coefficients can then be treated as the feature of the contour [108, 109]. Moments are descriptors that represent various contour properties, such as the sum of horizontal and vertical directed variance. Other contour features include Wavelet descriptors [110], shape signatures such as average distance between pixels on the contour, gradient shape features [111] and the centroid of the contour, etc.

Texture features are essentially various representations of the gradient patterns. Histogram of Gradient (HoG) [112] as one of the popular texture features is a histogram presentation of the local gradients. Scale-Invariant Feature Transform (SIFT) [113] is another popular texture feature. SIFT contains selected key points with coordinates and gradient descriptors. The key points are local extreme values in the Laplace of Gaussian (LoG) pyramid. The descriptors summarize the orientation and intensity of local gradients. The two important advantages of SIFT are its invariance against rotation and scale. Hence, the SIFT key points represent the edges and ridge-like textures regardless of the orientation and scale of the target object. Moreover, it can also be tolerant certain level of view point changing. Speeded-Up Robust Features (SURF) was originally presented by Herbert [114] based on the idea of SIFT. It also inherited in-plane rotation and scale invariance, which makes it desirable for Hand Posture Recognition. However, with high precision of texture matching, the 64 dimensional descriptor of SURF requires intense computations for both key point extraction and matching. [115] proved that this problem can be fixed by directly building a short binary descriptor with independent bits. That is called the Binary Robust Independent Elementary Feature (BRIEF) [115]. This binary descriptor uses Hamming distance as matching criteria, instead of Euclidean distance between the descriptors. That can largely fasten the texture matching process. However, the descriptors are not rotation and scale invariant. [116] proposed an improved binary descriptor, which is rotation invariant, called Oriented Fast and Rotated BRIEF (ORB). There is an additional advantage of ORB which is its robustness to noises. [117] also introduced a binary descriptor that is both rotation and scale invariant. It is called Binary Robust Invariant Scalable Key points (BRISK). The main characteristic of BRISK is that, in each scale pyramid octave, a corner detection method called Features from Accelerated Segment Test (FAST)[118], is used to detect the key points, instead of simply locating local extreme values as in SURF and SIFT. That makes the key point selection more efficient in BRISK, and ensures that the number of key points in BRISK descriptor is lower than SIFT and SURF. However, the robustness of BRISK descriptors can be affected by out-of-plane rotation or rapid texture changes. That is a vital drawback for applications such as HPR. Fast Retina Key points (FREAK) proposed by. [119] is the latest development on gradient-based

key point texture features. It simulates the principle of human retina visualization. A cascade of binary descriptors is used instead of a single descriptor. The cascade structure is constructed by comparing image intensities over a sampling pattern based on the human retinal ganglion cells distribution, in which the method picks more key points in the central area. Although . [119] reported better performance over SURF, its robustness against illumination and view point changes in HPR applications still remains untested.

For HGR applications, the aforementioned gradient-based material master point descriptors can also be used for texture-based traceability. After the method for tracking the hand has determined the position of the candidate hand in the frames, the temporal features are extracted to represent the tracks. In contrast to the spatial features of surroundings and texture, there are only a few commonly used pathway features. It can be classified into two types, local and global features. Local trajectory features include hand speed, position, and direction of movement [120, 52]. Hand speed, motion direction, and hand displacement coordinates between adjacent frames can be used to describe elementary path segments. General features are descriptors of shapes that are extracted from full-hand paths. All the previously mentioned contours and texture features can be used as the Universal Pathway features. Since the positions of the hand are the same, paths can be viewed as still images.

1.4.4 Gesture Classification

All classifiers are trying to summarized feature patterns from the training set, and then applying these patterns to classify the testing samples. For HGR, the hand trajectories are sequence data represented by both spatial and temporal features. The classifiers must be capable of processing sequential features in order to classify the hand trajectories. But for HPR, the classifier does not have to process sequence data since the samples are still images without any temporal information. In this section, a review of some of the popular classifiers for HGR and HPR is presented. The template matching is a strategy for directly measuring the distance between the testing sample and the predefined gesture class models. The advantages of this strategy are twofold. Firstly, minimum training is required. Since the methods directly calculate the distance of two feature vectors in the feature space, instead of extracting latent patterns or measuring elementary local patterns, the templates of gesture classes are usually pre-processed feature vectors of training samples. Hence, there is no need to build statistical models for the gesture classes through the training process. Secondly, the inference time cost is usually low. Namely, the calculation of the feature vector distances does not require complex computations. Although the computational complexity depends on the dimensionality of the feature vectors, template matching methods are still considered

less computational intensive than statistical models. Continuous Dynamic Programming (CDP) as one of the popular template matching methods is proposed by. [121] for segmenting and recognizing continuous hand gestures. A set of sequence patterns are used to represent trajectories in the spatiotemporal space. A dynamic programming based method is used to match the sequence patterns, which cumulatively adds the distances between corresponding elements in the sequence patterns. Decent results are reported on an 8-hand gesture database [121]. Alon et al. proposed a hand gesture segmentation scheme based on CDP [122].A pruning method is introduced in this scheme to discard hand trajectories with relatively short length. An improved version with template matching method based on Dynamic Time Warping (DTW) is proposed later [120]. This work introduces the concept of sub-gesture reasoning, which learns the relationships among the gesture classes. For Hand Gesture Spotting, subgesture reasoning can improve the ability of segmenting similar gestures. However, these two methods require extra computations on estimating the location and scale of the gestures. In other words, the methods of [122, 120] do not have gesture scale and location invariance property. The DTW classifier itself still requires estimated scale and location of the gesture trajectories. A pruning technique is also used in the method to reduce the number of hypotheses. DTW based methods are normally used for tackling temporal element displacements in the templates. If the sequential order of the elements has a certain level of variance in the testing set, DTW based methods can overcome the variance by matching the elements within a corresponding time window without considering the order of the elements. However, using the length of the trajectories as a criterion for pruning means the method is not gesture speed invariant. For the testing samples where the gesture performer signs the gestures relatively faster than the performers in the training samples, there is a high probability that the method of [122, 120] would prune off these testing samples regardless of the actual gestures labels of the samples. Hence for the uncontrolled environments, the methods of [122, 120] are sensitive to the various scales, speed and location of the gestures.

HMM based methods use a set of transitional probabilities to simulate the local dependencies between the adjacent observation states. The model needs to estimate the state and transitional probability matrices among the observations and hidden states in the training set. Then the inference task is performed through forward-backward propagation based on the trained probability matrices. [123, 124] introduced two HMM based models to perform HGR on a 40 words sign language vocabulary, and reported decent accuracy. A few HMM variations are proposed for different specific applications. [125] trained a dedicated HMM model for each gesture class, with various number of hidden states. [126] introduced a coupled HMM method for classifying two-handed signs. This method is proven to be robust against initial observation probability changes. [127] proposed a parametric HMM, which extends the original HMM by including a global

parametric variation on the output probabilities of the hidden states. But HMM is only able to take the last observation state into account for inferring the current hidden state. That means HMM is incapable of monitoring long-range dependencies within the observation sequences. In the context of HGR, the transition probabilities in HMM can only represent trajectory temporal features within adjacent frames. The transition probabilities are not considered under the context of the entire trajectory. Another popular concept in pattern recognition is Deep Learning. One of the main concepts of Deep Learning is to train the features instead of using man-made features. [128] showed the potential of Deep Learnin methods in solving various computer vision problems. Convolutional Neural Networks (CNN) proposed by [65], is one of the best examples of Deep Learning methods. The key innovation of CNN is choosing trainable features over heuristic features. It breaks the conventional concept of man-made spatial features. In the CNN model, a set of fix-sized trainable kernels are defined to extract local texture features. The training process is based on optimization methods with an error function on the whole training set as the objective function. Hence, the training process is essentially searching for optimized kernels that can minimize the error function of the training set. The kernels act like texture feature extractors. In each convolution layer of the neural network, the input image will be convolved with the trained kernels for feature detection. Every convolution layer is followed by a pooling layer which down samples the input image to one fourth of the size. As the input image goes deeper into the network, the kernels of fixed size are monitoring texture features on larger scales. The final output layer of the network produces the final scores based on the input signals from all previous layers. [129] simplified the CNN model. The simplified version does not require weight decay and averaging layers. Multi-column Deep Neural Networks (MCDNN) is introduced by [130], as a CNN structure with a large number of feature maps in each convolution layer, and a large number of convolution-pooling layer pairs. In other words, this is a wider and deeper neural network. It is proven in this work, that with more feature maps to monitor a large number of local texture features, the MCDNN is capable of producing state-of-the-art performance on various computer vision applications.

1.5 CONCLUSION

In this chapter we have studied the various methods of gesture recognition. Hand gesture recognition system is considered as a way for more intuitive and proficient human computer interaction tool. The range of applications includes virtually prototyping, sign language analysis and medical training. Also, we have identified how to classify the noninterest images and interested gestures from the taken actions or images, inertial sensor is used to identity the gestures. In this chapter we have discussed about the end-

Point algorithm and also a few techniques of it is Mono-vision technique. Each of them performed all the hand gestures.

Some recent publications proposed promising methods for activity recognition and prediction in uncontrolled environments. [131] proposed a novel hough- transform based voting method which uses random projection trees to perform feature voting. This method reported taking about 10 seconds to perform the activity classification for a video with four seconds length. It is far from real time, but the method is robust against crowded scenes. [132] proposed a forward human action prediction method which represents an activity as an integral histogram of spatiotemporal features. A novel recognition method called dynamic bag-of-words is also proposed in this work, which is capable of taking the sequential nature of human activities into account while maintaining the merit of noise tolerance of the bag-of-word method. This work can perform prediction in real time given that the features are fed to the method in real time. [133] proposed Hough Forests for human action recognition which are random forest variations utilized to perform a generalized Hough transform. They reported 10-second time cost for classifying pre-existing actions in 100 frames.

CHAPTER 2

Image Processing and Recognition

2.1 Image and Signal Processing

Image processing is a method of converting an image into digital form and performing other operations on it in order to produce an enhanced image or obtain useful information from it. The input is an image, like a video frame and the output can also be an object or an image. This is a kind of signal spread [134]. The image processing system contains various tools such as image acquisition, image enhancement, image recovery, colour image processing, wave processing, multiple solutions, segmentation, and object recognition.

Each tool will be defined as follows: Image acquisition is taking a picture and digitizing it then analysing the problem area and then following the steps according to the problem. Image optimization is carried out across time and frequency to optimize the image according to the requirements. Image recovery stores specific parts of an image with the Point Spread function. The colour image manipulation tool is used when the image is black and white. Wavelet and multi-resolution processing are used if the images are to be rendered in different degrees or wavelengths of different resolutions. Compression reduces the size of images using a specific function. Morphological processing is an external structure of an image using dilation and erosion. Splitting is performed by dividing the image into different parts. Object recognition is used to recognize and save an image description [134, 135] .

Image processing is faster and more cost-effective. Processing takes less time and reduces film and other imaging equipment. Image processing is more environmentally friendly. No processing or repairing chemicals are required to capture and process digital photos. Printing inks are an essential component when printing digital photos. The cloud-based tool for the Microsoft API allows developers to access advanced image processing and data algorithms, image transmission or image URLs, and visual content analysis in many ways that support input and user identification [136]. Amazon Rekognition is a cloud software used to integrate image and video analysis into user

applications. It can recognize objects, people, texts, scenes, and events in an image, video, or other inappropriate content [137]. SimpleCV is an open source computer vision platform that allows users to access various high-powered computer vision libraries like OpenCV without thinking about bit depths, file formats, colour spaces, buffer management, individual values, or matrix versus bitmap storage. Photoshop is a program used to edit digital images [138].

2.2 Pattern Recognition

As shown in Figure 2.1 [139], pattern recognition methods can be divided into two main categories: two-stages and end-to-end. Most traditional methods are two stages, i.e. with cascaded handcrafted feature representation and pattern classification.
The feature representation is to transform the raw data to a feature space with the property of within-class compactness and between class separability. Reprocessing (like removing noise and normalizing data) is firstly applied to reduce within-class variance, while feature extraction further enlarges between- class variance, and this procedure is usually domain-specific.
Actually, for solving new pattern recognition problems, the first thing is the design of feature representation, and a good feature will significantly reduce the burden on subsequent classifier learning. This kind of effort can be found in different applications like iris recognition, gait recognition, action recognition, and so on [139]. After feature

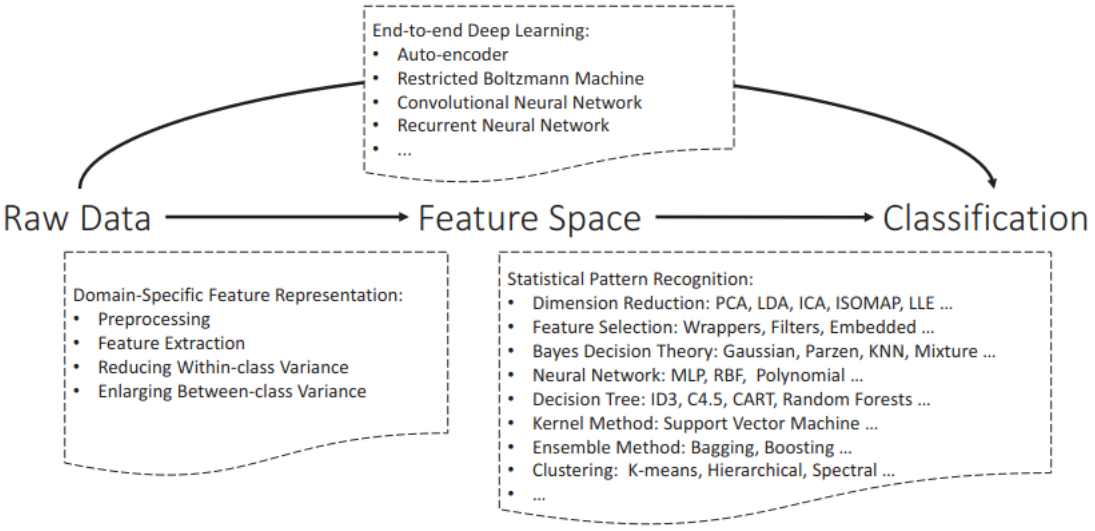

Fig. 2.1 A brief overview of pattern recognition methods.

representation, the second stage is pattern classification, which is a much more general problem. Actually, classification is the main focus of many textbooks including

Fukunaga [140], Duda et al. [141], Bishop [142], and so on. This stage is also known as statistical pattern recognition [143],where many different issues are considered from different perspectives. Firstly, dimensionality reduction is widely adopted to derive a lower-dimensional representation to facilitate subsequent classification task. Another approach of feature selection can be viewed as a discrete dimensionality reduction. After that, many classical classification models can be applied. The most fundamental one is the Bayes decision theory which integrates class-conditional density estimation with prior probability for maximum posterior probability classification.

Artificial neural network is also widely used for pattern classification, including MLP (multilayer perceptron), RBF (radial basis function), polynomial networks, and so on. Decision tree based methods use a tree structure to represent the classification rule. Kernel methods, have been widely applied to extend linear models to nonlinear ones by performing linear operations on higher or even infinite dimensional space transformed implicitly by a kernel mapping function, and the most representative method is SVM (support vector machine). Ensemble methods can further improve the performance by combining predictions from multiple complementary models. Clustering is widely used as an unsupervised strategy for pattern recognition [139].

In two-stage methods, we usually have multiple choices for both feature representation and classifier learning. It is hard to predict which combination will lead to the best performance, and in practice, different pattern recognition problems usually have different optimal configurations according to domain specific experiences. Contrarily, deep learning methods are end-to-end by learning the feature representation and classification jointly from the raw data. In this way, the learned features and classifiers are more cooperative toward the given task in a data-driven manner, which is more flexible and discriminative than two-stage methods.

Formerly, deep neural networks are usually layer-wise pre-trained by unsupervised models like auto-encoders and restricted Boltzmann machines. Nowadays, deeper and deeper neural networks can be trained end-to-end due to many improved strategies such as better initialization, activation, optimization, normalization, architecture, and so on. Due to shared-weight architecture and local connectivity characteristic, the convolutional neural network has been successfully used in many visual recognition tasks like image classification , detection, segmentation, and so on. Moreover, due to the ability of dealing with arbitrary-length sequences, the recurrent neural network has been widely used for sequence-based pattern recognition like speech recognition, scene text recognition, and so on. Furthermore, the attention mechanism can further improve deep learning performance by focusing on the most relevant information. Nowadays, deep learning has become the cutting-edge solution for numerous pattern recognition tasks.

Besides the broad class of statistical pattern recognition approaches, structural patterns recognition has been developed for exploiting and understanding the rich structural in-

formation in patterns. Unlike statistical feature representation, the structure of patterns is of variable dimensionality, and can be viewed as in non-Euclidean space. String matching and graph matching are basic problems in structural pattern recognition. To improve the learning ability of structural pattern recognition problems, kernel methods (with graph kernel), probabilistic graphical models, and graph neural networks have been used. Overall, the research and application of structural pattern recognition are less popular than that of statistical methods [139].

2.3 Computer Vision Systems

Computer vision is a field that aims to enable computers to interpret, recognize and process objects in the same manner as human vision. It is similar to giving intelligence and instincts to a human computer. In fact, it is a difficult task to recognize computer images of different objects. Computer vision is closely associated with artificial intelligence because machines need to understand what they see and then interpret or act appropriately [144].

Computer vision architecture involves processing digital images via different stages successively. The first stage is the image acquisition which captures an image and digitalizes it and then analyses it according to the problem domain. Image Processing is the second stage which is a method to transform an object into a digital form and perform certain operations on it in order to produce an enhanced photo or obtain useful information from it. Image processing is also a form of signal dispensing where the input is an image, such as a video frame or image, and the output can be an object or image-related features. The third stage is image analysis, which extracts a piece of information, and data processing. This method is typically necessary to ensure that certain assumptions suggested by the system are satisfied before a computer vision approach can be applied to image data. Feature Extraction is the fourth stage in computer vision systems which extract features of the object and are derived from the image data at different levels of complexity. The fifth stage is detection/segmentation where a decision is made at some point in the processing whether points or regions of the image require further processing. High-level processing is the sixth stage where the input is usually a small set of data at this level such as a set of points or an image area that should contain a specific object. Lastly, decision-making consists of releasing the final decision needed for the application [144].

The applications of computer vision include Optical Character Recognition (OCR)- interpreting handwritten letter codes and Automatic Number Plate Recognition (ANPR). An example of a computer vision application is a machine inspection where the quick quality inspection of aircraft wings or auto body parts or X-ray vision defects in steel casting using stereo vision with special lighting. It is also used in retail to classify items

for automated checkout lanes. It is used in 3D model creation (photogrammetry) where completely automated 3D photographic aerial models are used in applications like Bing Maps. Moreover, the field of medical imaging utilizes computer vision currently and is applied in several ways including capturing preoperative and intraoperative images as well as to perform long-term brain morphology studies in individuals as they age [144].

2.4 Neural Network

2.4.1 History

The history of neural networks can be arguably dated from 1943 when Warren McCulloch and Walter Pitts invented a mathematical model encouraged by the Biology of the central nervous systems of mammals.
This encouraged the invention of Perceptron, created in 1958 by Frank Rosenblatt. The perceptron used very modest model mimicking biological neuron that was based on the mathematical model of Pitts and McCulloh. Definition of the perceptron model also defined an algorithm for direct learning from data.
In the beginning, Perceptron looked very promising, but it was soon discovered that it had severe restrictions. Most projecting voices of criticism was Marvin Minsky. Minsky published a book in which he laid out a case that the perceptron model was unable to resolve complex problems [145]. Amongst others, the book contained mathematical proof that Perceptron is incapable of solving a simple XOR problem. More generally the perceptron is only proficient of solving linearly separable problems. However, according to Minsky, this criticism wasn't malicious; it in effect stifled the interest in NNs for over a period.
Awareness in NNs was rejuvenated in the early '80s when it was shown that any previously raised up deficiencies could have been resolved by the usage of multiple units. This was later exacerbated by the development of the back-propagation learning algorithm, which allowed the possibility to gather neurons into groups called layers, which can be weighted into hierarchical structures to form a network. NN of this type was generally called Multilayer Perceptron (MLP). In the 80s and 90s, the awareness of NNs plateaued again, and general research on AI was more focused on other machine learning methods. In the field of classification problems, it was particularly SVM and ensemble model. AI research communities also established several other paradigms of NNs that were likewise inspired by the biology of a certain aspect of the central nervous system but took different methods. The most significant examples were Self-Organizing Maps and Recurrent Neural Network (RNN).
By the year 2000, there were very few research groups that were applying enough attention to the NNs. There was also a certain disdain for NNs in the academic world and AI research community. The success of NNs that was promised almost half a century

ago was finally coming across around 2009 when the first networks with a huge number of hidden layers were effectively trained. This led to a typical adaptation of an umbrella term deep learning which by and large refers to Deep Neural Network (DNN). The term deep indicates that networks have many hidden layers.
The key theoretical vision was to learn complex functions that could represent high-level abstractions such as vision recognition, language understanding, etc. There is a requirement for deep architecture.
NNs in the times before Deep Neural Networks had only one or two hidden layers. These are currently called shallow networks. Typical Deep Networks can have a number of hidden layers in order of 10s but in some cases even hundreds. Still, the progress of Neural Network into the direction of structures with a high number of hidden layers was obvious; its training was an unresolved technical problem for a very long time. There were fundamentally three reasons why this invention didn't come sooner.

1. There was no procedure to allow the number of hidden layers to measure.
2. There wasn't enough of labels data required to train the NN.
3. The computer hardware wasn't powerful enough to train adequately large and complex networks successfully.

The first problem was tackled by the creation of CNN's. The second problem was explained simply when there were more data presented. This was primarily achieved thanks to an effort by large companies like YouTube, Google, Facebook, etc. But also, with the support of a large community of experts and hobbyists in data sciences. Both inventions in computational hardware and improvement of training methods were needed to resolve the third problem. One of the technical revolutions was the use of Graphics Processing Units (GPUs) for the demanding computation involved in the training of a complex network. Thanks to the fact that the training process of NNs is typically a large number of simple resulting computations, there is a great possibility for parallelization [146].

2.4.2 Structure of Neural Networks

The term NN is very general and it defines a comprehensive family of models. In this framework NN is distributed and parallel model that is capable of approximating complex nonlinear functions. The network is made from multiple computational components called neurons assembled topology.
Explanation of the NN structure will follow the convention laid out in the explanation of the learning algorithm. Meaning that an explanation of the learning algorithm is composed of the model, cost function and optimization technique. The difference comes

into performance with the fact that the model of NN is much more complicated than the model linear regression [146].
Therefore, the investigation is divided into a model of neuron and topology of the network.

2.4.3 Model of Neuron

A neuron is a computational unit carrying out the nonlinear transformation of its inputs

$$y = g(w^T x + b). \tag{2.1}$$

Argument $w^T x + b$ of function g is often observed as z . Therefore, the equation can be rewritten as

$$y = g(z). \tag{2.2}$$

The typical schema is shown in Figure 2.2 [146], which describes the inputs, weights bias and activation function. As it was already stated, model of the neuron was stim-

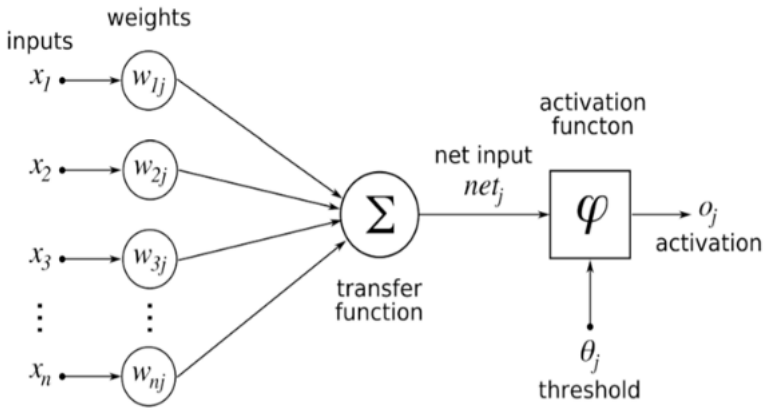

Fig. 2.2 Diagram of the artificial neuron.

ulated by biology. First attempts to make a model of a neuron had multiple elements equivalent to neurons of the human brain. As research proceeded, this equality ceased being as important and modern NN models correspond to their biological matching part only superficially.

Input For each neuron has multiple inputs that are combined to execute some operation. Each input has selected weight assigned to it. Here, we have considered an input of images with the size 50x50x1 (Height X width X Channel) pixels. If we input this to our model Convolutional Neural Network, we will have about 2500 weights in the first hidden layer itself.

Weights Inputs of a neuron are weighted by parameters w that are changed during the learning process. Each weight gives strength to each individual input into the nerve cell. The basic awareness is that when the weight is small input doesn't affect the output of the neuron very much. Its effect is large in the opposite case.

Bias Another changeable parameter is biased b that controls the impact of the neuron.

Activation Function Activation Function For NN to estimate nonlinear function, each neuron must perform the nonlinear transformation of its input. This is completed with activation function g(z) that performs the nonlinear transformation. There are numerous different normally used activation functions. Its usage depends on the type of network and on the type of layer in which they activate. One of the oldest and historically most frequently used activation functions if sigmoid function. It is defined by

$$g(z) = \frac{1}{1 + e^{-z}}. \tag{2.3}$$

Problems with sigmoid is that its gradient becomes flat on both extremes and as such it reduces the learning process.
One more activation function is the hyperbolic tangent. It is defined as

$$g(z) = tanh(-z). \tag{2.4}$$

The hyperbolic tangent function doesn't use that much in feedforward NN, but it is mostly used in RNN. Currently, the most commonly used activation function is restricted Linear Unit (ReLU). It is very generally used in both convolutional and fully connected layers. It is defined by

$$g(z) = max(0, z). \tag{2.5}$$

It has a disadvantage because it is not differentiable for z= 0, but it is not a problem in software execution and one of its biggest advantages is that it can learn very speedily. In this research, ReLU activation function was used as it is generally used in convolution and fully connected layers.
All three activation functions are illustrated in Figure 2.3 [146].

2.4.4 Topology of the Network

There are several different generally used topologies. The two most frequently used in deep learning are feed-forward and recurrent. Feed forward networks are categorized by the fact that during activation the information moves only in a forward direction

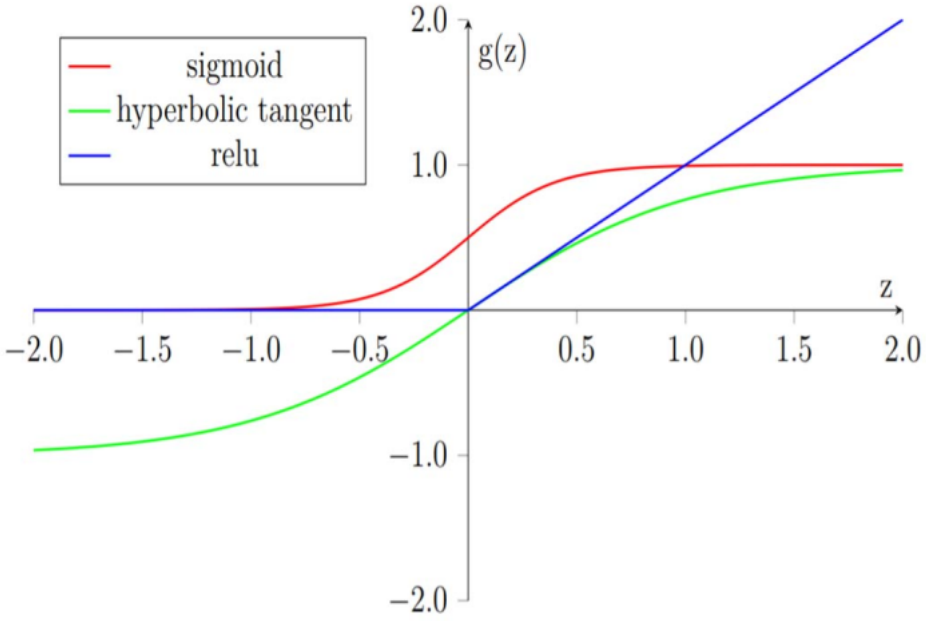

Fig. 2.3 Activation Functions

from inputs to outputs. A recurrent network has provided some sort of feedback loop. Another principle of topology is how are individual neurons in the network linked. Most commonly are NNs ordered in layers. In each layer, there can be from one to n neurons. Layers are hierarchically fixed. The first layer is called the input layer, the last layer is called an output layer and the layers intermediate are called hidden.

Description of the network recreations on interconnections between individual layers. The most common structure is called fully connected where to each neuron in hidden layer has input associates from all neurons from previous layer $l - 1$ and its output is associated with the input of each neuron in following $l + 1$ layer. The entire structure is illustrated in Figure 2.4. After this point on the term, NN will refer to Feed-forward Fully Connected Neural Network.

Types of neurons are dependent on the type of layers provided to the network. Currently, the core difference is in their activation function, which wasn't the case for a long time. In history, all layers had neurons with a sigmoid activation function. It was mostly because the output sigmoid layer can be easily mapped onto probability distribution since it obtains values between 0 and 1. Only relatively recently it was found that network composed of neurons with ReLU activation function in the hidden layers can be trained very speedily and are more resistant against over-fitting. Activation functions are still subject to ongoing research.

Neurons in the output layer necessitate output that can produce a probability distribution that can be used for approximating the probability of individual classes. For this reason, most frequently used activation function of output neuron is called SoftMax [146].

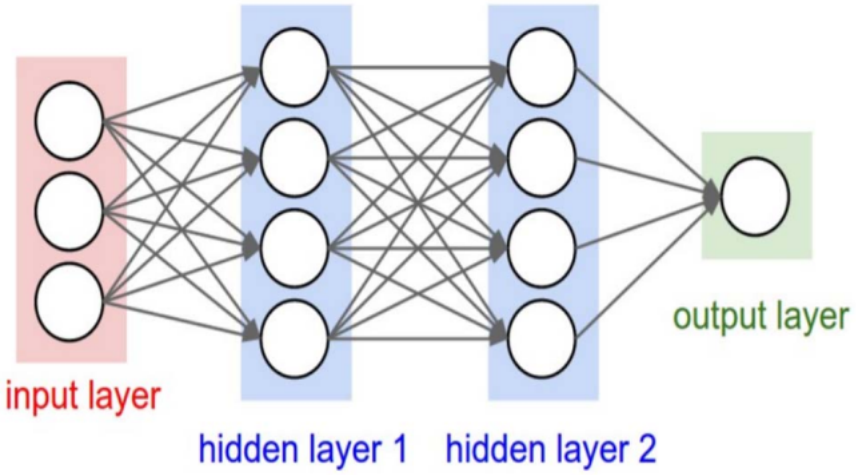

Fig. 2.4 Fully connected Feed Forward Neural Network.

SoftMax is a standardized exponential function. It is used to represent the probability of an instance existence member of class as

$$g(z)_j = \frac{e^{zj}}{\sum_k^{k=1} e^{zk}}. \tag{2.6}$$

where K is the total number of classes.

2.4.5 Cost Function

Cost functions of NNs are a complex subject that exceeds the scope of this thesis. One of the most common cost functions used in NNs for classification in multiple classes is categorical cross entropy. In SoftMax activation function from Equation 2.6 is a cost function defined as

$$C = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} lng(z^{(i)}) + (1 + y^{(i)}) ln(1 - g(z^{(i)})). \tag{2.7}$$

where $y^{(i)}$ is the correct class of the instance and n is the total number of instances.

2.4.6 Optimization Procedure

Every optimization technique for NN is constructed on a gradient descent. It is an iterative process to lower training errors of the network by differentiating the cost function and adjusting parameters θ of the model by following the negative gradient.
The problem is that the cost function of the whole network is very complex and has many parameters. To find the gradient of the cost function, it is essential to go through all the units in the network and estimate their contribution to the overall error. A method

that is used to solve this problem is called back propagation.
Back propagation is frequently confused to be a complete learning algorithm which is not the case, it is only the method to compute the gradient [146].

2.5 Medical Image Applications

With advancement in medical images such as Magnetic Resonance Imaging(MRI), Computed Tomography(CT)scan and ultrasound technologies, the need to process the image data in order to extract valuable information increases. This need attracts experts from many fields such as statistics, applied mathematics, biology, physics, engineering, computer science and medicine. In this section, it discusses some research work in the domain of medical image processing. [147]. Presented a detail review of different web-based interactive software tools for 2D/3D medical image processing. In an earlier work [148] proposed an image processing and visualization algorithm that works interactively for diagnosing moving organs such as the heart during the cardiac cycle. The system is tested on Magnetic Resonance(MR) and Single-Photon Emission Computed Tomographic(SPECT)images. These MR and SPECT images allow the user to change classification parameters and to zoom or rotate images on the screen.One of the major problems in the medical field is breast cancers in women, and in men also. It is estimated that approximately 20% of cancer patients relate to breast cancer in developed countries. [149]. Suggest a method of feature extraction and classification in order to diagnose breast cancer in its early stage. First, they collect 62 texture and photo metric image features and after a stepwise discriminate analysis, six of them can be used to detect the affected and non-affected areas of the breast; 72% average classification results are recorded. This system can be used by radiologists to analyse any pattern in mammograms. The regions identified by the system can have 72% of chances of developing a malignant mass. This could help in earlier diagnose of breast cancer. The goal of this system is to flag the suspicious area.Segmentation and visualization of medical images have been discussed by [150]. Where 3D interaction is achieved using stereo graphic and haptic feedback. Creating 3D models of human organs from the 2D images is a hot topic in medical fields these days. These 3D models of CT and MRI scans provide better perspective than 2D images. In order to construct 3D images, sequences based on CT and MRI, surface and volume rendering techniques are mostly used. Geometry of surface and rendering are two phases modelling techniques for constructing 3D models for complex biological structure. One of such work is by [151] in their research work, where they present their Marching Cube's algorithm for modelling. Image segmentation is the necessary pre-processing step used to detect the boundary of regions of interest. This segmentation is especially important in the clinical field as it increases the accuracy of the results. This procedure incorporates volume rendering and surface

rendering. The surface rendering is utilized increasingly because of its quick speed and low stockpiling utilization.

2.5.1 Motion Detection

In an earlier work [152]provide a unified model for detecting moving objects both in 2D and 3D scenes.The method used in this study is based on separating the object moving detection problem into different categories based on complexity and using a set of techniques to solve these problems which correspondingly increase in complexity. Examples of real image arrays were then used to illustrate these techniques [153] presents a data mining approach for motion detection in huge surveillance video databases collected by military surveillance cameras. They follow completely qualitative approach, based on signaller system consistency analysis, called QLS. This approach focuses on what is necessary to compute the solution hence reducing the computational cost and increasing the efficiency. [154]. Proposed a technique for detection motion regions in video sequences. This technique classifies image pixels into motion regions by applying 2D planar homographs, popular and geometric consistency constraints. Their main contribution is geometric consistency constraints derived from the camera poses from three successive frames. It is implemented within the Plan + Parallax framework. In 2007, [155]. Presented a Principle Component Analysis(PCA)-based approach to detect motion in surveillance videos. Ten frames are considered, where each of the ten frames is associated with one dimension of feature space. Then they apply PCA to map data in lower-dimensional space. These ten frames are then split into blocks. To detect motion within the blocks, inertia ellipsoids of the projected block are used. They recorded very few false positives and satisfying number of connected components as compared to other same purpose algorithms. Automatic detection of motion in human bodies has been discussed by [156]. Using a low-dimensional spatial-temporal model, they develop a presentation model that learned using motion capture data of humans.

2.6 Summary

The theory of image processing is explained in this chapter. Image processing is a method of applying some techniques to digital images. The applications used in image processing are a remote sensing, entertainment and geological processes. The second theory mentioned in this chapter is video processing which is an analytical technique implemented on video data that is allocated for the time and operation to achieve essential processes. Image processing algorithms such as empirical mode decomposition and Wavelet Transformsare discussed in this chapter. EMD is a method to analyse the non-stationary and non-linear data while WT applies signal analysis where signal frequency changes at the end of time. For classification, this chapter presents one of the

common classifiers, Artificial Neural Network. The main definition of ANN is an electronic model like the structure of human brain neurons. The functionality of ANN is the first node (input) will feed data to a set of hidden nodes to train, then will be classified in the end by producing one node or more (output). The last theory is one of deep learning method namely convolutional neural network. CNN is a multi-layer neural network and each layer of a CNN sends amounts of activation to another layer via function. For feature extractions, Convolution2DLayer, Rectified Linear Unit(ReLU)Layer and MaxPooling2DLayer are applied to data before transforming it to classification layers which are fully connected layers, Softmax layer and classification output layer.The following chapter presents the background of HCI and the history of gesture recognition with its fundamental types.

CHAPTER 3

Gesture Recognition

3.1 Background

This chapter presents the background of HCI and the history of gesture recognition with its fundamental types. An overview of hand gesture recognition and its types is discussed in this chapter, involving the types of cameras used for 2D and 3D images. Lastly, the chapter will focus specifically on the fundamental concept of holoscopic 3D imaging system camera. Users interact with computers through provided interfaces, motions or vocal. These different interactions need to be such, that information retrieval is easier and Human Computer Interaction (HCI) is concerned with the way humans interact with technology. It deals with how humans work with computers and how computer systems can be designed to facilitate the users in achieving their goals. With the advent of the third and fourth generation languages, the user interfaces have improved quite dramatically. In future days, Human Computer Interaction HCI will become a field with a variety of sectors that need to characterize it. Users will be able to use any type of interaction which is a potential part of HCI, Interaction can be the body movements, facial features and vocals [157, 158].

A Human Computer Interaction (HCI) has several types of interaction and one of those is called gestures. One simple definition of a gesture is a non-verbal method of communication utilized in HCI interfaces. The high target of gesture is to design a specific system that can identify human gestures and use these gestures to convey information for device control. Recently, HCI has increased in relevance as its usage increases across different applications including human motion acquisition. Initially, it must define the idea of human motion acquisition which records the movements of a human or an object and convey them as 2D or 3D image data. To provide life to 3D digital models or analyse the motions need to study the converted 3D data deeply. Producing a 3D digital object needs special applications and specific tools which are considered exclusively to certain companies [157, 158]. For instance, Disney is one of the most famous companies which use human motion capture as a modern technology in the cartoon world production. Avatar characters, as the first 3D cartoon characters,

inspire all production companies to develop their methods in film production. Currently, adults and children have enjoyed watching 3D movies without realizing the way these types of films are produced.

According to [159] there were several attempts to apply motion capture technology (MOCAP) by some photographers and producers in the nineteenth and twentieth centuries. Everyone placed his or her thumbprint on this developing technology. Because of that, there are certain major steps needing to be considered. First, preproduction is a part of the procedure that allows the designer to divide everything into parts and organize them before doing anything else. A project pipe signal is the second step which begins with preproduction and ends with post-production, which means the character in a game or animation is eventually where you would prefer the Motion Capture (MOCAP) data to go to. Cleaning and editing data are an important step which needs to be more concentrated and done with high proficiency alongside the last point, which is skeleton editing. However, there are other steps which may be applied to depend on the character shape and motion [159].

According to [160], Most of the old approaches depended on the 2D data such as pictures. Recently, the direction of development at the Time of Flight (ToF) cameras and other types of depth sensors became improved by creating opportunities to support this area. The survey presented the overview of traditional approaches which achieve human motion analysis involving depth and skeleton-based activity recognition such as facial expression detection, facial performance capture, head pose estimation, hand pose estimation and hand gesture recognition.

One of the primary factors in this research is the matching between the system and the real world which ensures that the system should use the users' movements; following real-world conventions, making information appear in a natural and logical order. The next section defines the subject of gesture recognition in detail in order to cover it accurately.

3.2 Definition of Gesture Recognition

In the present day, HCI is assuming greater significance in our daily lives. Gesture recognition can be named as a method as long this path [161, 162, 163]. Therefore, what is gesture recognition? In the previous section Gesture Recognition were defined as non-verbal motions used as a method of communication in HCI interfaces. Gestures are one of the significant aspects of HCI in both interpersonal and in the device inter-

faces. Another definition of gestures is physical movements or positions of a human's fingers, hands, arms or full body used to convert information. Gestures, in a virtual reality system can be used to navigate, control or interact with a computer [161, 162, 163]. The process by which gestures are formed in certain ways by a person is made known to a system is the main principle of gesture recognition. Signs can be expressed in a multitude of ways by gestures, for example, sign language used by hearing-impaired people. Other examples of gestures developed outside the computer field can be seen in use by traffic police, construction labours, and airport ground controllers. Gestures can be static, which means that the user adopts a pose, or dynamic where the motion is a gesture by itself. Attached devices such as gloves, data suits, Six Degrees of Freedom (6 DOF) trackers generally provide information along all the 3D geometries. For instance, hand and body gestures are used by pilots to direct aircraft operations aboard aircraft carriers [161, 162, 163].

Mathematical models based on hidden Markov chains, or methods based on soft computing can handle gesture recognition. The major advantage of using the hidden Markov model is the ability to recognize a variety of information for gesture recognition. Any applied implementation of gesture recognition needs the use of diverse imaging and tracking devices or tools such as data gloves, body suits, and marker based optical tracking. Pens, 2D keyboards, mice and oriented graphical user interfaces are frequently not appropriate to work in virtual systems unlike devices used to sense any part of body orientation and position, facial expression, sound and speech, skin response and other human behaviours or states which may be utilized to present communication between humans and the environment. Gestures may be static or dynamic or both in certain cases such as sign language. The automatic recognition of gestures needs their temporal segmentation, which usually requires specifying the start and end points of the gesture in terms of the frames of movement, in both time and space. Additionally, the preceding context also affects gestures alongside other gestures [161, 162, 163].

There are many aspects that have been successfully used for many gesture recognition systems such as computer vision and pattern recognition techniques, including feature extraction, clustering, classification and object recognition. Analysis and detection of texture, shape, motion, colour, image enhancement, optical flow, contour modelling and segmentation are image processing techniques that have been found to be effective. Gesture recognition uses connectionist methods, including multilayer perceptron, time delay of neural networks and radial basis function network [161, 162, 163].

Static gesture recognition may be achieved by neural networks template matching and standard pattern recognition. While the dynamic gesture recognition issue

includes the use of certain techniques such as Time Delay Neural Network (TDNN), dynamic time warping, Hidden Markov Models (HMMs), and time-compressing templates [161, 162, 163].

The last paragraph discussed the principles and background of some of the common tools used in gesture recognition. Essentially, human gestures generate a significant volume of motion uttered by the body, face, and hands. Recently, gestures have been classified into multi-categories; one of these is gesticulation, which is a spontaneous motion of the hands and arms with speech. This means it is combined into a spoken pronouncement, replacing a spoken word or phrase. Another category is a pantomime; gestures that represent objects or actions with or without associated speech. Emblem is also another gesture category which is a familiar gesture such as the V-sign which means victory, thumbs up means OK, and various other gestures. The sign language is a linguistic system for example, Universal Sign Language, which is defined very well. It can be defined as a visual language contains three main elements are finger spelling, word-level sign vocabulary and non-manual features. A spelling word letter by following the letter is a method used by finger spelling. The second element is utilized for common communication. Whereas non-manual features are composed of the body, facial expression, mouth and position of the tongue. One of the protentional fields in gesture recognition is sign language, which is totally useful for hearing-impaired people [161, 162, 163].

For example, robotics can be used to apply the sign language recognition using some suitable sensors used on the body of a patient. By analysing the received values from those sensors, robots can help inpatient therapy. Another example maybe stroke rehabilitation. Speech recognition has an ability to record speech and convey it as a text. There are certain types of gesture recognition tools which can record the symbols represented via the sign language into a text [161, 162, 163].

The operation inside each computer firstly is a gesture which represented as a region in some feature space. In this approach, the features will be X, Y and Z coordinates of different emphases on the human's body, with also the orientations of a few appendages. In an image-based approach, the sensor measures the intensity of a 2D grid of pixels. Generally, the picture is pre-processed, firstly to improve differentiates and to decrease the percentage of noise. The following feature extraction process localises the points around the picture, such as edges. Then, links all these processed features to form the illustration of full limits in the picture. Usually, the illustration limit is the source of a segmentation process which does the separation from each other in the regions matching to other parts of the human's body. After that is done, all positions and orientations

of the parts in the picture will be measured and the space of all positions and orientation would be the nominee for the feature space into which the regions of the gesture are well defined [161, 162, 163].

Sensing machines take each measurement, the computer attempts to recognize the gesture by locating the area in feature space inside which estimation falls and this process may be completed in pattern recognition or a neural network classifier [159]. These algorithms usually apply a simple model for the gesture areas. Gestures will be defined by storing a limited number of prototypes using algorithms. The provided prototype has a gesture region which will be well-defined as a set of points which are closer to the prototype than other stored prototypes. The prototype similar to the input vector is consistently detected by the recognition process. Algorithms are well known by the name of nearest neighbour search algorithms. For instance, neural network architecture is a superior network for recognition tasks. By using neural networks, the loads of the connections of the network will have stored prototypes. The recognition of a pattern is completed by an algorithm that joins one of the prototypes [159].

3.3 Types of Gesture Recognition

Gesture recognition has been introduced briefly in the previous sections [161, 162, 163]. The gestures are made by the user then are recognized by the receiver. It is for the meaningful body motions including movements of the fingers, hands, arms, head, face, or body to convey meaningful information. In this section, some essential type of gesture recognition is addressed briefly.

Hand gesture recognition is one of the understandable ways to generate a convenient, high adaptability interface between devices and users [161, 162, 163]. Using a series of finger and hand movements through the operation of complex machines is allowed by hand gesture recognition. This technique will eliminate the need for physical interaction between user and device [161]. The next section is introduced facial gesture recognition extensively.

As it is known, the face is a significant feature of a human. A human face is a non-rigid object with some variability in shape, size, texture, and colour. People can recognize and detect features in a scene easily with little or no effort at all. Since the substantial characteristic changes in the visual encouragement because of viewing conditions such as dissimilarity in facial expression, luminance, gender, ageing, interruptions or occlusions such as hair, glasses, hat or other camouflages [161, 162, 163].

Facial expressions include removing sensitive features from facial landmarks such as areas around the nose, mouth, and eyes of an image. Frequently, dynamic image frames of these areas are tracked to create appropriate features. Additionally, the dynamics, location and intensity of the facial actions are significant for identifying an expression and the concentration measurement of natural facial expressions is most often harder than that of posed facial expressions [161, 162, 163].
Facial gesture recognition is an additional technique for generating non-contact interface effectively between users and machines. The main target of facial gesture recognition for machines is to recognize emotions and other communication signs within humans, despite the countless physical variances between users [161, 162, 163].
The goal of face detection is similar to facial gesture recognition which is identifying and detecting human faces with efficiency despite their scales, positions, poses, locations and illuminations [161, 162, 163]. Low-bandwidth transmission of facial data, criminal identification, missing child recovery, surveillance, credit card verification, office security, telecommunication, video document retrieval, High-Definition Television (HDTV), human computer interfaces, medicine and multimedia facial queries are examples which require an automatic system for facial gesture recognition [161, 162, 163].

Automatic facial recognition has two main approaches, firstly analytics which is a flexible mathematical model developed to integrate illumination changes and facial deformation. Discrete local features such as irises and nostrils can be extracted for retrieving and recognizing faces [161, 162, 163]. It may also be implemented on these measurements using statistical pattern recognition techniques such as HMMs. There are other approaches used in facial recognition involve Wavelet Transform, active contour models and knowledge or rule-based techniques like facial action coding system. The second approach is holistic and involving grey-level template matching by using world-wide recognition. To represent the entire face template requires using feature vectors. Signaller discriminants, ANNs, PCA, optical flow, singular value and decomposition using eigen-faces are included in the holistic approach [161, 162, 163].

Many aspects need to be understandable such as overall muscle tension, hand tension, pupil dilation and locations of self-contact. To specify all these principles, the human body position, configure rations like angles, rotations and movement such as speeds need to be detected. All aspects may be completed through sensing devices attached to the user. The sensing devices may be magnetic field trackers, data gloves or body suits. Otherwise, using computer vision techniques and cameras can also be called sensing devices. An individual of sensing technology differs laterally, some including accuracy, dimensions, latency, resolution, user comfort, cost and range of motion. The

user is required to wear the device and carry cables connecting the device to a computer by using glove-based gestural interfaces [161, 162, 163].

Normally, there are many meanings of one or more gestures which are unclear for some people. For instance, the raising a hand and the waving of both hands over the head indicates the concept of 'stop'. It is not just using gestures that have different meanings between different people, speech and handwriting also have these differences. Furthermore, gestures are frequently obtained from language and have a cultural impact [161, 162, 163].They may be of these following types: hand and arm gestures are a recognition of a hand pose; sign language and applications used for entertaining, such as kids allowed to play and interact in a virtual reality environment; head and facial gestures, such as shaking or nodding of the head, opening the mouth to talk or looks of happiness, anger and fear, etc.; and lastly, body gestures are full-body movements such as understanding the movements of a dancer to create a match between music and graphics and tracking the motions of two humans interacting outside and more [161, 162, 163].

3.4 Overview of Hand Gesture Recognition

The hand is often well known as the most natural and instinctive interaction with humans' interaction. In the HCI world, an appropriate hand tracking is the first phase to develop instinctive HCI systems that may be used in applications such as virtual object manipulation, gaming and gesture recognition. Moreover, hand tracking is an interesting principle point which deals with three main parts of computer vision which are segmentation of hand, detection of hand parts, and tracking of the hand. Hand gestures are frequently the most expressive way and the most used in a gesture recognition system involving a posture is a static finger shape ration without hand motion and a gesture which is dynamic hand motion with or without finger movements [161, 162, 163].

A hand gesture requires tracking of 27 degrees of freedom of hand, including two major categories. A hand posture is a static hand pose without any movements; While hand motion is any movement of the hand, either the full hand or fingers. A hand movement consists of three major types are data gloves based, vision-based and electrical field sensing. Measuring the human body or body parts requires electrical field sensing, and this device is used officially to measure the distance of human hand or other body part from the device. Currently, most of the significant types almost all researchers are interested in studying, are data-glove-based and vision-based technologies. The data glove based is simply a glove that has multi-variety of sensors used to detect hand and

finger motions. There are many styles of data glove and each one has its uses, such as MIT Data Glove, CyberGlove III, CyberGlove II, Fifth Dimension Sensor Glove Ultra, X- IST Data Glove and P5 Glove. Each one of these types will be introduced in detail in Section 3.5 [162].

Recently, vision-based is one of the concepts requiring essential development. Simply, it is defined as to detect hand motion using a device such as cameras. Vision based mainly has two approaches which are used in a gesture recognition system; model-based and image-based techniques. The main definition of model-based is attempting to generate a 3D model of the human hand and use this model for recognition while images based are detecting a gesture by capturing pictures of the user's movement through the sequence of a gesture. Model-based, also called spatial gesture models, has two various categories which are 3D model based and appearance based both have diverse types. For example, the 3D model has skeletal and volumetric algorithms whereas appearance-based has deformable 2D templates and image sequences. Under volumetric algorithms, there are three other types of algorithms: Non-uniform rational basis spline (NURBS), super quadrics and primitives. The next section will provide more information regarding the vision-based concept. Recently, realtime in vision-based is more possible for an HCI system through the assistance of the latest developments in computer vision and pattern recognition field [161, 162, 163, 164].

s usual, there is no perfection in the hand gesture world. It means there are serious issues faced by researchers like, self-occlusion, hand deformation, irregular motion and appearance similarity making 3D hand tracking a challenging mission [157]. The proposed 3D hand tracking technique in this thesis can be used to extract accurate hand gesture features and enable the complex human-machine interaction such as gaming and virtual object manipulation [165].

3.5 Hand Gesture Recognition (Vision Based)

3.5.1 Overview of Vision Based Systems

Gestures recognition is one of the most natural communicative methods between human and computers in virtual environments [162]. Camera techniques are used to identify hand gestures. It started laterally in the early development of the first data gloves. The first computer vision gesture recognition system was reported in the 1980s. Moreover, vision-based recognition is normally natural and comfortable. As shown in Figure 3.1 [61], a flow diagram of a normal gesture recognition plan [163].

Using vision-based techniques require contending with other issues related to oc-

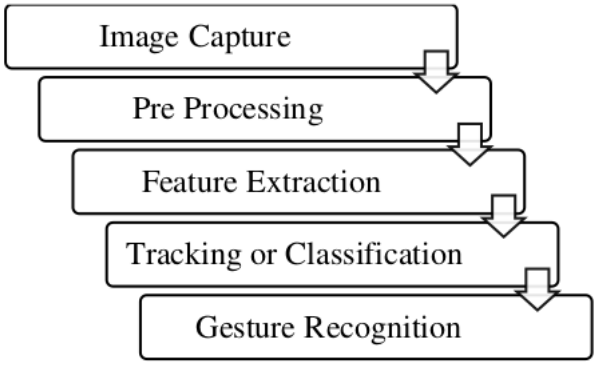

Fig. 3.1 Typical computer vision-based gesture recognition approach.

clusion of the user's body parts. Although tracking devices had the ability to detect movements of hands quickly while the human's body moving. Vision-based devices could grasp properties such as colour and texture for analysing a gesture, while typical tracking devices may not handle these [166].

Vision-based techniques may also be different between themselves in the number of cameras used, their speed and latency, the structure of the environment, like speed of movement and lighting, user requirements each user must wear something unique, the low-level features used, such as the region, edges, histogram, silhouette and moment; and nor 2D or 3D representation is used and either time is represented [166].

3.5.2 Types of Cameras

Currently, gestures are detected by different devices while cameras became the first device to detect most gestures. This section will introduce most of the current cameras used in the gesture recognition world. The type of cameras used in gesture recognition, with brief information for each type, is shown in Figure 3.2 [61].

As shown in Figure 3.3 [167],a stereo camera is a camera which has two lenses with almost the same distance separating them, such as human eyes. It takes two pictures at the same time. This copies the method that humans use to see, by and consequently generates the 3D effect when viewed. By using two cameras whose relations to one another are recognized, a 3D representation may be approached by the output of the cameras [168].

Figure 3.4 [169] shows the depth-aware cameras use cameras such as time-of-flight or structured light cameras. One could create a depth map of what is being seen by the camera at a short range. This data used to estimate a 3D representation of what is being viewed. These cameras may detect hand gestures effectively because of their

Fig. 3.2 Types of Cameras used in hand gesture recognition.

Fig. 3.3 Stereo Camera.

Fig. 3.4 Depth-aware camera.

short-range skills [168].

A thermal camera is an infrared camera that detects infrared radiation such as temperature from an object as shown in Figure 3.5 [170]. It converts the temperature of the object into an electronic signal to create a thermal picture on a screen; Or, to make temperature calculations on it. Infrared cameras can capture the temperature and can measure or quantify precisely. However, it is not efficient in detecting hand gestures like other cameras and is negatively affected by the weather. Therefore, the thermal behaviour may be observed but also the relative scale of temperature-related issues may be known and distinguished as shown in Figure 3.5 [168].

Controller-based gestures simulate a part of the body. Then, once gestures are made, some of their movements may be captured conveniently by software as shown in Figure 3.6 [171]. For example, the motion of a mouse device is connected to a sign which is being drawn by a person's hand. Another example is the Wii Remote which may learn the changes in acceleration over time to represent gestures [168].

Figure 3.7 [172], [173] shows the single camera which is defined as a normal camera that may be used for gesture recognition where the environment or resources would not be suitable for alternative forms of image-based recognition. A single camera may not be as effective as depth aware or stereo cameras despite a challenge to this concept by Flutter. This is an application that has been released which can be downloaded to Win-

Fig. 3.5 Thermal camera.

Fig. 3.6 Controller-based hand gesture.

Fig. 3.7 Single Camera.

Fig. 3.8 Holoscopic3Dcamera prototype by 3DVJVANT project at Brunel University.

dows or Mac computers with a webcam [168].

Figure 3.8 [174] shows the holoscopic 3D camera proposals, the easiest method to accomplish recording and replaying the light field 3D scene. The concept of this technique was proposed by Gabriel M. Lippmann in 1908. The innovative technology contains a microlens array architecture that proposed to double the spatial resolution of a holoscopic 3D camera horizontally by trading horizontal and vertical resolutions [175]. As shown in Figure 3.9 [176], The holoscopic camera can be in the form of a planar strength distribution, by using MLA [176, 175]. In spite of using the same features of holographic technique, it records the 3D image in 2D form and views it in complete 3D through an optical component, without the required bright light source and restrain dark fine. Moreover, it enables post-production processing such as refocusing [177].
[177].

Figures 3.10 [176] and 18 [175] show the description of the structure of Holoscopic 3D camera which are L0 = Nikon 35 mm F2 wide-angle lens, NF = Nikon F-mount, AP = adaptor plate, ER = 6 mm diameter extension rods, RM = ¡5arcminute accuracy rotation mount, MLA = plane of MLA, which is slanted in the process method, T0-T2

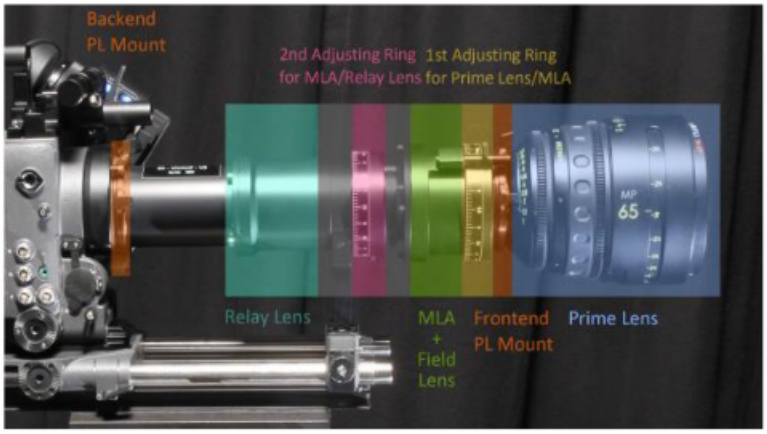

Fig. 3.9 3Dintegral Imaging camera PL: Prime lens, MLA: Microlens array, RL: Relay lens.

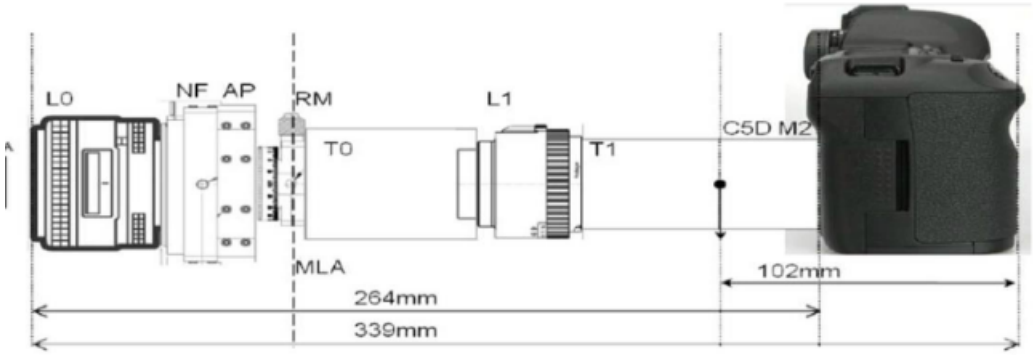

Fig. 3.10 Square Aperture Type 2 camera integration with canon 5.6k sensor.

= extension tubes, L1 = Rodagon 50 mm F2.8 relay lens $\times 1.89$, C5D M2 = Canon 5D Mark2 DSLR. Arrow displays the position of centre of gravity, SA = Square aperture mouthed to the L0.

3.6 Summary

This chapter introduced the background of HCI as a fundamental field of gesture recognition. HCI is how humans can interact with devices, and how computers respond to humans' requests. There are different types of HCI. One of them, called gestures represent a non-verbal way of interaction applied in user interfaces. Gestures are also defined as a physical movement of any part of the human body such as the finger, hand or full body. For instance, sign language is a model of gestures used by hearing-impaired people. An overview of hand gesture recognition is discussed in this chapter along with its types. Hand gestures are defined as a type of gesture recognition, which is how to move

a hand randomly and detect it by certain devices. It has two types, data gloves and computer vision. Data glove is a wired glove with linked sensors connected with fingers, or joints of the glove which is worn by a human. Many researchers invented different data gloves. Each data glove has a different purpose. For examples, MIT data glove is used in video games, body rehabilitation or sports training. However, the computer vision is the natural interaction way between humans and devices in a virtual environment. In the computer vision section, it also presents different cameras as types of devices used for detection, such as stereo cameras, depth cameras, thermal cameras, single cameras and a holoscopic imaging system camera were introduced. Next chapter presents a study of various image acquisition, including image acquisition techniques, data processing, followed by a brief discussion of image segmentation methods and features extraction.

CHAPTER 4

Methods

First, we started by presenting a study of various image acquisition , including image acquisition techniques. Then we come to the data processing, followed by a brief discussion of image segmentation methods, and then we get to extract the features with a focus on the most important methods and uses that are part of the one of our contributions to this work. Before we finish, we start the object recognition mission using deep learning and feature selection. Finally, the classification task is performed.

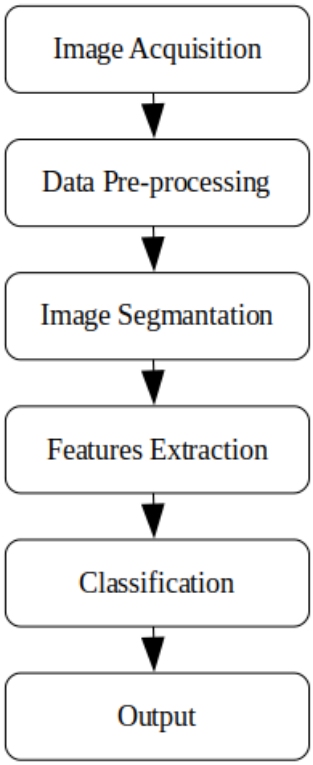

Fig. 4.1 Steps for Image Pre-Processing.

4.1 Image Acquisition

An image can be defined as a 2-D function f(x,y) where (x, y) is coordinate in two-dimensional space and f is the intensity of that coordinate. Each coordinate position is called as pixels. Pixel is the smallest unit of the image it is also called as picture elements or pel. So digital images are composed of pixels, each pixel represents the color (gray level for black and white images) at a single point in the image. Pixel is like a tiny dot of particular color. A digital image is a rectangular array of pixels also called as Bitmap.From the point of view of photography, the digital images are of two types [178, 179].

•Black and white Images

Black and white images are made of different shades of gray. These different shades lies between0 to 255, where 0 refers to black, 255 refers to white and intermediate values refer to different shades of black and white. Gray scale refers to the range of neutral tonal values (shades) from black to white.

•color Images

color images are made up of colored pixels.color can capture a much broader range of values than gray scale. "The spectrum – the band of colors produced when sunlight passes through a prism –includes billions of colors, of which the human eye can perceive seven to ten million." The electronic capture and display of color are complicated. RGB (Red, Green, and Blue) is the most commonly adopted color system.

The general aim of Image Acquisition is to transform an optical image (Real World Data) into an array of numerical data which could be later manipulated on a computer, before any video or image processing can commence an image must be captured by the camera and converted into a manageable entity [180]. The Image Acquisition process consists of three steps:

1.Optical system which focuses the energy

2.Energy reflected from the object of interest

3.A sensor which measures the amount of energy.

Image Acquisition is achieved by suitable cameras. We use different cameras for different applications. If we need an X-ray image, we use a camera (film) that is sensitive to X-ray. If we want infrared image, we use a camera which is sensitive to infrared radiation. For normal images (family pictures, etc.), we use cameras which are sensitive to the visual spectrum. Image Acquisition is the first step in any image processing system.

4.1.1 Image Acquisition Concept

In order to capture an image, a camera requires some sort of measurable energy. The energy of interest in this context is light or more generally electromagnetic waves. An

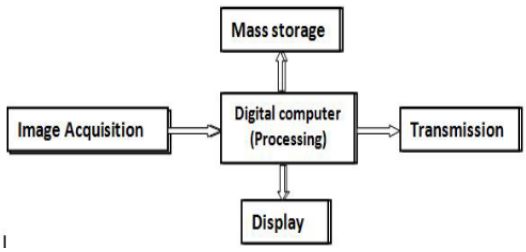

Fig. 4.2 Image Processing.

EM waves can be described as a massless entity, a photon, whose electric and magnetic field varies sinusoidal, hence the name waves. A photon can be described in three different ways:

1. A photon can be described By its energy E (measured in eV).
2. A photon can be described by its frequency f(H2).
3. A photon can be described by its wave length λ(m).

$$E = (hc)/(\lambda) \tag{4.1}$$

$$E = hf \tag{4.2}$$

4.1.2 Quantum Detectors

Quantum Detector is the most important mechanism of image sensing and acquisition it relies upon the energy of absorbed photons being used to promote electrons from their stable state to a higher state above an energy threshold. Whenever this occurs, the properties of that material get altered in some measurable way. Planck/Einstein came up with a relationship between λ of the incident photon and the E that it carries:

$$E = (hc)/\lambda \tag{4.3}$$

4.1.3 Image Acquisition Model

The images are generated by a combination of an illumination source and the reflection or absorption of energy by the elements of the scene being imaged. Illumination may be originated by radar, infrared energy sources, computer-generated energy patterns, ultrasound energy sources, X-ray energy sources, etc.
To sense the image, we use sensors according to the nature of illumination. The process of image sense is called image acquisition.
By the sensor, basically illumination energy is transformed into a digital image. The idea is that incoming illumination energy is transformed into a voltage by combination

of input electrical energy and sensor material that is responsive to the particular energy that is being detected. The output waveform is the response of sensors and this response is digitalized to obtain a digital image.
Image is represented by 2-D function f(x, y).Practically an image must be non-zero and finite quantity that is [181]:

$$0 < f(x, y) < \infty \tag{4.4}$$

•It is also discussed that for an image f(x, y),we have two factors:
•The amount of source illumination incident on the scene being imaged. Let us represent it by :

$$i(x, y) \tag{4.5}$$

The amount of illumination reflected or absorbed by the object in the scene. Let us represent it by:

$$r(x, y) \tag{4.6}$$

Then f(x, y) can be represented by :
f(x, y) = i(x, y).r(x, y)
Where

$$0 < i(x, y) < ” \tag{4.7}$$

It means illumination will be a non-zero and finite quantity and its quantity depends on illumination source. and

$$0 < r(x, y) < 1 \tag{4.8}$$

Here 0 indicates no reflection or total absorption and 1 means no absorption or total reflection.

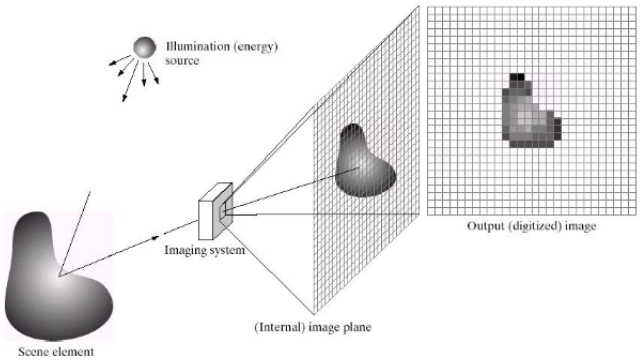

Fig. 4.3 Image Acquisition Model.

4.1.4 Techniques to Perform Image Acquisition

Image Acquisition process totally depends on the hardware system which may have a sensor that is again a hardware device. A sensor converts light into electrical charges. The sensor inside a camera measures the reflected energy by the scene being imaged. The image sensor employed by most digital cameras is a charge-coupled device (CCD). Some cameras use complementary metal oxide semiconductor (CMOS) technology instead [181]

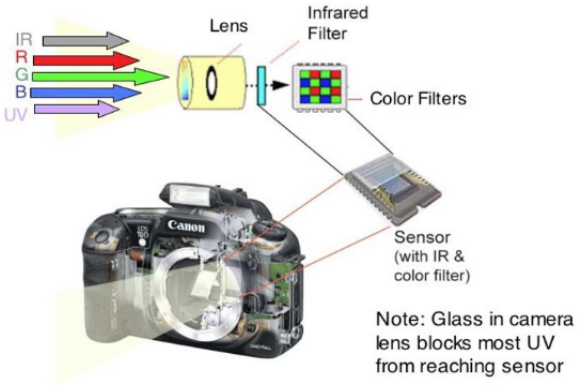

Fig. 4.4 Inside a Digital Camera.

Both CCD and CMOS image sensors convert light into electrons. A simplified way to think about these sensors is to think of a 2-D array of thousands or millions of tiny solar cells. (In this case the sensors are called photo sites.) Once the sensor converts the light into electrons, it reads the value(accumulated charge) of each cell in the image. A CCD transports the charge across the chip and reads it at one corner of the array. An analogue-to-digital converter (ADC) then turns each pixel's value into a digital value by measuring the amount of charge at each photo site and converting that measurement to binary form. CMOS devices use several transistors at each pixel to amplify and move the charge using more traditional wires.CCD sensors create high-quality, low-noise images. CMOS sensors are generally more susceptible to noise.
CMOS sensors traditionally consume little power. CCDs, on the other hand, use a process that consumes lots of power. CCDs consume as much as 100 times more power than an equivalent CMOS sensor. CCD sensors have been mass-produced for a longer period of time, so they are more mature. They tend to have higher quality pixels, and more of them [181].

4.2 Data Pre-processing

Data Preprocessing is the process of simply transforming raw data into understandable format. Real world data is sometimes incomplete, inconsistent, redundant and noisy.

Data preprocessing involves various steps that help to convert raw data into processed and sensible format.
The diagram below is used to depict the various steps involved in data preprocessing [182].

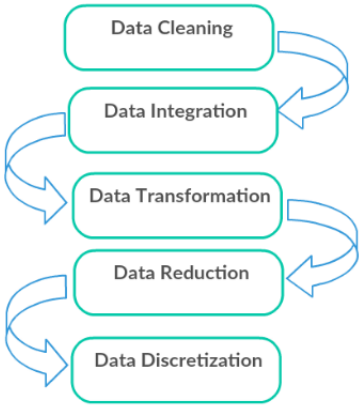

Fig. 4.5 Data Preprocessing Steps.

4.2.1 Data Cleaning

Data cleaning is the process of detecting corrupt data and inaccurate records from a record set or database table. The main use of cleaning step is based on detecting incomplete, inaccurate, inconsistent and irrelevant data and applying techniques to modify or delete this useless data [182].

4.2.2 Data Integration

Data Integration focuses on unification of data residing in different sources and presenting a unified view of these data. Data with different representations are put together and any conflicts resulting from it is resolved. This process becomes vital in a number of scientific and commercial applications. With increasing volume and exponential growth of data, integrating it becomes even more significant [182].

4.2.3 Data Transformation

Data transformation plays a pivotal role in converting unprocessed data into understandable form. It consists of data normalization, aggregation and generalizations. Data normalization helps to arrange the columns and tables of a database such that redundancy is minimum. This helps cut down on the processing time and complexity. Data aggregation helps in creating a brief summary for faster overview. The process of data

generalization is also known as rolling-up data. It helps in generalizing data and creates successive layers of summary in evaluation database [182].

4.2.4 Data Reduction

Data reduction is the process of transforming digital into ordered and simplified- form. This data is generally derived through empirical and experimental means. It involves reducing large amounts of data into smaller and meaningful fragments [182].

4.2.5 Data Discretization

Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals with minimal loss of information [182].

4.3 Segmentation

Since efficient hand tracking and segmentation is the key to success towards any gesture recognition, due to challenges of vision-based methods, such as varying lighting condition, complex background and skin color detection; variation in human skin color complexion required the robust development of algorithms for natural interface. color is a very powerful descriptor for object detection. So for the segmentation purpose color information was used, which is invariant to rotation and geometric variation of the hand [183]. Human perceives characteristics of color components such as brightness, saturation and hue components than the percentage of primary color red, green, and blue. color models are useful to specify a particular color in standard way. It is space-coordinated system within which any specified color represented by a single point. Here, three techniques were introduced using different color spaces for robust hand detection and segmentation. Hand tracking and segmentation (HTS) technique using HSV color space are identified for the reprocessing of HGR system.

4.3.1 Anticipated Static Gesture Set

Static gesture is a specific posture assigned to meaning. Following is the static gesture set specified for the proposed system with the specific meaning. application interface will be provided after recognition of specified posture for action. Simplicity and user-friendliness were taken into consideration for the design of anticipated posture set. For the American Sign Language movement, the centre of the hand gesture window was passed as a sign Language. Figure 4.6. shows anticipated static gestures set with defined tasks [184].

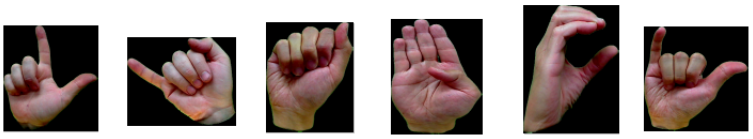

Fig. 4.6 a) sign L (b) sign J (c) sign A (d) sign B (e) sign C (f) sign Y .

4.3.2 Hand Segmentation Using HSV Color Space and Sampled Storage Approach

A Novel Approach for Image segmentation algorithm has been developed and tested for green color glove. In this approach, color based segmentation was attempted using HSV color space. The H, S and V separation was done using following equations [185].

$$V = max(R, G, B) \tag{4.9}$$

$$\delta = V - min(R, G, B) \tag{4.10}$$

$$S = \delta / V \tag{4.11}$$

To obtain value for hue following are the cases:
(i) if R=V then:

$$H = 1/6(G - B)/\delta \tag{4.12}$$

(ii) if G=V then:

$$H = 1/6(2 + (B - R)/\delta) \tag{4.13}$$

(iii) if B=V then:

$$H = 1/6(4 + (R - G)/\delta) \tag{4.14}$$

The input image of green color samples was passed to the algorithm and from H-S histogram the H-range = [0.4 0.55 0.6 0.6] and S-range = [0.2 1.0] were experimented for segmentation. The algorithm could be able to subtract dynamic background. Skin color samples needed to be passed to the algorithm for skin color detection. The drawback of this algorithm was training samples of the color need to be stored. It was sensitive to little variation in color brightness [185].

4.3.3 Hand Segmentation Using Lab Color Space (HSL)

The input captured RGB image was converted to lab color space. In CIE L* a* b* co-ordinate,where L* defines lightness, a* represent red/green value and b* denotes the blue/yellow color value. a* axis and +a direction shift towards red while along the b* axis +b movement shift toward yellow. Once the image gets converted into a* and b* plane, thresholding was done. Convolution operation was applied on binary images for the segmentation. Morphological processing was done to get the superior hand

shape. This algorithm was found to work for skin color detection but it was sensitive to complex background. Figure 4.7 shows the output with intermediate steps [184].

4.3.4 HSL Algorithm

i) Capture the Image.
ii) Read the input image.
iii) Convert RGB image into lab color space.
iv) Convert the color values in I into color structure specified in cform.
v) Compute the threshold value.
vi) Convert Intensity image into binary image.
vii) Performing morphological operations such as erosion.

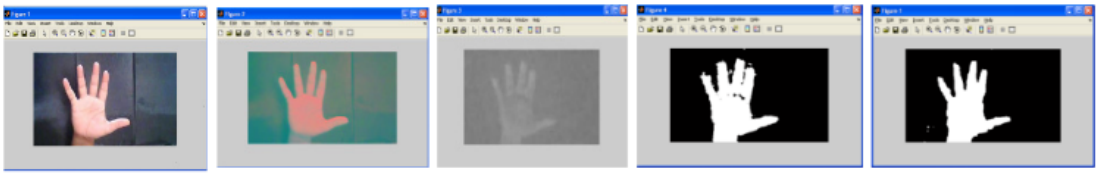

Fig. 4.7 (a) Input Image (b) rgb2clb (c) Gray Scale (d) Black and white (e) Image after erosio.

4.4 Machine Learning Approach

Supervised machine learning approach was used in this research because each image has an assigned label.

4.4.1 Artificial Intelligence

Research in Artificial Intelligence (AI) is very general and spans across numerous different areas such as mathematics, computer science, philosophy, economics and even ethics. This field is very wide and can be attempted by many different viewpoints. Therefore, this explanation will not be very exhaustive. For comprehensive and more complex description into the subject, refer to [186].
One of the many possible definitions of AI can be briefly explained as a search method to develop an artificially intelligent agent. In other words, it is an effort to create intelligent machines that are either intelligent or can be perceived as intelligent ones. One of the most important skills of intelligent agents is a sense of vision. A sense of vision is usually required for a positive degree and not always, it is necessary that it rivals the abilities of the human visual apparatus [146].

First, it tries to resolve the vision problems which were tackled from the so-called bottom-up approach in which the system was instructed with a hard-coded set of protocols describing the vision. It was expected that as the understanding of an instrument allowing humans to abstract information from the visual scene, the hard-coded systems could be fed this understanding and thus more skilled systems can be created. The problem with this method was that it highly underestimated the difficulty of reinforcement of these protocols. Therefore, it mainly failed to devise such a system.
This main idea why the investigators postulated was that in order to resolve the problem of deploying vision capabilities for the artificially intelligent system, it is compulsory to introduce a procedure that would allow AI systems to extract patterns from provided data. It is an overview of systems that can learn. A process that enables systems to learn is usually called machine learning.
Machine learning is again a relatively extensive term that can be used in different frameworks. In this work, it is meant to be understood as a technique that is used to create mathematical representations for image detection. There are numerous types of machine learning models that are useful for different tasks.
The most common task attempted, is called the task of classification, in which it classifies the occurrence of input into a correct discreet and mainly predetermined class. One most common type of machine learning task is called regression, which is based on the input data trying to estimate unknown continuous valued quantity [146].

4.4.2 Machine Learning

Machine Learning is a process that is used to create models that can abstract information from data to resolve a given problem and consequently repeatedly improves their performance. The interesting viewpoint that can be used to view machine learning as encoding information using fewer bits than the original representation, where the machine learning model is trying to get more details from input to evaluate model summary [146].

4.4.3 Machine Learning Approaches

There are mainly two types of machine learning approaches:

- Unsupervised Learning.
- Supervised Learning.

4.4.3.1 Unsupervised Learning

In the unsupervised learning method, the model is trained by detecting new data and find patterns in the data without being instructed on what they are. Contrasting to supervised

learning, the benefit of this method is that the model can learn from data without supervision. This means that there is no need for input data to be labelled. Therefore, it takes less time and resources to deploy these models in practice. The biggest difficulty of the supervised learning method in real-world applications is to obtain appropriate data. Appropriate data in this context means, data that were somewhat classified into different categories, which may not exist in all situations.
The mainstream of unsupervised learning procedures belongs to a group called clustering algorithms. These algorithms are centred on the idea to analyse ordered clustering of data in the input space to determine their relationship. This is achieved by the belief that data point clustering in input space is likely to exhibit similar properties. Illustrations of unsupervised learning models are:

- K-means -clustering model
- Self Organizing Maps (SOMs)
- Principal Component Analysis (PCA) - dimensionality reduction

Image classification usually does not depend on the use of unsupervised learning methods [146].

4.4.3.2 Supervised Learning

Supervised learning was used in this research as the dataset images have assigned labels. The supervised learning method is more commonly used when data is known. This method needs training data with a specific format. Each instance must have assigned label. These labels make available supervision for the learning algorithm. The training process of supervised learning is constructed on the following principle. First, the training data are fed into the model to produce estimates of output. This estimate is compared to the assigned label of the training data in order to evaluate the model error. Based on this error, the learning algorithm distorts the model's parameters in order to reduce it [146].

4.4.4 Artificial Neural Network

ANN is defined as an interconnected assembly of nodes like the neural structure of the human brain and can solve different types of problems in an easy manner. The brain works by learning from experiences [187, 188]. ANN is a system that processes information in a similar manner to the biological nervous system. The key aspect of this system is the unique structure of the information processing system [187, 188]. The system is composed of a large number of unified processing elements working together to solve certain issues [187, 188].It is specifically configured for data classification or

pattern recognition applications via learning processes. The architecture of an ANN is composed of three main layers including an input layer, the hidden layer (one layer or more) and the output layer.
ANN can be trained to use a supervised or unsupervised approach. In a supervised approach, ANN is simply trained by matched input and output while the unsupervised approach is an attempt to obtain the ANN to realize the structure of input data. [187, 188]. There are several benefits associated with using ANN such as self-learning and large data handling. The advantage of using an ANN is ANN has the ability to learn and train data models for non-linear and complicated relationships. Different applications may be used by an ANN such as image processing, object detection and forecasting [187, 188].

4.4.5 Deep learning

Deep learning is a machine learning based model that instructs systems to perform the task humans likely to do [189, 190]. For instance, deep learning is the basic technology behind the automated cars, helping them to sense the traffic signals and pedestrians. It is also the main idea behind the recognition of audio and voices in different devices such cell phones and tablets. Deep learning becoming famous because it is doing the tasks which could not be performed earlier. A deep learning model is based on the layers of the data which could be pictures, text, or audio, into different and small classification layers. Artificial Intelligence could provide 100% accurate results with close to the human level accuracy and even exceeding the human pace. These models are trained by using large datasets and machine learning techniques such as CNN or ANN which contains many classification layers [189, 190].

In machine learning techniques, the system would guide how to use the model accurately on the graphics, audios, and text. Deep learning models give precision based accurate results even exceeding the human level. These models are framed according to the data given and transforming that data into artificial neural based systems containing large layers of classified data. Deep learning attains more precision and accuracy ever than before which help it to meet the users' expectations. It is used in useful applications such as automated cars. advances in the past years have shown that artificial intelligence can even surpass the humans in classifying images [189, 190].

Deep learning needs large amount of classified data. For instance, developing automated cars hundreds of thousands of images and videos. Deep learning requires an excessive amount of power. Elite GPUs have an equal design that is proficient for deep

learning. Cloud computing and clusters, when combined takes less time as compared before when it took weeks [189, 190].

As deep learning is consisting of neural networks, so it is also known as deep neural networks. The expression "deep" typically mentions to the quantity of concealed layers in the neural system. Usually neural networks just contain 2-3 concealed layers, while deep systems can have up to of 150 layers. To implement the deep learning models, they must train them. For training these models, they need a large number of labelled data and neural networks. This will help them to learn the features directly from data without any kind of human interaction.
The CNN is one of the most famous deep neural networks algorithms. It stands for Conventional Neural Network. It involves classified layers of input data and uses 2D convolutional layers to process 2D data [189, 190].

Contrastive Divergence (CD) algorithm is different training methods to approximate Maximum- Likelihood (ML) learning algorithms which represents the relationship between weights and its error, and it called the gradient. This method implemented to learn the weight of the Restricted Boltzmann Machines (RBMs) with a gradient ascent. The formula of this method is shown as follows:

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \eta \frac{\partial log(p(v))}{\partial w_{ij}} \tag{4.15}$$

The P(v) is the probability of the visible vector which is given by:

$$p(v) = \frac{\Sigma_k exp(-E(v,h))}{z} \tag{4.16}$$

The $E(v,h)$ is the energy function allocated to the state of the network which is given by $E(v,h) = -v^T W h$.$\frac{\partial log(p(v))}{\partial w_{ij}}$has the simple form of $\langle v_i h_j \rangle_{data^-} \langle v_i h_j \rangle_{model}.\langle ... \rangle_p$ signifies an average with respect to distribution p. Sampling $\langle v_i h_j \rangle_{model}$ requires alternating Gibbs sampling for a long time. The CD changed this phase by running alternating Gibbs sampling for n phases. After n phases, the data are sampled, and that sampled data is utilised in place of $\langle v_i h_j \rangle_{model}$.

Deep learning applications are utilized in projects from computerized heading, to clinical devices [189, 190] . Automobiles companies are using machine learning mod-

els identify traffic signs, etc. Due to the use of deep learning accidents of walking people has significantly decreased. In aerospace and defence deep learning is utilized to recognize objects from satellites that find special regions and distinguish guarded or unguarded areas for troops. Cancer analysts are trying to identify malignant growth cells using deep learning models and artificial intelligence. UCLA teams have manufactured a microscope that includes a high-dimensional informational set used to train a deep learning application to precisely recognize cancer cells. Use of deep learning models helps workers in their field area where there is heavy machinery by identifying people and things in the safe and unsafe zones. Deep learning is being used in recognizing the audio and the voice, such as the devices that detect your speech and give the results according to it. These all functions are done by deep learning [189, 190] .

4.4.6 Convolutional Neural Network

A convolutional neural network (CNN) is a type of artificial neural network specifically designed for image recognition [191, 192, 193]. A neural network following the activity of human brain neurons is a patterned hardware and/or software system. CNN is also defined as a different type of multi-layer neural network and each layer of CNN converts one number of activation to another through a function. CNN is a special architecture used for deep learning [191, 192, 193].CNN is frequently used in recognizing scenes and objects, and to carry out image detection, extraction and segmentation.

CNN can be categorized in two phases, namely Training and Inference. To build a CNN-based architecture, it applies three key types of layers: Convolutional Layer, Pooling Layer and the Fully Connected Layer. The first layer is a convolutional layer which is the main block of CNN. It takes many filters that are applied to the given image and create different activation features in the picture. The second layer is pooling, which is used to down samples. It will obtain input from non-linear activation and the output will depend on the window size. The last layer is fully connected where a target is identified to determine the category of final output. Due to the three layers, which removes the necessity for feature extraction by using image processing tools, the image data is learned directly by CNN. CNN causes the recognition results to be unique and it might be retrained easily for new recognition missions while it is allowed to build on the pre-existing network. All the following factors have made the usage of CNN significant in the last few years [191, 192, 193].

If a correct filter is applied to the temporal and spatial dependency in an image, it can be effectively captured by CNN. The number of parameters (weights) will in-

crease rapidly in a neural network with fully connected neurons as the size of the input increases [191, 192, 193]. A convolutional neural network reduces the number of parameters with fewer connections, mutual weights and down sampling [191, 192, 193]. Weight sharing is another major feature of CNNs. CNNs are an efficient extractor for a completely new task or for problems in photo performance, text, audio, video recognition and classification functions. A convolutional neural network also reduces the number of parameters with fewer connections, mutual weights, and down sampling. Besides that, CNNs remove the need for manual processing of features then discovers the features directly [191, 192, 193].

CNN Algorithm contains convolutional layers that are represented by an input called map I, many filters K and biases b. In the images case , it may have as input which is an image with height H, width W and C = 3 channels which are red, blue and green such that $I \in R^{H*W*C}$ Consequently for many D filters will have $K \in R^{k1*k2*C*D}$ and biases $b \in R^{D}$, one for each filter. The output from this convolution process is shown as follows:

$$(I * K)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} \sum_{c=0}^{c} K_{m,n,c} * I_{i+m,j+n,c} + b \tag{4.17}$$

The convolution procedure implemented previously is the same as the cross-correlation,exclude that the kernel is flipped horizontally and vertically.For simplicity purposes, it should utilize the argument where the input image is gray scale such as single-channel C=1.The equation (4.17) will be transformed as follows.

$$(I * K)_{ij} = \sum_{m=0}^{k_1-1} \sum_{n=0}^{k_2-1} K_{m,n} * I_{i+m,j+n} + b \tag{4.18}$$

Search engines, recommender systems and social media are the primary fields to use a CNN in identification and classification of objects. Social media, identification procedures and surveillance are using face recognition which is worth mentioning separately [191, 192, 193].This image recognition section involves more complex images such as pictures that could have human or other living beings, including animals, fish and insects.A banking insurance using optical character recognition has been designed to process symbols that are written and printed [191, 192, 193]. The medical image involves a whole lot of additional data analysis that will spur the initial recognition of the image. A CNN medical image classification detects microorganisms with higher accuracy than the human eye on the X-ray or MRI images. Drug discovery is another

important area of health care that uses CNNs extensively. CNN is one of the most innovative implementations used in various fields [191, 192, 193].

4.4.7 Structure of Machine Learning Algorithm

Although there are numerous machine learning algorithms, all have a common structure that can be generalized. Structure of nearly all machine learning algorithms can be defined as composed of the following components:

- Dataset description
- Model
- Cost function
- Optimization technique

Almost all supervised learning algorithms use the same dataset description. The other three components can vary intensely. This level of analysis is suitable for developing the intuition for Neural Networks (NNs) and a description of its individual components [146].

4.4.7.1 Dataset description

Supervised learning requires datasets with detailed properties. Each dataset holds a set of n instances which contain a pair of the input vector x_i and output scalar y_i. Input vector

$$x_i^T = [x_1, x_2, ..., x_p] \tag{4.19}$$

Specific components of the input vector must be of a uniform type. In the case of the image as input data, it is the value of individual pixels (e.g. 0-255). In other cases, they could be real values. As a general rule, input data should be normalized. This holds in images automatically since each pixel must have its values in a fixed range. It is very important in other types data, where this is not guaranteed.
Output scalar y_i characterizes a class of the given instance. The type of this output value thus must obtain only certain values. To put it differently, it must be a set of cardinalities equal to the number of all possible classes[146].

4.4.7.2 Dataset description

The model is predicted by taking input x_i to predict values of its output y_i. Each model has parameters represented by vector θ, which are used during the training process.

The simplest example of the model type is a linear model, also called linear regression. Parameters θ of this model are:

$$\theta^T = [\theta_1, \theta_2, ..., \theta_p] \tag{4.20}$$

Where p is the number of parameters equal to the size of the input vector x_i.
Prediction $\hat{y}$ of the model on instance is computed as

$$\hat{y}_i = \sum_{j=1}^{p} x_{ij}\theta_j \tag{4.21}$$

Estimate of the model on the entire dataset in matrix notation is given by:

$$\hat{y} = X\theta \tag{4.22}$$

Estimates in expanded notation are given as:

$$\begin{pmatrix} \hat{y}_1 \\ . \\ \hat{y}_n \end{pmatrix} = \begin{pmatrix} X_{11} & . & X_{1p} \\ . & . & . \\ X_{n1} & . & X_{np} \end{pmatrix} \begin{pmatrix} \phi_1 \\ . \\ \phi_n \end{pmatrix} \tag{4.23}$$

4.4.7.3 Cost Function

To achieve the learning accuracy of the machine-learning algorithm, it is necessary to approximate the error of its predictions. This is assessed with so-called cost function and called loss function. This function must have specific properties. The ability of the machine-learning algorithm to learn rests on the approximation of its improvement with the change of its parameters. Therefore, cost function must be at least partially differentiable. In the case of linear regression, it is most common to use sum of the square error. The main aim is that derivative of this function for a linear model has only one global minimum [146].
The cost function is defined as:

$$J(\theta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - x_i^T\theta)^2 \tag{4.24}$$

For the optimization determinations, it is usually useful to express the cost function in matrix notation

$$J(\theta) = (y - X\theta)^T(y - X\theta) \tag{4.25}$$

4.4.7.4 Optimization technique

The last part of the learning algorithm is the optimization technique. It consists of an update of the model's parameters θ in order to increase prediction accuracy. In other words, to find θ such that the value of cost function $J(\theta)$ for given dataset is as small as possible.
To examine the change of the cost function on given dataset, it is necessary to compute the derivative of $J(\theta)$ with respect to θ

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{\partial[(y_i - X\theta)^T(y_i - X\theta)]}{\partial \theta} \tag{4.26}$$

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{\partial[(y^T y + \theta^T X^T X\theta - 2y^T X\theta]}{\partial \theta} \tag{4.27}$$

$$\frac{\partial J(\theta)}{\partial \theta} = 2X^T X\theta - 2X^T y. \tag{4.28}$$

For the linear model, it is possible to find an optimal result which is a global minimum of the cost function. The optimal result

$$\theta = (X^T X)^{-1} X^T y. \tag{4.29}$$

is found by comparing the partial derivative of $J(\theta)$ to 0. The only condition is that $X^T X$ must be non-singular.
Unfortunately, only very simple problems can be handled using the model as simple as linear regression. The more complex model usually means more complex cost function. The optimization process of more complex cost functions cannot ensure the obtaining of the global minimum. In this case, the optimization technique must have an iterative character. To put it in a dissimilar way, the algorithm must be a method with minimum iterations. Many of the iterative approaches belong to the group called gradient-based optimization [146].

4.4.8 Model Complexity

In the first calculation, it could be said that the task of supervised machine learning is to model the relationship between input and output data most correctly. The problem with this definition is that in the real-world application, there have never been enough data to capture the true relationship between the two. Therefore, the task of machine learning is the attempt to suppose the true relationship by detecting an incomplete picture.
Hence the most significant property of the machine learning model is its generalization capability. That is the capability to produce meaningful outcomes from data that were not previously detected.
Generalization ability is reliant on the complexity of the model and its relationship to

the complexity of the underlying problem. When the model does not capture the complexity of the problem appropriately, it is defined as underfitting.In case the complexity of models surpasses the complexity of the underlying problem, then this phenomenon is called overfitting [146].
In both extremes, the generalization ability suffers. In the earlier case, the model is unable to capture true intricacies of the problem and therefore is unable to predict wanted output reliably. In the last case, it tries to capture even the subtlest data perturbation that might be in fact an outcome of the stochastic nature of the problem and not the underlying real relationship. This can also cause the fact that input data is lost some variable necessary to capture the true relationship. This fact is inescapable, and it thus must be careful when designing a machine learning model.Representation of this phenomenon in the case of two variable inputs is shown in Figure 4.8.

Typically, the machine learning model is trained on as much input data as possible

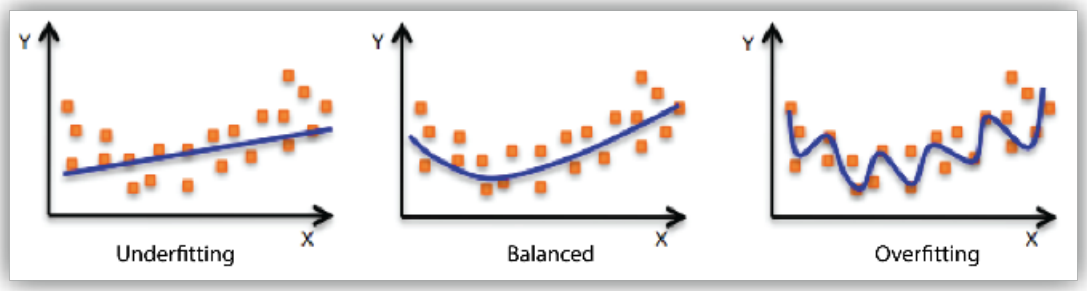

Fig. 4.8 Figure shows different levels of generalization of the model.

in order to reach the best possible performance. At the same time, its error rate must be verified with independent input data to check whether the generalization ability is not deteriorating. This is typically accomplished by splitting available input data into training and testing set, frequently in 4:1 fraction of training to test data. The model is trained with training data only and the presentation of the model tests on the test data. A connection between test and train error can be found in Figure 4.9. Even though the true generalization error can never be truly detected, its estimate of the test error rate is enough for most machine learning tasks [146].

4.4.8.1 Regularization

Regularization is any alteration that is made to the learning algorithm that is intended to decrease its generalization error, but not its training error [194]. As it has already been stated, the most significant aspect of machine learning is striking the stability between over and under fitting of the model. To support this problematic concept of regularization was devised. It is a method that helps to penalize the model for its complexity.

The basic idea consists of adding a term in the cost function that increases with model complexity. When this is applied to cost function from equation 4.30 [146].

$$J(\theta) = (y - X\theta)^T(y - X\theta) + \lambda\theta^T\theta \tag{4.30}$$

where λ is a parameter that controls the strong point of the preference [194].

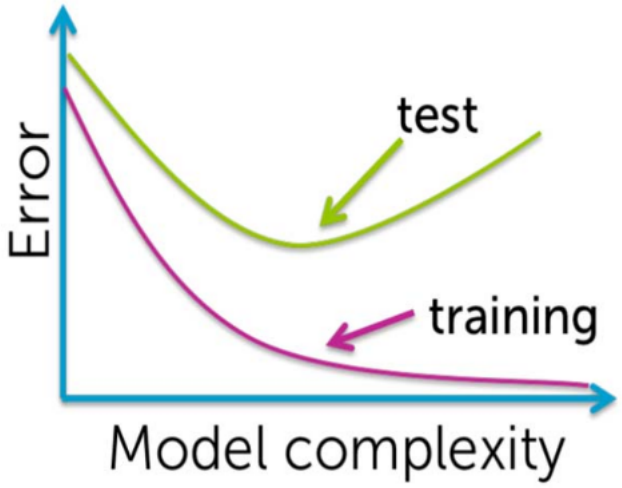

Fig. 4.9 Relationship between the model complexity and its ultimate accuracy is the relationship between training and testing error.

4.5 Classification

Classification is a process of categorizing a given set of data into classes. It can be performed on both structured or unstructured data. The process starts with predicting the Class of a given data point. The classes are often referred to as a target, label or categories.

The classification predictive modelling is the task of approximating the mapping function from input variables to discrete output variables. The main goal is to identify which class/category the new data will fall into [195].

Classification Terminologies in Machine Learning [195]:

- **Classifier** – It is an algorithm that is used to map the input data to a specific category.
- **Classification Model** – The model predicts or draws a conclusion to the input data given for training, it will predict the class or category for the data.
- **Feature** – A feature is an individual measurable property of the phenomenon being observed.

- **Binary Classification** – It is a type of classification with two outcomes, for example – either true or false.
- **Multi-Class Classification** – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- **Multi-label Classification** – This is a type of classification where each sample is assigned to a set of labels or targets.
- **Initialize** – It is to assign the classifier to be used for the
- **Train the Classifier** – Each classifier in sci-kit learn uses the fit(X, y) method to fit the model for training the train X and train label y.
- **Predict the Target** – For an unlabelled observation X, the predict(X) method returns predicted label y.
- **Evaluate** – This basically means the evaluation of the model i.e. classification report, accuracy score, etc.

Types Of Learners In Classification:

Lazy Learners – Lazy learners simply store the training data and wait until a testing data appears. The classification is done using the most related data in the stored training data. They have more predicting time compared to eager learners. Eg – k-nearest neighbour, case-based reasoning [195].

Eager Learners – Eager learners construct a classification model based on a given training data before getting data for predictions. It must be able to commit to a single hypothesis that will work for the entire space. Due to this, they take a lot of time in training and less time for a prediction. Eg – Decision Tree, Naive Bayes, Artificial Neural Networks [195].

4.5.1 Classification Algorithms

In machine learning, classification is a supervised learning concept which basically categorizes a set of data into classes. The most common classification problems are – speech recognition,gestures recognition, face detection, handwriting recognition, document classification, etc. It can be either a binary classification problem or a multi-class problem too. There are a bunch of machine learning algorithms for classification in machine learning. Let us take a look at those classification algorithms in machine learning [195].

4.5.1.1 Random Forest

Random decision trees or random forest are an ensemble learning method for classification, regression, etc. It operates by constructing a multitude of decision trees at training time and outputs the class that is the mode of the classes or classification or mean prediction(regression) of the individual trees [195].

A random forest is a meta-estimator that fits a number of trees on various subsamples

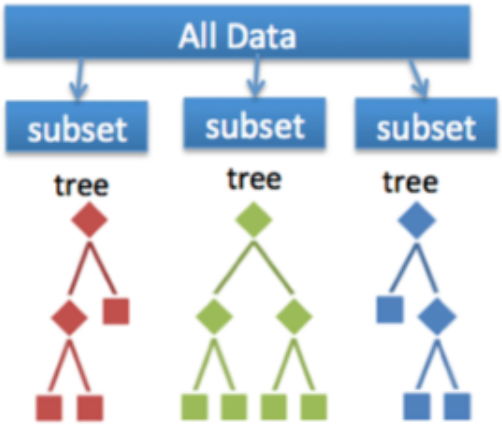

Fig. 4.10 Random Forest.

of datasets and then uses an average to improve the accuracy of the predictive nature of the model. The sub-sample size is always the same as that of the original input size but the samples are often drawn with replacements [195].

Advantages and Disadvantages

The advantage of the random forest is that it is more accurate than the decision trees due to the reduction in the overfitting. The only disadvantage with the random forest classifiers is that it is quite complex in implementation and gets pretty slow in real-time prediction [195].

Use Cases:

- Industrial applications such as finding if a loan applicant is high-risk or low-risk
- For Predicting the failure of mechanical parts in automobile engines
- Predicting social media share scores
- Performance scores

4.5.1.2 Artificial Neural Networks

A neural network consists of neurons that are arranged in layers, they take some input vector and convert it into an output. The process involves each neuron taking input and

applying a function which is often a non-linear function to it and then passes the output to the next layer [195].

In general, the network is supposed to be feed-forward meaning that the unit or neuron

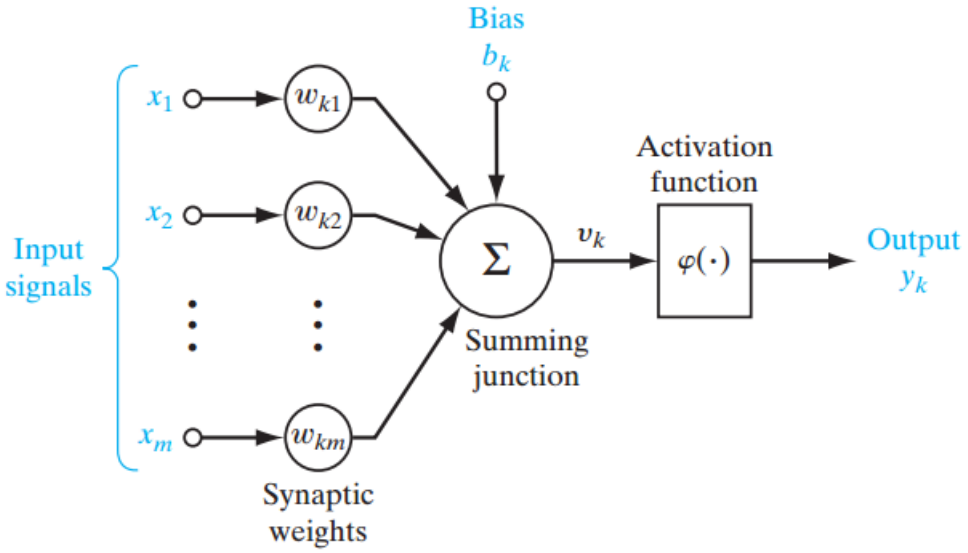

Fig. 4.11 Artificial Neural Networks.

feeds the output to the next layer but there is no involvement of any feedback to the previous layer.

Weighing are applied to the signals passing from one layer to the other, and these are the weighting that are tuned in the training phase to adapt a neural network for any problem statement [195].

Advantages and Disadvantages

It has a high tolerance to noisy data and able to classify untrained patterns, it performs better with continuous-valued inputs and outputs. The disadvantage with the artificial neural networks is that it has poor interpretation compared to other models [195].

Use Cases:

- Handwriting analysis
- Colorization of black and white images
- Computer vision processes
- Captioning photos based on facial features

4.5.1.3 Support Vector Machine

The support vector machine is a classifier that represents the training data as points in space separated into categories by a gap as wide as possible. New points are then added

to space by predicting which category they fall into and which space they will belong to [195].

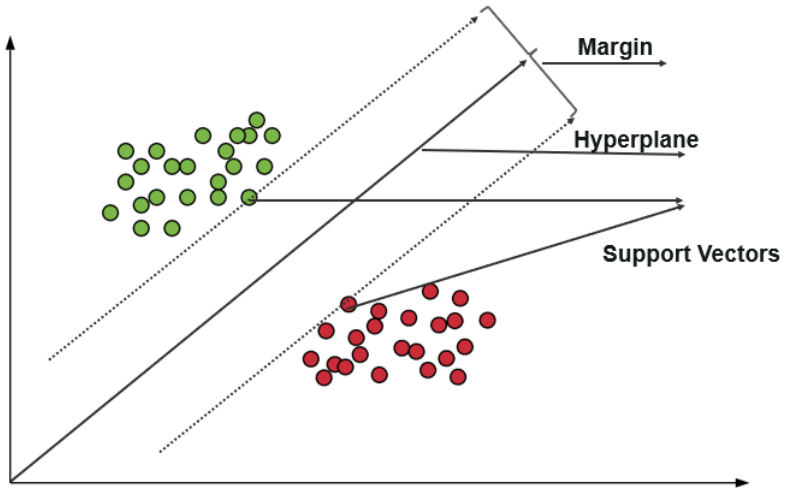

Fig. 4.12 Support Vector Machine.

Advantages and Disadvantages

It uses a subset of training points in the decision function which makes it memory efficient and is highly effective in high-dimensional spaces. The only disadvantage with the support vector machine is that the algorithm does not directly provide probability estimates [195].

Use cases:

- Business applications for comparing the performance of a stock over a period of time
- Investment suggestions
- Classification of applications requiring accuracy and efficiency

CHAPTER 5

Results and Discussions

Hand gestures provide humans a convenient way to interact with computers and many applications. However, factors such as the complexity of hand gesture models, differences in hand size and position, and other factors can affect the performance of the recognition and classification algorithms. Some developments of deep learning such as Convolutional Neural Networks (CNN) and Capsule Networks (CapsNets) have been proposed to improve the performance of image recognition systems in this particular field. While CNNs are undoubtedly the most widely used networks for object detection and image classification, CapsNets emerged to solve part of the limitations of the former. For this reason, in this work a particular ensemble of both networks is proposed to solve the American Sign Language recognition problem very effectively. The method is based on increasing diversity in both the model and the dataset. Obtained results show that the proposed ensemble model together with a simple data augmentation process produces a very competitive accuracy performance with all considered datasets.

5.1 Datasets description

The ASL serves as the predominant sign language of deaf communities in the United States and part of Canada. Commonly, all ASL datasets, and in particular, those used here, include at least alphabetic letters from A to Z, but applying a special consideration over signs J and Z, because their corresponding gesture are dynamic and require motion. Usually J and Z were excluded directly from the dataset, alternatively were acquired in some particular conditions to avoid confusion with other static signs. In this work four public ASL image datasets were used to evaluate several classification models over the hand gestures problem:

1. Massey University dataset [196]. The dataset contains 1,815 images of ASL gestures acquired from 5 subjects, with varying lighting conditions and hand positions. Originally the dataset comprises 36 different alphanumeric classes, we used the letter-specific classes only. Signs J and Z were rotated slightly to differentiate them from others that might be similar. Figure 5.1 shows the ASL alphabet.

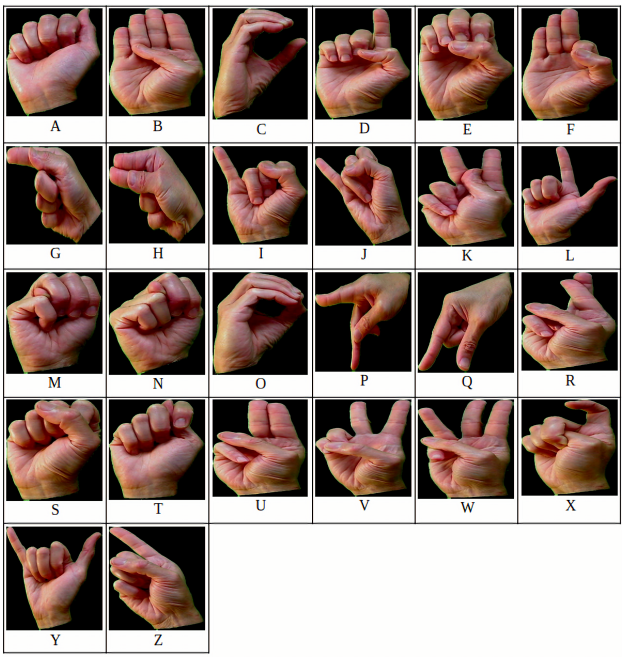

Fig. 5.1 Massey University dataset.

2. Static Hand Gesture ASL dataset [197]. The dataset contains 860 images that were built by capturing the static gestures of ASL alphabet from 8 people, except for the letters J and Z. To increase the dataset diversity the authors applied 30-degree rotations and 20% of vertical scaling. The final dataset comprises 24 gestures and a total of 4,848 samples. Figure 5.2 shows the ASL alphabet.

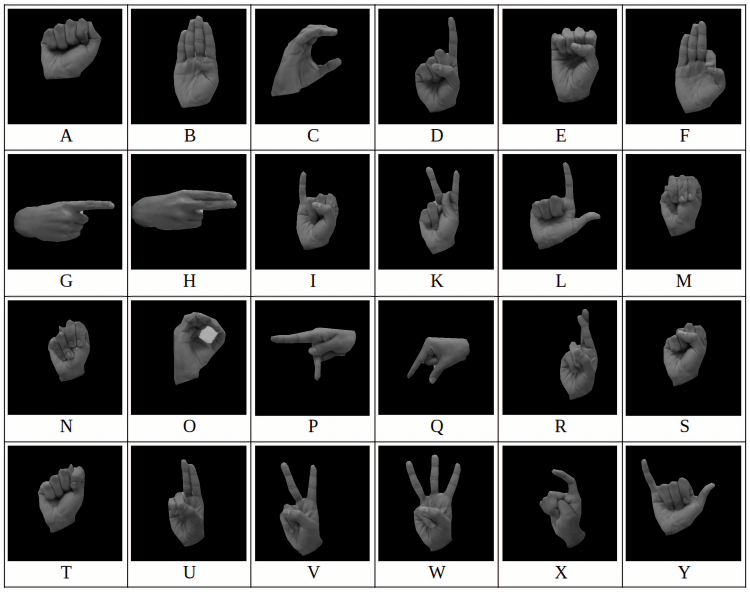

Fig. 5.2 Static Hand Gesture ASL dataset.

3. Kaggle ASL Alphabet dataset [198]. The dataset contains 78,000 hand gesture images. There are 29 classes, of which 26 are for the letters A-Z and 3 classes

not considered in this work for SPACE, DELETE and NOTHING. Signs J and Z were rotated slightly to differentiate them from others that might be similar. Figure 5.3 shows the ASL alphabet.

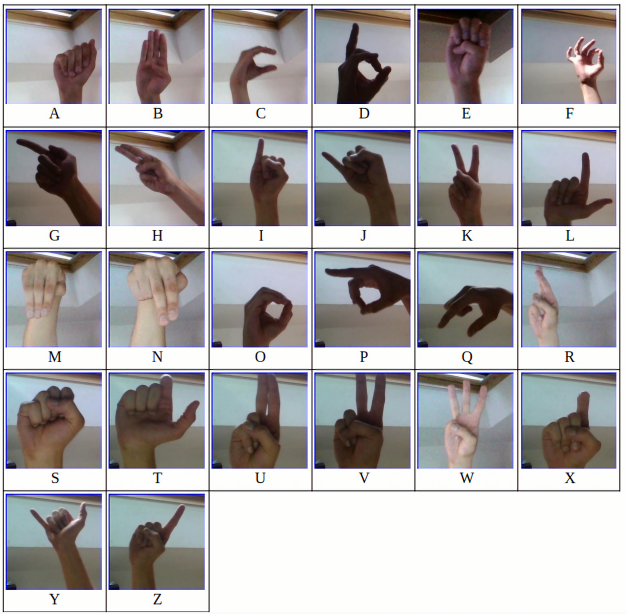

Fig. 5.3 Kaggle ASL Alphabet dataset.

4. MNIST ASL dataset [199]. This dataset is composed by 34,627 images of the alphabetic letter from A to Z. Letters J and Z were no captured in the dataset. The original dataset is divided into two sets, one for training (27,455 cases) and one for testing (7,172 cases). Figure 5.4 shows the ASL alphabet.

Fig. 5.4 MNIST ASL dataset.

The first and second datasets are mainly selected owing to their diversity and for comparison with other authors. The third dataset was chosen due to its noisy nature in order

to test the robustness of the proposed sign recognition method. This dataset contains images with changing background and illumination conditions, as well as different hand sizes, locations and rotations. In the case of the fourth dataset, several works have also been evaluated with this public dataset, and some of them with similar initial approach to ours.

Table 5.1 Number of samples of the original, training, test, and augmented-training sets for the datasets. The original sets are divided into two sets (training and test) according to the percentage (%) shown.

Dataset	Original	Training / %	Test / %	Augmented-Training
Massey University	1815	1415 / 80	364 / 20	15961
Static Hand Gesture ASL dataset	960	672 / 70	288 / 30	7392
Kaggle ASL Alphabet dataset	78000	60632 / 78	17368 / 22	666952
MNIST ASL dataset	34627	27455 / 80	7172 / 20	302005

In all below experiments[1], the original sets are divided into training and test sets according to the percentage shown in Table 5.1. The optimization was performed by the Stochastic Gradient Descent method of Adam algorithm, taking batches of 128 images. An augmented training dataset was also generated in order to improve the networks performance in operation mode. According to the data augmentation mechanism described in Section 5.5, 14510 / 6720 / 606320 / 274550 augmented hand gesture images was generated and added to the original training datasets, Massey University, Static Hand Gesture ASL, Kaggle ASL Alphabet and MNIST ASL datasets, respectively.

5.2 Features Extraction

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large datasets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.

The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learn-

[1]The source code that allows to reproduce all the experiments of this work can be accessed from `https://github.com/bousbai/Improving_Hand_Gestures_Recognition`

ing process.

Using Regularization could certainly help reduce the risk of overfitting, but using instead Feature Extraction techniques can also lead to other types of advantages such as:
Accuracy improvements.
Overfitting risk reduction.
Speed up in training.
Improved Data Visualization.
Increase in explainability of our model.

In machine learning, diversity is an extremely significant concept because, in addition to raising accuracy, it also has a greater ability of generalization acting as a regularization element of learning. Diversity can be achieved in several ways. For example, it can be introduced in the training data, such that it provides more discriminative information for the model. Diversity can also be obtained by the combination of the outputs of a model trained several times. A third option for introducing diversity is to combine the outputs of several types of models. This method is called ensemble learning [2]. Ensemble learning was initially used for classification [3] although it can also be used in other fields as regression [4] and feature selection [5] among others. The outputs of models can be combined in different manners as using maxvoting, averaging, weighted averaging or other advanced techniques as stacking, blending, bagging or boosting.

In this work, we prove the efficiency of a particular ensemble machine learning that combines the feature spaces of several deep machines to solve the HGR problem. In particular, the ensemble is formed from the feature spaces of a standard Convolutional Neural Net (CNN) and a Capsule Network (CapsNet). Diversity is also introduced in the data set by means of a data augmentation procedure. This regularization technique has turned out to be critical to obtain the best possible result on the MNIST data set of American Sign Language (ASL).

5.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) are one of the engines that have driven the breakthrough of DL in recent years, being the preferred option for many computer vision tasks [200, 201, 202]. Inspired by the organization of the Visual Cortex, CNN are composed of individual neurons that respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover

the entire visual area.

CNN is a sequence of three different kinds of layers: convolutional layer (with non lineal ReLu activation), pooling layer and fully-connected layer. The convolutional and pooling layers are used to be applied several times in order to reduce the input image into a smaller one which is easier to process and contain features which are critical for getting a good prediction. The resulting image can be flattened in a vector form that is finally classified by a set fully-connected layers. The most relevant advantage of a CNN is the automatic feature extraction for the given task. Besides this, CNN have others important advantages mainly by using convolution and pooling operations. In particular, pooling

- makes the input representations smaller and more manageable;
- reduces the number of parameters and computations in the network, therefore, controlling overfitting;
- increases the field of view of higher layers thus allowing to obtain more general features of the input image.
- makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in input will not change the output of pooling – since we take the maximum/average value in a local neighborhood).

Meanwhile convolution

- helps us to obtain a method for the recognition of objects that is almost invariant under translation, what suppose a very powerful feature since we can detect objects in an image no matter where they are located.

Nevertheless, CNN present a several limitations:

- CNN lack of ability to be invariant to large transformations, distortions and translations in the input image. Although CNN solves this problem slightly by using max pooling and convolution, these are simply a bad approach to the solution;
- max pooling can cause loss of valuable information;
- the internal representation of a CNN does not take into account the spatial relationships between objects, nor the existing hierarchy between simple objects and the composite objects of which they are a part.

It can be concluded that the internal representation of a CNN does not take into account the spatial relationships between objects, nor the existing hierarchy between simple objects and the composite objects of which they are a part.

5.2.2 Capsule Networks

There are many research proposals to address the limitations of CNN, some example are [203, 204, 205, 206]. Among them, the so-called Capsule Networks (*CapsNets*) stand out. They are a novel structure of CNN which simulates the visual processing system of human brain and were proposed by G. E. Hinton to solve the intrinsic problems of the CNNs [207, 208]. In particular, to achieve CNNs with internal data representation that take into account spatial hierarchies between simple and complex objects.

This network is composed of capsules. A capsule, like a classical neuron, is a non-linear computation node of a learning machine. Nevertheless, there is a remarkable difference between them: in a capsule, both the input components and the output are vectors instead of scalars, as it happens in a real neuron [209].

The length of the output vector denotes the probability that the entity (an object, for example) exists and the state of the detected entity is encoded in the direction of the vector. The most important characteristic of the CapsNets is that their output components, called 'instantiation parameters', are equivariant. This means that they allow maintaining the information of spatial relationships between the object components, what makes the network invariant to the viewpoint. Thus, a relevant consequence of this property emerges: a CapsNets can identify new, unseen variations of the class without ever being trained on them [209].

Let's take a look at capsules and how they go about solving the problem of providing spatial information.When we look at some of the logic that's behind CNN's, we begin to notice where its architecture fails. Take a look at this picture [209].
It doesn't look quite right for a face, even though it has all the necessary components to make up a face. We know that this is not how faces are supposed to look, but because CNN's only look for features in images, and don't pay attention to their pose, it's hard for them to notice a difference between that face and a real face [209].
How capsule networks solve this problem is by implementing groups of neurons that encode spatial information as well as the probability of an object being present. The length of a capsule vector is the probability of the feature existing in the image and the direction of the vector would represent its pose information [209].

In computer graphics applications such as design and rendering, objects are often created by giving some sort of parameter which it will render from. However, in capsules networks, it's the opposite, where the network learns how to inversely render an image; looking at an image and trying to predict what the instantiation parameters for

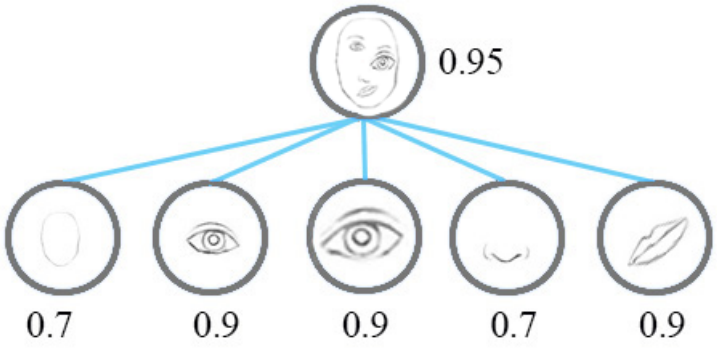

Fig. 5.5 How a CNN would classify this image.

it are [209].

It learns how to predict this by trying to reproduce the object it thinks it detected and comparing it to the labelled example from the training data. By doing this it gets better and better at predicting the instantiation parameters. The Dynamic Routing Between Capsules paper by Geoffrey Hinton proposed the use of two loss functions as opposed to just one [209].

The main idea behind this is to create equivariance between capsules. This means moving a feature around in an image will also change its vector representation in the capsules, but not the probability of it existing. After lower level capsules detect features, this information is sent up towards higher level capsules that have a good fit with it [209].

As seen in this picture, all of the pose parameters of the features are used to determine the final result.

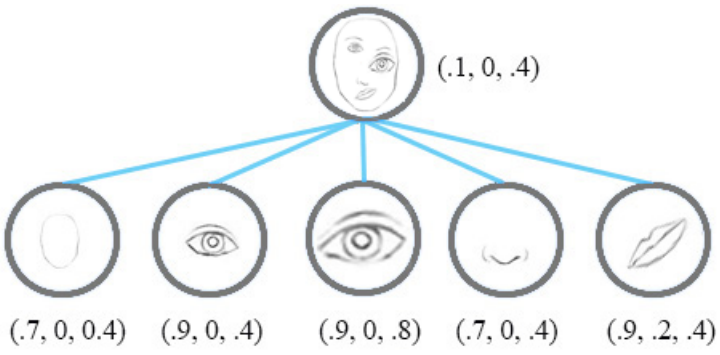

Fig. 5.6 How a Capsule Network would classify this face.

Equivariance is due, in part, to a new training algorithm called *routing by agreement* which ensures that the output of a capsule gets sent to an appropriate parent in the layer above. In particular, each output of the lower level capsules is multiplied by a coeffi-

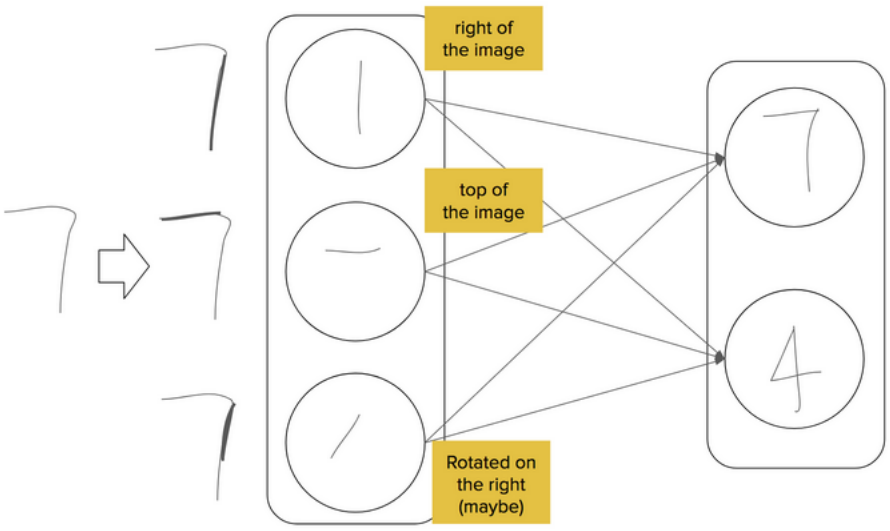

Fig. 5.7 Process of Dynamic Routing.

cient and sent to every capsule in upper level layer. The routing algorithm increases or decreases these coupling coefficients according to the agreement between the output of the lower capsule and those of the layer above.

5.2.2.1 Architecture of Capsule Network on MNIST

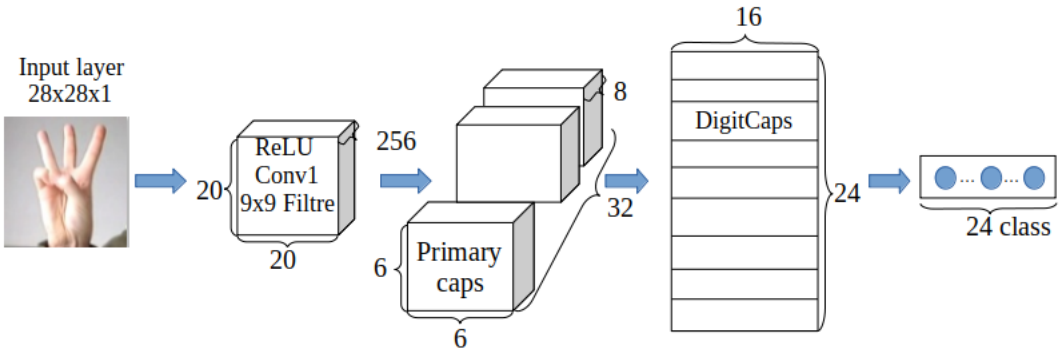

Fig. 5.8 CapsNet Architecture.

The Encoder takes the image input and learns how to represent it as a 16-dimensional vector which contains all the information needed to essentially render the image [209].

- Conv Layer: Detects features that are later analyzed by the capsules. As proposed in the paper, contains 256 kernels of size 9x9x1.

- Primary(Lower) Capsule Layer: This layer is the lower level capsule layer which I described previously. It contains 32 different capsules and each capsule applies eighth 9x9x256 convolutional kernels to the output of the previous convolutional layer and produces a 4D vector output.
- Digit(Higher) Capsule Layer: This layer is the higher level capsule layer which the Primary Capsules would route to(using dynamic routing). This layer outputs 16D vectors that contain all the instantiation parameters required for rebuilding the object.

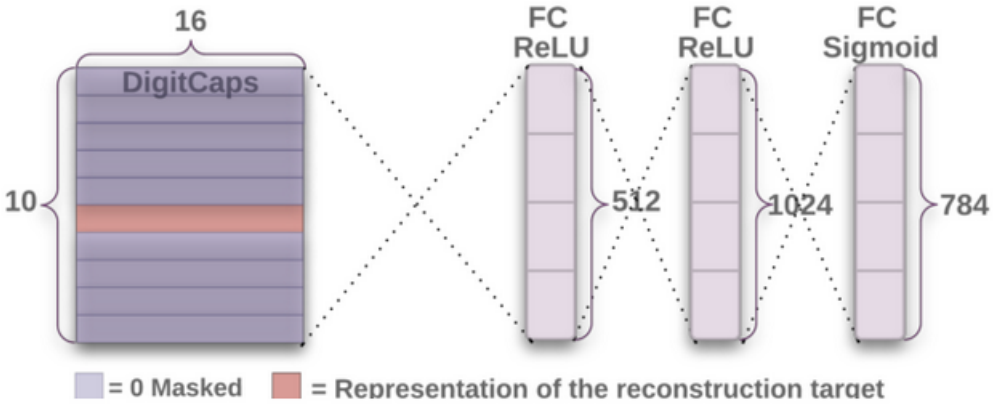

Fig. 5.9 Decoder Architecture.

The decoder takes the 16D vector from the Digit Capsule and learns how to decode the instantiation parameters given into an image of the object it is detecting. The decoder is used with a Euclidean distance loss function to determine how similar the reconstructed feature is compared to the actual feature that it is being trained from. This makes sure that the Capsules only keep information that will benefit in recognizing digits inside its vectors. The decoder is a really simple feed-forward neural net that is described below [209].

- Fully Connected (Dense) Layer 1.
- Fully Connected (Dense) Layer 2.
- Fully Connected (Dense) Layer 3.
- Final Output with 24 classes.

The capsule networks have aroused great interest in the scientific community. Different works have emerged recently proposing improvements to the original CapsNets. For example, a modified routing algorithm, called Cognitive Consistency Routing Algorithm, is presented in [210]. The main motivation of this work is to ensure each capsule

layer makes the prediction of the target as consistent as possible. In [211], however, Path Capsule Networks (PathCapsNets) are presented as an effort to explain behavior and decisions of deep and capsule networks. They are proved to achieve better or comparable performance to CapsNet with significant parameter savings.

5.3 Reduce a Dataset Dimensionality

Feature Extraction aims to reduce the number of features in a dataset by creating new features from the existing ones (and then discarding the original features). These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarized version of the original features can be created from a combination of the original set.

Another commonly used technique to reduce the number of features in a dataset is Feature Selection. The difference between Feature Selection and Feature Extraction is that feature selection aims instead to rank the importance of the existing features in the dataset and discard less important ones (no new features are created).

5.3.0.1 Principle Components Analysis (PCA)

PCA is one of the most commonly used techniques for reducing linear dimensions and has been used in our research. When using PCA, we take our original data as input and try to find a set of input features that can better summarize the distribution of the original data to reduce its original dimensions. PCA can do this by increasing contrasts and reducing reconstruction error by looking at even distances. In PCA, our original data are displayed in a set of orthogonal axes and each axis is arranged in order of importance.
PCA is an unsupervised learning algorithm, so it is not concerned with data labels but only diversity. In some cases, this can lead to misclassification of data. The eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data. Before getting to the explanation of these concepts, let's first understand what we mean by principal components.
PrinPrincipal components are new variables that are constructed as linear combinations ormixtures of the initial variables. These combinations are done in such a way that the new variables (i.e. principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on [212], until having something like shown in the scree

plot below. Organizing information in principal components this way will allow you

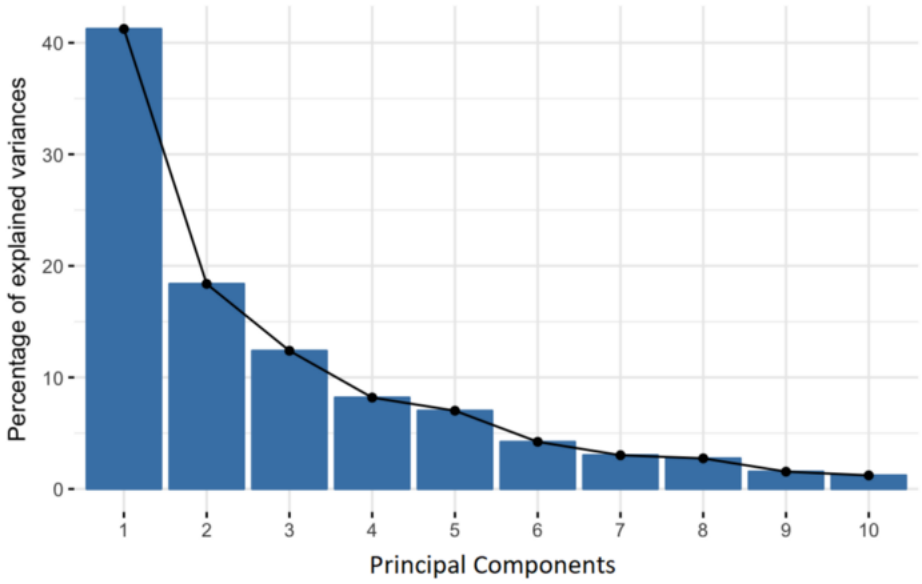

Fig. 5.10 Percentage of Variance (Information) for each by principal components.

to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

An important thing to realize here is that, the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables. Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. The relationship between variance and information here is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible [212].

5.4 Classification Using Support Vector Machine (SVM)

The Support Vector Machine is a supervised learning algorithm mostly used for classification but it can also be used for regression. The main idea is that based on the labelled data (training data) the algorithm tries to find the optimal hyperplane which can be used to classify new data points. In two dimensions the hyperplane is a simple line Figure 5.11 [213].

Usually a learning algorithm tries to learn the most common characteristics (what differentiates one class from another) of a class and the classification is based on those representative characteristics learnt (so classification is based on differences between classes). The SVM works in the other way around. It finds the most similar examples between classes. Those will be the support vectors [214].

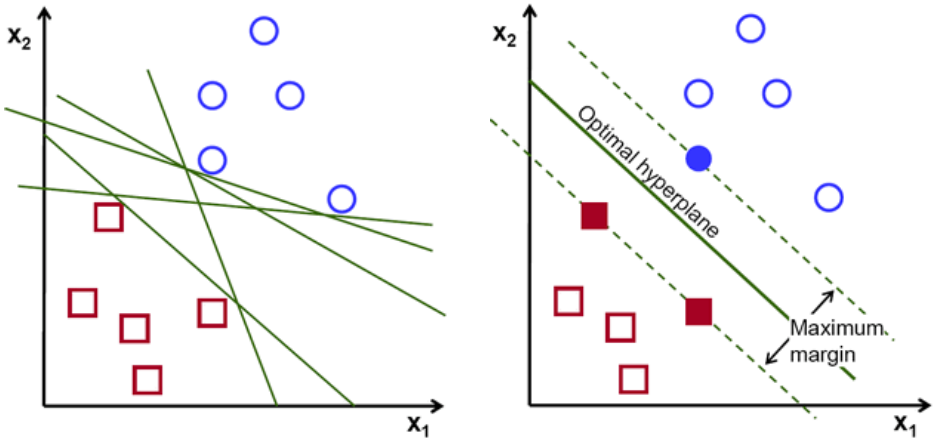

Fig. 5.11 Possible hyperplanes.

5.5 Proposed Method

As previously described in the section, diversity is a very important concept in machine learning because, it produces a greater capacity of generalization acting as a regularization element of learning which translates into greater accuracy. Diversity of the learned model (diversity in parameters of each model or diversity among different base models) makes each parameter/model capture unique or complement information. Here, we proposed an ensemble method that combines the feature space of a CNN network with the feature space of CapsNets in order to improve the accuracy of HGR problems. It is expected that the capabilities of the well-known CNN networks for image pattern recognition problems will be strengthened by those from the CapsNet architecture.
CNNs and CapsNet are considered in this way as feature extractors mechanisms that increase the diversity by means of the conjunction of their feature spaces. Once the feature spaces are combined, a new expanded feature space is obtained, where the classification problem has to be solved. Figure 5.12 shows the whole network and the new feature space Z.

As can be observed in Figure 5.12, vectors in the expanded feature space conform the input to be classified by a dense neural network, in this case, a Support Vector Machine (SVM). Notice that, previously to this classification task, a Principal Component Analysis (PCA) procedure is applied in order to reduce both redundancy and data di-

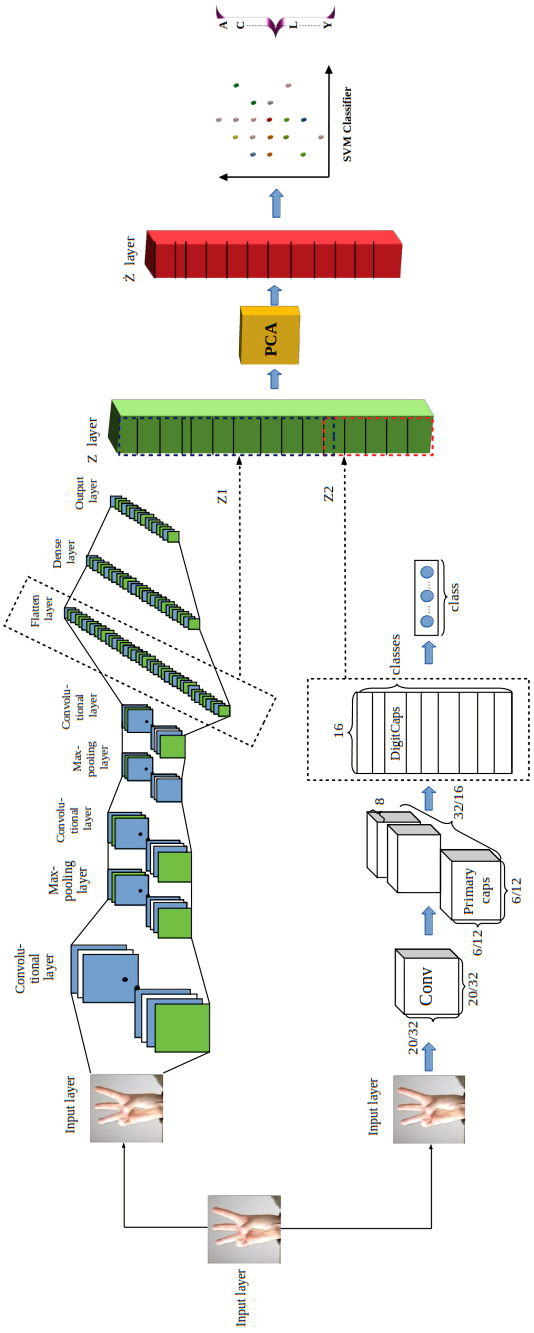

Fig. 5.12 Proposed ensemble network.

mensionality.
In this work, the proper design of that CNN is based on classical LeNet-5 architecture [65], but making some updates following the current practice: ReLU activation on convolutional layers, max pooling layers instead of average pooling, and one dropout layer. One of the major benefit of the ReLU activation function is the reduced likelihood of the gradient to vanish in contrast with the sigmoid functions. The other benefit is sparsity because of it produces more values close or equal to zero which translate to a sparse representation that are more beneficial than a dense one. Average pooling was often used historically but has recently fallen out of favor compared to the max pooling operation, which has been shown to work better in practice.
The dropout method is use because it is prove to be one of the better choice to reduce the overfitting [215].LeNet-5 was initially proposed for recognition of handwritten digits on MNIST database [216], a quite similar problem to the one in this work.
Finally, diversity of the training data is also considered in the proposed method. Diversity in the data ensures that the training data can provide more discriminative information for the model. Now, diversity is introduced by means of a particular data augmentation method due to its high effectiveness in improving results.
More specifically, the initial size of the training dataset was enlarged using standard data augmentation techniques: operations of rotation (up to 15 degrees) and translation (up to 2 pixels) was applied to the original images, with "nearest neighbor" padding for pixels that come from outside the boundaries.

Each of the components of the proposed method is described below.

5.5.1 Baseline CNN

As initial starting point, a standard CNN is evaluated and its performance is considered as baseline for comparative purposes. As it was previously commented, the design of this CNN was inspired on LeNet-5, since some kind of variation of such network is a common election in problems with similar training datasets to the one considered here [208, 217].
As can be seen in Table 5.2, the selected baseline CNN consists of two sets of convolutional and max-pooling layers, one additional convolutional and flattening layer, one fully-connected classifier and finally a softmax output layer of 24 or 26 dimension depending on the dataset. The input layer accepts single channel grayscale images with dimensions of 28×28 pixels for MNIST ASL dataset and 40×40 pixels for other ASL datasets. From input to output, the three convolutional layers have 16, 32 y 32 filters with kernel size of 5×5 and stride of one. The size of the fully-connected layer is 64 and that of the output layer 24 or 26 depending on the dataset. These hyperparameters

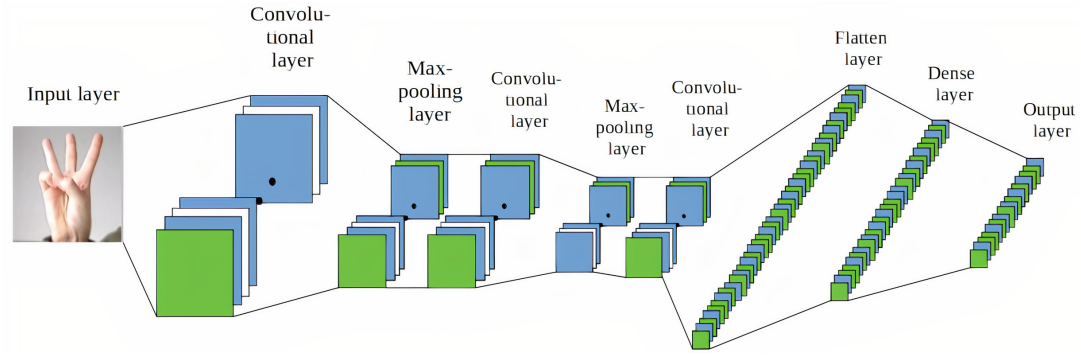

Fig. 5.13 Proposed CNN Architecture.

was tuned to achieve the best performance on all datasets, while keeping the computational cost close to CapsNet one. The CNN was trained using standard cross entropy loss function.

Table 5.2 Proposed CNN architectures.

Depth	**CNN for MNIST ASL Dataset**	**Output shape**	**CNN for the rest of datasets**	**Output shape**
Input	Input (28×28)		Input (40×40)	
1	Convolutional (5×5)	28×28×16	Convolutional (5×5)	40×40×16
2	Max pooling (2×2)	14×14×16	Max pooling (2×2)	20×20×16
3	Convolutional (5×5)	14×14×32	Convolutional (5×5)	20×20×32
4	Max pooling (2×2)	7×7×32	Max pooling (2×2)	10×10×32
5	Convolutional (5×5)	7×7×32	Convolutional (5×5)	10×10×32
6	Flatten	1,568	Flatten	3,200
7	Dense	64	Dense net	64
Output	Softmax	24	Softmax	24 / 26

5.5.2 CapsNet

Presumably due to the similarity of problems, and after repeated attempts, the CapsNet hyperparameters (Figure 5.14) remain unchanged from those originally proposed the authors [208] in order to reach the optimal performance.

As can be seen in Table 5.3, the first convolutional layer uses 256 filters of 9×9×1 to generate a new representation of the input image. Depending on the dataset, the images have 28×28 or 40×40 pixels. Focusing, for example, on the MNIST dataset, the new representation is of 20×20×256 dimension. The second layer have 32 channels (16 for the rest of datasets), each one with 6×6 capsules (PrimaryCaps) of 8×1 dimension (see Figure 5.14). The last layer is the DigitCaps layer, which contains as many capsules as the number of classes (24), one for each sign in the alphabet. A digitcap is a 1×16 vector that is obtained multiplying the output vector of a primarycap by a 8×16 trainable weight matrix.

Table 5.3 Proposed Capsule Network Encoder architectures.

Depth	CapsNet for MNIST ASL Dataset	Output shape	CapsNet for the rest of datasets	Output shape
Input	Input (28×28)		Input (40×40)	
1	Convolutional (9×9)	20×20×256	Convolutional (9×9)	32×32×256
2	Convolutional (9×9×256)	6×6×256	Convolutional (9×9×256)	12×12×256
3	PrimaryCaps	1,152×8	PrimaryCaps	2,304×8
Output	DigiCaps	24×16	DigitCaps	24/26×16

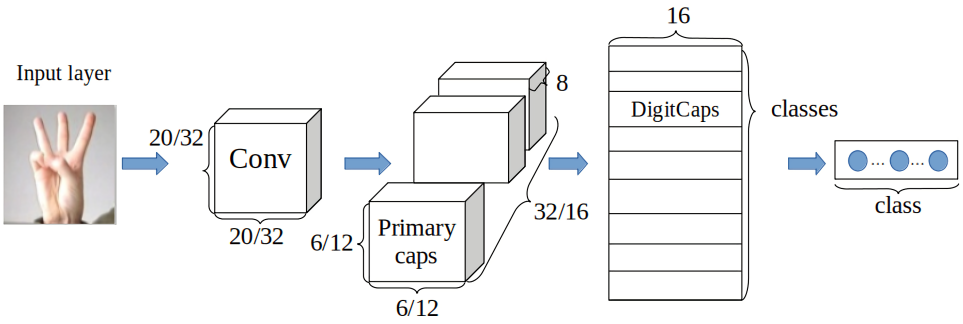

Fig. 5.14 Architecture of Encoder CapsNet.

5.5.3 Ensemble model

As earlier pointed out, the proposed model consist of an ensemble of the two previously described networks (CNN and CapsNets), working as feature extractors and finally combined to improve the classification accuracy in the HGR problem.

In the proposed model (see Figure 5.12), the expanded feature space Z is composed by the conjuntion of $Z1$ and $Z2$, the feature spaces of the CNN and CapsNet respectively. In this work, it will be considered the following feature vectors: $Z1_{3200\times1}$ / $Z2_{384\times1}$, $Z1_{1568\times1}$ / $Z2_{400\times1}$, $Z1_{3200\times1}$ / $Z2_{416\times1}$ and $Z1_{3200\times1}$ / $Z2_{416\times1}$ for the MNIST ASL, Static Hand Gesture ASL, Massey University and Kaggle ASL Alphabet datasets, respectively. The obtained vectors Z ($Z_{3584\times1}$ / $Z_{1968\times1}$ / $Z_{3616\times1}$ / $Z_{3616\times1}$) will be finally classified by a Support Vector Machine (SVM) classifer.

In order to reduce the dimensionality of Z, a subsequent PCA procedure is applied and the first most relevant features are preserved. For the datasets considered here, the best performances have been obtained with the 1,200 / 800 / 1,500 / 1,500 most relevant features, i. e., with reduced feature vector $\hat{Z}_{1200\times1}$ / $\hat{Z}_{800\times1}$ / $\hat{Z}_{1500\times1}$ / $\hat{Z}_{1500\times1}$.

5.6 Comparative performance

In order to obtain robust statistical conclusions, the testing performance values were computed by averaging the results of a model after 20 training runs. Moreover, the best

Table 5.4 Average and standard deviation accuracy for considered methods (CNN, CapsNet, Proposed model) and different datasets. DA denotes Data Augmentation.

% Test Accuracy (Mean ± std)	CNN	CapsNet	Ensemble	CNN DA	CapsNet DA	Ensemble DA
Massey University dataset	91.48 ± 0.03	86.03 ± 0.19	91.21 ± 0.26	98.05 ± 0.01	98.52 ± 0.01	**99.18 ± 0.39**
Static Hand Gesture ASL dataset	73.26 ± 0.25	67.70 ± 0.33	74.65 ± 0.31	98.26 ± 0.26	95.71 ± 0.48	**98.96 ± 0.22**
Kaggle ASL Alphabet dataset	88.20 ± 0.35	92.13 ± 0.13	96.67 ± 0.38	91.94 ± 0.55	93.08 ± 0.21	**99.13 ± 0.17**
MNIST ASL dataset	96.32 ± 0.41	88.72 ± 0.28	94.42 ± 0.05	98.11 ± 0.55	99.08 ± 0.38	**99.69 ± 0.17**

average results are indicated in bold for each dataset. A method is considered better than another when the difference between the means of the corresponding MSEs is at least greater than the average of their standard deviations. This criterion has been applied in the results showed in Tables 5.4 and 5.5.

Table 5.4 collects obtained classification accuracy for every dataset. The results are showed in terms of mean and standard deviation accuracy provided by the different models.

As can be observed, CapsNet outperforms the CNN model when data augmentation is applied, obtaining a poor result when it is not considered. This can be explained by the presence of two major defects of CNN: their failure to look at the important spatial hierarchy between features, and their lack of rotational stability.

Table 5.5 shows the obtained accuracy by proposed method compared with the results achieved in other works using similar and recent approaches, i.e. they make use of some variant of convolutional networks and apply it over some of the same datasets tested here. Taskiran et al. [218] achieved an accuracy of 98.05% with standard CNNs over Massey University dataset. Rastgoo et al. [219] proposed a deep-based model using Restricted Boltzmann Machine attaining an accuracy of 99.31% over the same dataset, and Verma et al. [220] capture effective features by employing attention mechanism and dilated convolution provides global features reaching an accuracy of 98.80% for the best situation. Pinto et al. [197] recently achieved respectively 99.40% and 98.24% of classification accuracy over Massey University and Static Hand Gesture ASL datasets. Over MNIST ASL Zhao et al. [221] used a basic optimized ConvNet to reach a classification accuracy of 89.32%, and Bilgin et al. [217] introduced the CapsNets to the recognition of ASL hand gestures and compared their performance with classic LeNet CNNs. They accomplished a 95.08% of accuracy in the best case. In general terms, it can be observed on Table 5.5 that the proposed ensemble provides state-of-the-art results, improving the classification accuracy of hand gestures achieved by the referenced works. Note that these works provide only average accuracy values, and the standard deviation of their experiments were not available for comparison. It should be pointed out that in the case of noisy dataset Kaggle ASL Alphabet the performance has not deteriorated, achieving a remarkable accuracy greater than 99%.

Table 5.5 Comparison with methods based on CNNs from other authors using the same datasets.

Reference method	**% Test Accuracy** Massey University	**% Test Accuracy** Static Hand Gesture ASL	**% Test Accuracy** Kaggle ASL Alphabet	**% Test Accuracy** MNIST ASL
[219] Rastgoo et al. 2018	**99.31**	-	-	-
[218] Taskiran et al. 201	98.05	-	-	-
[221] Zhao and Wang 2018	-	-	-	89.32
[197] Pinto et al. 2019	**99.40**	98.24	-	-
[217] Bilgin and Mutludo˘gan	-	-	-	95.08
[220] Verma et al. 2021	98.80	-	-	-
Proposed ensemble	**99.18** $\pm$ **0.39**	**98.96** $\pm$ **0.22**	**99.13** $\pm$ **0.17**	**99.69** $\pm$ **0.17**

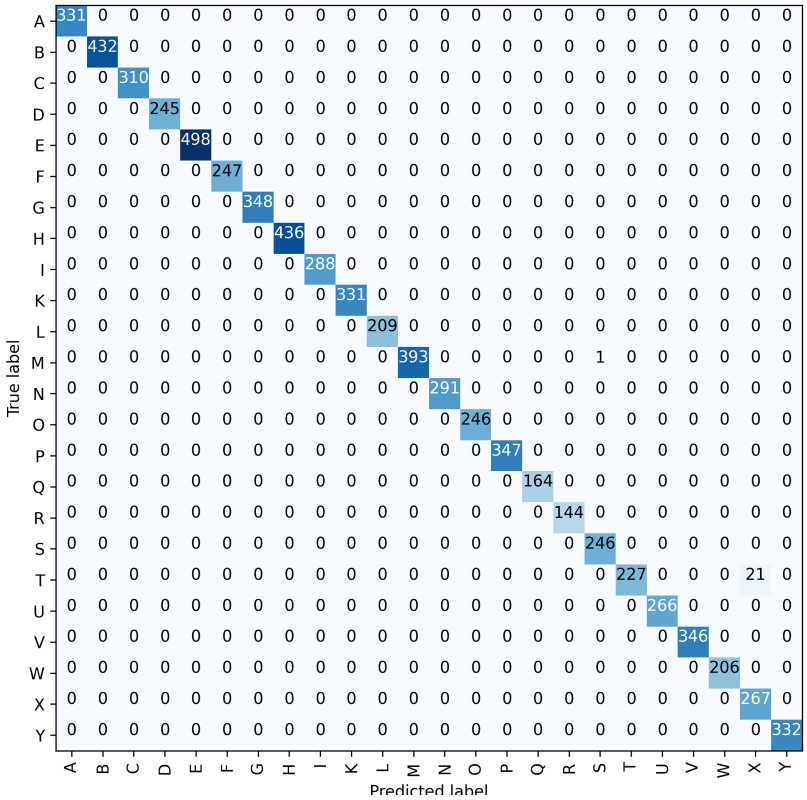

Fig. 5.15 MNIST ASL dataset.

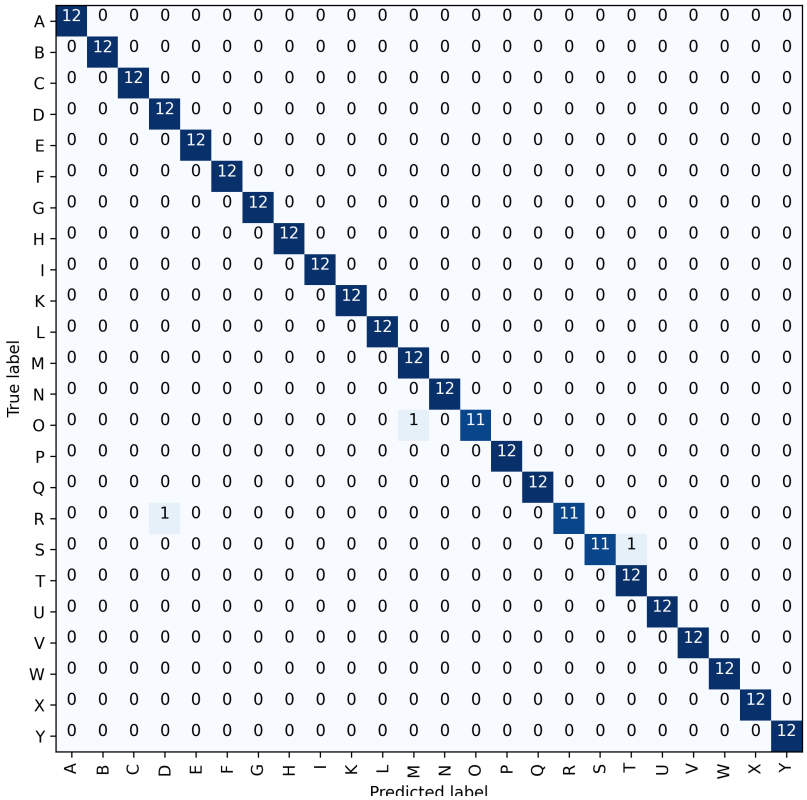

Fig. 5.16 Static Hand Gesture ASL dataset.

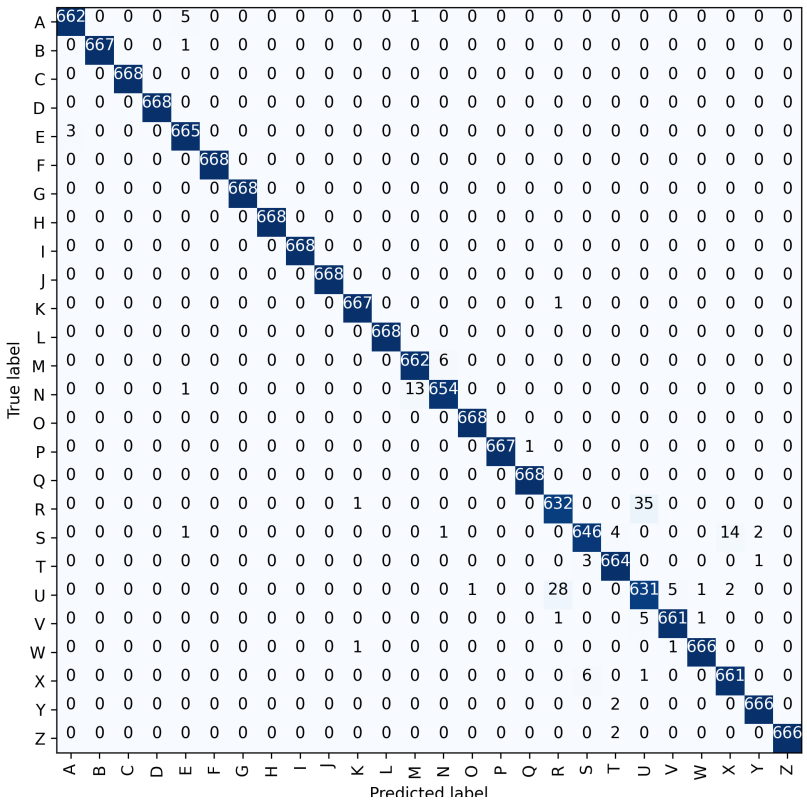

Fig. 5.17 Kaggle ASL Alphabet dataset.

Table 5.6 Precision and recall values of proposed ensemble model for MNIST ASL and Static Hand Gestures datasets using data augmentation.

		MNIST ASL dataset			**Static Hand Gesture ASL dataset**	
Sign	Samples	Precision	Recall	Samples	Precision	Recall
A	331	0.94	1.00	12	1.00	1.00
B	432	1.00	1.00	12	1.00	1.00
C	310	1.00	1.00	12	1.00	1.00
D	245	1.00	1.00	12	0.92	1.00
E	498	1.00	1.00	12	1.00	1.00
F	247	1.00	1.00	12	1.00	1.00
G	348	1.00	1.00	12	1.00	1.00
H	436	1.00	1.00	12	1.00	1.00
I	288	1.00	1.00	12	1.00	1.00
K	331	1.00	1.00	12	1.00	1.00
L	209	1.00	1.00	12	1.00	1.00
M	394	1.00	1.00	12	0.92	1.00
N	291	1.00	1.00	12	1.00	1.00
O	246	1.00	1.00	12	1.00	0.92
P	347	1.00	1.00	12	1.00	1.00
Q	164	1.00	1.00	12	1.00	1.00
R	144	1.00	1.00	12	1.00	0.92
S	246	1.00	1.00	12	1.00	0.92
T	248	1.00	0.92	12	0.92	1.00
U	266	1.00	1.00	12	1.00	1.00
V	346	1.00	1.00	12	1.00	1.00
W	206	1.00	1.00	12	1.00	1.00
X	267	0.93	1.00	12	1.00	1.00
Y	332	1.00	1.00	12	1.00	1.00

Table 5.7 Precision and recall values of proposed ensemble model for Kaggle ASL Alphabet and Massey University datasets using data augmentation.

		Kaggle ASL Alphabet Dataset			**Massey University Dataset**	
Sign	Samples	Precision	Recall	Samples	Precision	Recall
A	668	1.00	1.00	14	1.00	1.00
B	668	1.00	1.00	14	1.00	1.00
C	668	1.00	1.00	14	1.00	1.00
D	668	1.00	1.00	14	1.00	1.00
E	668	1.00	1.00	14	1.00	1.00
F	668	1.00	1.00	14	1.00	1.00
G	668	1.00	1.00	14	1.00	1.00
H	668	1.00	1.00	14	1.00	1.00
I	668	1.00	1.00	14	1.00	1.00
J	668	1.00	1.00	14	1.00	1.00
K	668	1.00	1.00	14	1.00	1.00
L	668	1.00	1.00	14	1.00	1.00
M	668	0.98	0.99	14	1.00	1.00
N	668	0.99	0.99	14	1.00	0.79
O	668	1.00	1.00	14	1.00	1.00
P	668	1.00	1.00	14	1.00	1.00
Q	668	1.00	1.00	14	1.00	1.00
R	668	0.95	0.94	14	1.00	1.00
S	668	0.98	0.97	14	1.00	1.00
T	668	0.98	0.99	14	0.82	1.00
U	668	0.93	0.95	14	1.00	1.00
V	668	0.99	0.99	14	1.00	1.00
W	668	1.00	0.99	14	1.00	1.00
X	668	0.98	0.98	14	1.00	1.00
Y	668	1.00	1.00	14	1.00	1.00
Z	668	1.00	1.00	14	1.00	1.00

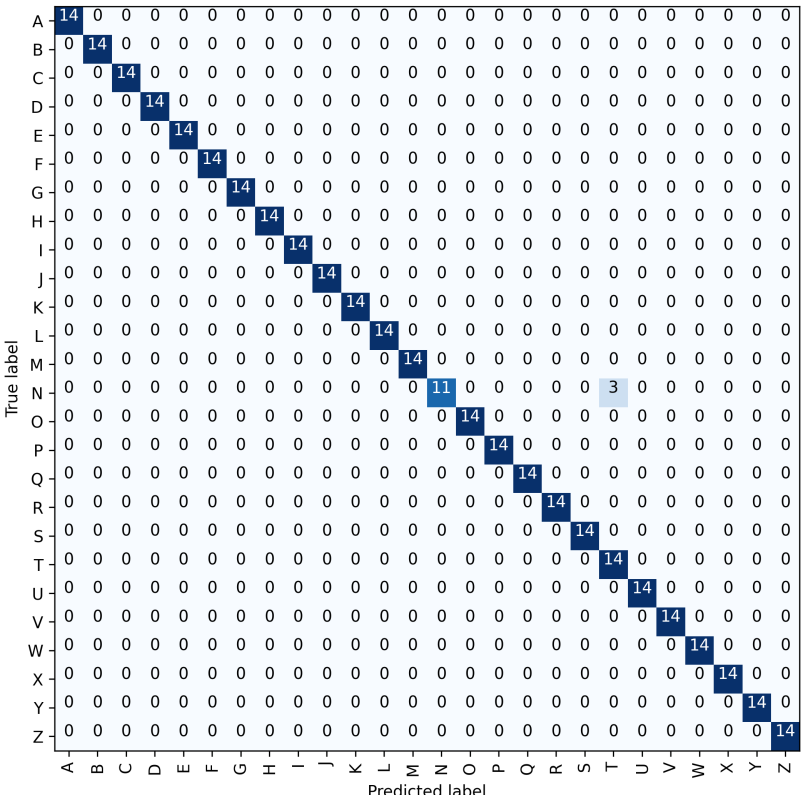

Fig. 5.18 Massey University dataset.

Finally, to better analyze the proposed method, the metrics of Precision, Recall and Confusion Matrices are presented in Tables 5.6 and 5.7 and Figure 5.15, 5.16, 5.17 and 5.18 respectively.

Each column of the matrix represents the instances in a predicted gesture, instead each row represents the instances in a current gesture. The main diagonal of the matrix represents the instances correctly classified by the proposed model. The gestures incorrectly classified as not belonging to a class of interest, while the off-diagonal values show the mistakes made. As can be observed, the proposed method does not suffer of ambiguity issues. The misclassifications are concentrated in a few letters, and these letters differ from one data set to another: 3 for entry N predicted as S in the Massey University dataset and this is confirmed by the ratio of the letter N in Recall (0.79) (see Table 5.7). 1 for entries O, R, S predicted as M, D, T respectively, in the Static Hand Gesture dataset which are confirmed by the ratio of this letters in Recall (see Table 5.6). In the ASL Kaggle Alphabet data set, there are a larger amount of errors. The most relevant are 35, 28, 14 for the letters R, U, S predicted as U, R, X. This misclassifications are obvious and identical to the proportions of each letter in the corresponding recall values in Table 5.7. Finally, in the MNIST ASL dataset, there are 21 and 1 error for input T, M respectively predicted as X, S. Again, this misclassifications are confirmed by the proportions of each letter in Recall shown in Table 5.6. This result is not surprising

due to the distinction of some gestures is very hard, since they are very similar to other gestures in the dataset as shown in Figure 5.15, 5.16, 5.17 and 5.18 respectively.

5.7 Conclusions

In this work, we address the problem of hand gestures recognition using software-based deep learning techniques. More specifically, the problem to be solved is the sign classification on the American Sign Languages (ASL). To tackle it, standard CNNs, the most used networks in image recognition problems, and capsule networks (CapsNets) have been used. Both networks can be considered complementary because CapsNets were designed to improve the performance of the CNN. In particular, they satisfy the equivariance property, which means they allow maintaining the information of spatial relationships between the object components.

Looking for the best possible solution, a new model is proposed that combines the networks by joining the feature spaces of both. This ensemble model reduces the dimension of the resulting feature space by means of a PCA, and classifies that representation of the input data by an SVM classifier. Experiments show that with this idea and a proper and simple data augmentation process, it is possible to achieve an excellent performance with most errors concentrated in some particular and difficult hand gestures. This fact suggests the possibility of increasing the accuracy by means of voting, bagging or boosting schemes. The method is based on increasing diversity in both the model by means of the ensemble mechanism and the dataset by means of data augmentation process.

Conclusion

In this thesis, we have proposed a method for hand gestures recognition, which in the operational environment will have a direct impact on the performance of human operators through a qualified and acceptable HCI interface, will push the limits of HCI, especially in the event of infectious epidemics such as the outbreak of the coronavirus COVID 19.
The aim of this research is to propose an approach applying artificial vision methods to American sign language (ASL) images by comparing our work with different computer vision strategies proposed in recent years. A list of the tasks we performed is shown below:

- Diversity: Here, we proposed an ensemble method that combines the feature space of a CNN network with the feature space of CapsNets in order to improve the accuracy of HGR problems.
- We have suggested the use of capsule networks to avoid the shortcomings of convolutional neural networks, because it can be seen that the internal representation of a CNN does not take into account the spatial relationships between objects, nor the existing hierarchy between simple objects and the composite objects of which they are a part.
- The proposal shows four essential experiments with four different datasets in the field of hand gesture.
- Reducing the dimensionality of the final vector of features by using the Principal Component Analysis (PCA)
- Classification of objects, mimicking the process of human thought, which allows complex judgments to be made with precision, speed and consistency.

Improving of a hand gesture recognition without going through segmentation using deep learning approaches. We have proposed a new approach using a collection of two CNNs.
First, for the improvement of capacity of generalization, the results show that our proposed method is the best technique compared to the other literature methods. Secondly, Massey University datasets, Static Hand Gesture ASL datasets, Kaggle ASL

Alphabet datasets, MNIST ASL datasets, with CapsNet or CNN model presents a good performance by a test accuracy equal to 98.52%, 98.26%, 93.08% and 99.08%, respectively. Furthermore, our proposed model provided better results than others tested algorithms showing the efficiency of the system. In particular, the system achieved 99.18%, 98.96%, 99.13% and 99.69%, for the datasets mentioned earlier in succession. On the other hand, for object detection by machine learning, we noted that the results obtained are very satisfactory with a very small number of false detections.

Although the results obtained in this thesis are more than satisfactory and promising, nevertheless, there are some limitations to this research. First, it is difficult to obtain complete information on existing commercial systems, or even on the data obtained from these systems, for reasons of ownership (whether the systems are certified or not). Also, the data used for our research its public data, the limitation is that all the data collected was limited to the small set of hand gestures.

As pointed out in the introduction section, a complete sign language recognition system requires the integration of different and usually separable modules. The authors intend to develop a DL-based segmentation algorithm that can be integrated with current work, so that the hand gesture images that feed the classification module can be located and extracted directly from image frames captured by a simple image device such as a webcam. Pretrained networks or transfer learning also need to be experimented for comparison or optimization purposes.

LIST OF PUBLICATIONS

Articles

1. Bousbai, K., and J Morales-Sánchez and Merah, M. and JL Sancho-Gómez (2022). Improving Hand Gestures Recognition Capabilities by Ensembling Convolutional Networks. Journal of Expert Systems.DOI: 10.1111/exsy.12937

International Conferences

1. Bousbai, K., and Merah, M. (2019, November). A Comparative Study of Hand Gestures Recognition Based on MobileNetV2 and ConvNet Models. In 2019 6th International Conference on Image and Signal Processing and their Applications . ISPA) (pp. 1-6). IEEE)

2. Bousbai, K., & Merah, M. (2022, May). Hand Gesture Recognition Using Capabilities of Capsule Network and Data Augmentation. In 2022 7th International Conference on Image and Signal Processing and their Applications (ISPA) (pp. 1-5). IEEE

REFERENCES

[1] Fan Zhang, Yue Liu, Chunyu Zou, and Yongtian Wang. Hand gesture recognition based on hog-lbp feature. In *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE, 2018.

[2] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5 – 20, 2005. Diversity in Multiple Classifier Systems.

[3] Nils Nilsson. *Learning machines; foundations of trainable pattern-classifying systems*. McGraw Hill, New York, 1965.

[4] João Moreira, Carlos Soares, Alípio Jorge, and Jorge Sousa. Ensemble approaches for regression: A survey. *ACM Computing Surveys*, 45:10:1–10:40, 11 2012.

[5] Yun Yang. Chapter 4 - ensemble learning. In Yun Yang, editor, *Temporal Data Mining Via Unsupervised Ensemble Learning*, pages 35 – 56. Elsevier, 2017.

[6] Leap motion. `https://www.leapmotion.com/`, view in 2020.

[7] Leigh Ellen Potter, Jake Araullo, and Lewis Carter. The leap motion controller: a view on sign language. In *Proceedings of the 25th Australian computer-human interaction conference: augmentation, application, innovation, collaboration*, pages 175–178, 2013.

[8] Shortcuts. `https://apps.leapmotion.com/apps/shortcuts/windows`, view in 2020.

[9] Ultraleap hand tracking cameras and software demonstrated at work. `https://apps.leapmotion.com/apps/cyber-science-motion/windows`, view in 2020.

[10] Francesco Camastra and Domenico De Felice. Lvq-based hand gesture recognition using a data glove. In *Neural Nets and Surroundings*, pages 159–168. Springer, 2013.

[11] Mubashira Zaman, Soweba Rahman, Tooba Rafique, Filza Ali, and Muhammad Usman Akram. Hand gesture recognition using color markers. In *International Conference on Hybrid Intelligent Systems*, pages 1–10. Springer, 2016.

[12] Time of flight. https://en.wikipedia.org/wiki/Time_of_flight, view in 2020.

[13] wikipedia. Structured light. https://en.wikipedia.org/wiki/Structured_light, view in 2020.

[14] Time of flight. https://en.wikipedia.org/wiki/Stereo_camera, view in 2020.

[15] Iker Vazquez Lopez. Hand gesture recognition for sign language transcription. 2017.

[16] Javier Molina, Marcos Escudero-Viñolo, Alessandro Signoriello, Montse Pardàs, Christian Ferrán, Jesús Bescós, Ferran Marqués, and José M Martínez. Real-time user independent hand gesture recognition from time-of-flight camera video using static and dynamic models. *Machine vision and applications*, 24(1):187–204, 2013.

[17] Dominique Uebersax, Juergen Gall, Michael Van den Bergh, and Luc Van Gool. Real-time sign language letter and word recognition from depth data. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*, pages 383–390. IEEE, 2011.

[18] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Real time hand pose estimation using depth sensors. In *Consumer depth cameras for computer vision*, pages 119–137. Springer, 2013.

[19] Alexey Kurakin, Zhengyou Zhang, and Zicheng Liu. A real time system for dynamic hand gesture recognition with a depth sensor. In *2012 Proceedings of the 20th European signal processing conference (EUSIPCO)*, pages 1975–1979. IEEE, 2012.

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

[21] Duc-Hoang Vo, Trong-Nguyen Nguyen, Huu-Hung Huynh, and Jean Meunier. Recognizing vietnamese sign language based on rank matrix and alphabetic

rules. In *2015 International Conference on Advanced Technologies for Communications (ATC)*, pages 279–284. IEEE, 2015.

[22] Nicolas Pugeault and Richard Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *2011 IEEE International conference on computer vision workshops (ICCV workshops)*, pages 1114–1119. IEEE, 2011.

[23] Zhou Ren, Junsong Yuan, Jingjing Meng, and Zhengyou Zhang. Robust part-based hand gesture recognition using kinect sensor. *IEEE transactions on multimedia*, 15(5):1110–1120, 2013.

[24] Adam Baumberg. Reliable feature matching across widely separated views. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 774–781. IEEE, 2000.

[25] Yong Wang, Tianli Yu, Larry Shi, and Zhu Li. Using human body gestures as inputs for gaming via depth analysis. In *2008 IEEE International Conference on Multimedia and Expo*, pages 993–996. IEEE, 2008.

[26] Nebojsa Jojic, Barry Brumitt, Brian Meyers, Steve Harris, and Thomas Huang. Detection and estimation of pointing gestures in dense disparity maps. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 468–475. IEEE, 2000.

[27] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BmVC*, volume 1, page 3, 2011.

[28] Sotiris Malassiotis, Niki Aifanti, and Michael G Strintzis. A gesture recognition system using 3d data. In *Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission*, pages 190–193. IEEE, 2002.

[29] Anant Agarwal and Manish K Thakur. Sign language recognition using microsoft kinect. In *2013 Sixth International Conference on Contemporary Computing (IC3)*, pages 181–185. IEEE, 2013.

[30] William C Stokoe. Sign language structure. 1978.

[31] William C Stokoe, Dorothy C Casterline, and Carl G Croneberg. *A dictionary of American Sign Language on linguistic principles*. Linstok Press, 1976.

[32] Philippe Dreuw, Thomas Deselaers, David Rybach, Daniel Keysers, and Hermann Ney. Tracking using dynamic programming for appearance-based sign language recognition. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 293–298. IEEE, 2006.

[33] Kikuo Fujimura and Xia Liu. Sign recognition using depth image streams. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 381–386. IEEE, 2006.

[34] Kai Nickel, Edgar Scemann, and Rainer Stiefelhagen. 3d-tracking of head and hands for pointing gesture recognition in a human-robot interaction scenario. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 565–570. IEEE, 2004.

[35] Thad E Starner. Visual recognition of american sign language using hidden markov models. Technical report, Massachusetts Inst Of Tech Cambridge Dept Of Brain And Cognitive Sciences, 1995.

[36] Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, and Michael Brady. A linguistic feature vector for the visual interpretation of sign language. In *European Conference on Computer Vision*, pages 390–401. Springer, 2004.

[37] Richard Bowden12, Andrew Zisserman, Timor Kadir, and Mike Brady. Vision based interpretation of natural sign languages. 2003.

[38] Sushmita Mitra and Tinku Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(3):311–324, 2007.

[39] Rogerio Feris, Matthew Turk, Ramesh Raskar, Kar-Han Tan, and Gosuke Ohashi. Recognition of isolated fingerspelling gestures using depth edges. In *Real-Time Vision for Human-Computer Interaction*, pages 43–56. Springer, 2005.

[40] Omar Al-Jarrah and Alaa Halawani. Recognition of gestures in arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1-2):117–138, 2001.

[41] Scott K Liddell et al. *Grammar, gesture, and meaning in American Sign Language*. Cambridge University Press, 2003.

[42] Chung-Hsien Wu, Yu-Hsien Chiu, and Kung-Wei Cheng. Error-tolerant sign retrieval using visual features and maximum a posteriori estimation. *IEEE transactions on pattern analysis and machine intelligence*, 26(4):495–508, 2004.

[43] Kinect. Microsoft. kinect. `http://www.microsoft.com/en-us/kinectforwindows/`, view in 2020.

[44] MYO armband. Thalmiclabs. myo armband. https://www.thalmic.com/en/myo/, view in 2020.

[45] LeapMotion Inc. Leapmotion inc. leap motion controller. https://www.leapmotion.com/, view in 2020.

[46] accessed July-2014 web. American sign language dictionary. http://www.lifeprint.com/index.htm, view in 2020.

[47] Hee-Deok Yang and Seong-Whan Lee. Simultaneous spotting of signs and fingerspellings based on hierarchical conditional random fields and boostmap embeddings. *Pattern Recognition*, 43(8):2858–2870, 2010.

[48] Hee-Deok Yang and Seong-Whan Lee. Robust sign language recognition with hierarchical conditional random fields. In *2010 20th International Conference on Pattern Recognition*, pages 2202–2205. IEEE, 2010.

[49] EyeSight Technology Itay Katz. Eyesight technology. http://eyesight-tech.com/, view in 2020.

[50] DANIEL VAN NIEUWENHOVE. Softkinetic. http://www.softkinetic.com/en-us/softkinetic.asx, view in 2020.

[51] Chieh-Chih Wang and Ko-Chih Wang. Hand posture recognition using adaboost with sift for human robot interaction. In *Recent progress in robotics: viable robotic service to human*, pages 317–329. Springer, 2007.

[52] Mahmoud Elmezain, Ayoub Al-Hamadi, Samy Sadek, and Bernd Michaelis. Robust methods for hand gesture spotting and recognition using hidden markov models and conditional random fields. In *The 10th IEEE International Symposium on Signal Processing and Information Technology*, pages 131–136. IEEE, 2010.

[53] Mahmoud Elmezain and Ayoub Al-Hamadi. Ldcrfs-based hand gesture recognition. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2670–2675. IEEE, 2012.

[54] pointgrab. Pointgrab ltd. pointgrab. http://www.pointgrab.com/, view in 2020.

[55] Mister Gloves. A wireless usb gesture input system. https://courses.cit.cornell.edu/ee476/FinalProjcts/s2010/ssc88_egl27/References, view in 2021.

[56] Mark Schelbert Faisal Yazadi. Cyber glove systems - worldwide leader in data glove technology. http://www.cyberglovesystems.com/index.php, view in 2020.

[57] Michael Egmont-Petersen, Dick de Ridder, and Heinz Handels. Image processing with neural networks—a review. *Pattern recognition*, 35(10):2279–2301, 2002.

[58] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[59] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[60] Paul Werbos. Beyond regression:" new tools for prediction and analysis in the behavioral sciences. *Ph. D. dissertation, Harvard University*, 1974.

[61] Norah Alnaim. *Hand gesture recognition using deep learning neural networks*. PhD thesis, Brunel University London, 2020.

[62] David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106, 1962.

[63] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE, 2012.

[64] Cicero Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 69–78, 2014.

[65] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[66] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[67] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545, 2014.

[68] David B Fogel, Eugene C Wasson, Edward M Boughton, and Vincent W Porto. A step toward computer-assisted mammography using evolutionary programming and neural networks. *Cancer letters*, 119(1):93–97, 1997.

[69] Shih-Chung B Lo, Heang-Ping Chan, Jyh-Shyan Lin, Huai Li, Matthew T Freedman, and Seong K Mun. Artificial convolution neural network for medical image pattern recognition. *Neural networks*, 8(7-8):1201–1214, 1995.

[70] Berkman Sahiner, Heang-Ping Chan, Nicholas Petrick, Datong Wei, Mark A Helvie, Dorit D Adler, and Mitchell M Goodsitt. Classification of mass and normal breast tissue: a convolution neural network classifier with spatial domain and texture images. *IEEE transactions on Medical Imaging*, 15(5):598–610, 1996.

[71] Mohamed N Ahmed and Aly A Farag. Two-stage neural network for volume segmentation of medical images. *Pattern Recognition Letters*, 18(11-13):1143–1151, 1997.

[72] Adhish Prasoon, Kersten Petersen, Christian Igel, François Lauze, Erik Dam, and Mads Nielsen. Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In *International conference on medical image computing and computer-assisted intervention*, pages 246–253. Springer, 2013.

[73] Wenlu Zhang, Rongjian Li, Houtao Deng, Li Wang, Weili Lin, Shuiwang Ji, and Dinggang Shen. Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *NeuroImage*, 108:214–224, 2015.

[74] ImageNet. Imagenet large scale visual recognition challenge 2015. `http://image-net.org/challenges/LSVRC/2015/index`, view in 2020.

[75] kaggle. galaxy zoo the galaxy challenge. `https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge.`, view in 2020.

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[77] Jon Coaffee. Rings of steel, rings of concrete and rings of confidence: designing out terrorism in central london pre and post september 11th. *International Journal of Urban and Regional Research*, 28(1):201–211, 2004.

[78] Gil Levi and Tal Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 34–42, 2015.

[79] Mel Slater, Bernhard Spanlang, Maria V Sanchez-Vives, and Olaf Blanke. First person experience of body transfer in virtual reality. *PloS one*, 5(5):e10564, 2010.

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

[81] Michael J Jones and James M Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96, 2002.

[82] Vladimir Vezhnevets, Vassili Sazonov, and Alla Andreeva. A survey on pixel-based skin color detection techniques. In *Proc. Graphicon*, volume 3, pages 85–92. Moscow, Russia, 2003.

[83] Jure Kovac, Peter Peer, and Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE, 2003.

[84] Alberto Albiol, Luis Torres, and Edward J Delp. Optimum color spaces for skin detection. In *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, volume 1, pages 122–124. IEEE, 2001.

[85] Maricor Soriano, Birgitta Martinkauppi, Sami Huovinen, and Mika Laaksonen. Skin detection in video under changing illumination conditions. In *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, volume 1, pages 839–842. IEEE, 2000.

[86] Benjamin D Zarit, Boaz J Super, and Francis KH Quek. Comparison of five color models in skin pixel classification. In *Proceedings International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems. In Conjunction with ICCV'99 (Cat. No. PR00378)*, pages 58–63. IEEE, 1999.

[87] Kenny Morrison and Stephen J McKenna. An experimental comparison of trajectory-based and history-based representation for gesture recognition. In *International Gesture Workshop*, pages 152–163. Springer, 2003.

[88] Stan Birchfield. Elliptical head tracking using intensity gradients and color histograms. In *Proceedings. 1998 IEEE Computer Society conference on computer vision and pattern recognition (Cat. No. 98CB36231)*, pages 232–237. IEEE, 1998.

[89] Wen-Hsiang Lai and Chang-Tsun Li. Skin colour-based face detection in colour images. In *2006 IEEE International Conference on Video and Signal Based Surveillance*, pages 56–56. IEEE, 2006.

[90] Danhang Tang, Hyung Jin Chang, Alykhan Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3786–3793, 2014.

[91] Sameh Khamis, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2540–2548, 2015.

[92] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

[93] Dorin Comaniciu, Visvanathan Ramesh, and Peter Meer. Real-time tracking of non-rigid objects using mean shift. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 142–149. IEEE, 2000.

[94] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Efficient mean-shift tracking via a new similarity measure. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 176–183. IEEE, 2005.

[95] Gary R Bradski. Computer vision face tracking for use in a perceptual user interface. 1998.

[96] John G Allen, Richard YD Xu, Jesse S Jin, et al. Object tracking using camshift algorithm and multiple quantized feature spaces. In *ACM International Conference Proceeding Series*, volume 100, pages 3–7. Citeseer, 2004.

[97] Zhaowen Wang, Xiaokang Yang, Yi Xu, and Songyu Yu. Camshift guided particle filter for visual tracking. *Pattern Recognition Letters*, 30(4):407–413, 2009.

[98] O-D Nouar, Ganoun Ali, and CANALS Raphael. Improved object tracking with camshift algorithm. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 2, pages II–II. IEEE, 2006.

[99] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *European conference on computer vision*, pages 28–39. Springer, 2004.

[100] Changjiang Yang, Ramani Duraiswami, and Larry Davis. Fast multiple object tracking via a hierarchical particle filter. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 212–219. IEEE, 2005.

[101] Zia Khan, Tucker Balch, and Frank Dellaert. An mcmc-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision*, pages 279–290. Springer, 2004.

[102] Caifeng Shan, Tieniu Tan, and Yucheng Wei. Real-time hand tracking using a mean shift embedded particle filter. *Pattern recognition*, 40(7):1958–1970, 2007.

[103] Björn Stenger, Arasanathan Thayananthan, Philip HS Torr, and Roberto Cipolla. Model-based hand tracking using a hierarchical bayesian filter. *IEEE transactions on pattern analysis and machine intelligence*, 28(9):1372–1384, 2006.

[104] Berthold KP Horn and Brian G Schunck. Determining optical flow. In *Techniques and Applications of Image Understanding*, volume 281, pages 319–331. International Society for Optics and Photonics, 1981.

[105] John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

[106] James M Rehg and Takeo Kanade. Visual tracking of high dof articulated structures: an application to human hand tracking. In *European conference on computer vision*, pages 35–46. Springer, 1994.

[107] Shan Lu, Dimitris Metaxas, Dimitris Samaras, and John Oliensis. Using multiple cues for hand tracking and model refinement. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–443. IEEE, 2003.

[108] Timothy P Wallace and Paul A Wintz. An efficient three-dimensional aircraft recognition algorithm using normalized fourier descriptors. *Computer Graphics and Image Processing*, 13(2):99–126, 1980.

[109] Guangyi Chen and Tien D Bui. Invariant fourier-wavelet descriptor for pattern recognition. *Pattern recognition*, 32(7):1083–1088, 1999.

[110] GC-H Chuang and C-CJ Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions on Image Processing*, 5(1):56–70, 1996.

[111] Chia-Hung Wei, Yue Li, Wing-Yin Chau, and Chang-Tsun Li. Trademark image retrieval using synthetic features for describing global shape and interior structure. *Pattern Recognition*, 42(3):386–394, 2009.

[112] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[113] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[114] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[115] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[116] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[117] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011.

[118] Edward Rosten and Tom Drummond. Fusing points and lines for high performance tracking. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1508–1515. Ieee, 2005.

[119] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 510–517. Ieee, 2012.

[120] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. A unified framework for gesture recognition and spatiotemporal gesture segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 31(9):1685–1699, 2008.

[121] Takuichi Nishimura and Ryuichi Oka. Spotting recognition of human gestures from time-varying images. In *Proceedings of the second international conference on automatic face and gesture recognition*, pages 318–322. IEEE, 1996.

[122] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff. Simultaneous localization and recognition of dynamic hand gestures. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, volume 2, pages 254–260. IEEE, 2005.

[123] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-based recognition*, pages 227–243. Springer, 1997.

[124] Thad Starner, Joshua Weaver, and Alex Pentland. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1371–1375, 1998.

[125] Mahmoud Elmezain, Ayoub Al-Hamadi, Jorg Appenrodt, and Bernd Michaelis. A hidden markov model-based continuous gesture recognition system for hand motion trajectory. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008.

[126] Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden markov models for complex action recognition. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 994–999. IEEE, 1997.

[127] Andrew D Wilson and Aaron F Bobick. Parametric hidden markov models for gesture recognition. *IEEE transactions on pattern analysis and machine intelligence*, 21(9):884–900, 1999.

[128] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[129] Patrice Y Simard, David Steinkraus, John C Platt, et al. Best practices for convolutional neural networks applied to visual document analysis. In *Icdar*, volume 3, 2003.

[130] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[131] Gang Yu, Junsong Yuan, and Zicheng Liu. Propagative hough voting for human activity recognition. In *European Conference on Computer Vision*, pages 693–706. Springer, 2012.

[132] Michael S Ryoo et al. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *2011 International Conference on Computer Vision*, pages 1036–1043. IEEE, 2011.

[133] Juergen Gall, Angela Yao, Nima Razavi, Luc Van Gool, and Victor Lempitsky. Hough forests for object detection, tracking, and action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 33(11):2188–2202, 2011.

[134] J Russ. The image processing handbook,(5ta. ed). 6. ee. uu, 2006.

[135] ScienceDirect. Segmentation - an overview — sciencedirect topics.[online], 2019.

[136] PatrickFarley. Computer vision documentation - quickstarts, tutorials, api reference- azure cognitive services," quickstarts, tutorials, api reference - azure cognitive services — microsoft docs. [online]., 2019.

[137] A. Mishra. Machine learning in the aws cloud add intellegence to appllcatgions with amazon sagemaker and amazon rekognition," amazon, 2019. [online], 2019.

[138] SimpleCV. Simplecv [online]., 2019.

[139] Xu-Yao Zhang, Cheng-Lin Liu, and Ching Y Suen. Towards robust pattern recognition: a review. *Proceedings of the IEEE*, 108(6):894–922, 2020.

[140] Larissa Welti-Santos. References: Strang, gilbert. linear algebra and its applications, 1988, harcourt. fukunaga, keinosuke. introduction to statistical pattern recognition, 1990, academic press.

[141] Richard O Duda, Peter E Hart, and David G Stork. Pattern classification. a wiley-interscience publication. *ed: John Wiley and Sons, Inc*, 2001.

[142] Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.

[143] Anil K Jain, Robert P. W. Duin, and Jianchang Mao. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, 22(1):4–37, 2000.

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

[145] Marvin Minsky and Seymour Papert. An introduction to computational geometry. *Cambridge tiass., HIT*, 479:480, 1969.

[146] Sandipgiri Goswami. *Real-time static gesture detection using machine learning*. PhD thesis, Laurentian University of Sudbury, 2019.

[147] Seyyed Ehsan Mahmoudi, Alireza Akhondi-Asl, Roohollah Rahmani, Shahrooz Faghih-Roohi, Vahid Taimouri, Ahmad Sabouri, and Hamid Soltanian-Zadeh. Web-based interactive 2d/3d medical image processing and visualization software. *computer methods and programs in biomedicine*, 98(2):172–182, 2010.

[148] Maria Filomena Santarelli, Vincenzo Positano, and Luigi Landini. Real-time multimodal medical image processing: a dynamic volume-rendering application. *IEEE Transactions on Information Technology in Biomedicine*, 1(3):171–178, 1997.

[149] Mohammad Sameti, Rabab Kreidieh Ward, Jacqueline Morgan-Parkes, and Branko Palcic. Image feature extraction in the last screening mammograms prior to detection of breast cancer. *IEEE journal of selected topics in signal processing*, 3(1):46–52, 2009.

[150] I Nystrom, Filip Malmberg, Erik Vidholm, and Ewert Bengtsson. Segmentation and visualization of 3d medical images through haptic rendering. 2009.

[151] Avnish Patel and Kinjal Mehta. 3d modeling and rendering of 2d medical image. In *2012 International Conference on Communication Systems and Network Technologies*, pages 149–152. IEEE, 2012.

[152] Michal Irani and P Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE transactions on pattern analysis and machine intelligence*, 20(6):577–589, 1998.

[153] Zhongfei Zhang. Mining surveillance video for independent motion detection. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 741–744. IEEE, 2002.

[154] Chang Yuan, Gerard Medioni, Jinman Kang, and Isaac Cohen. Detecting motion regions in the presence of a strong parallax from a moving camera by multiview geometric constraints. *IEEE transactions on pattern analysis and machine intelligence*, 29(9):1627–1641, 2007.

[155] Nicolas Verbeke and Nicole Vincent. A pca-based technique to detect moving objects. In *Scandinavian Conference on Image Analysis*, pages 641–650. Springer, 2007.

[156] Ronan Fablet and Michael J Black. Automatic detection and tracking of human motion with a view-based representation. In *European Conference on Computer Vision*, pages 476–491. Springer, 2002.

[157] R. P. K. Poudel. 3d hand tracking,",. In *roQuest Dissertations Publishing*, 2014.

[158] Poornima Ramachandra and Neelima Shrikhande. Hand gesture recognition by analysis of codons. In *Intelligent Robots and Computer Vision XXV: Algorithms, Techniques, and Active Vision*, volume 6764, page 67640M. International Society for Optics and Photonics, 2007.

[159] Brian Windsor. *MoCap for Artists: Workflow and Techniques for Motion Capture*. Taylor Francis, 2008.

[160] Mao Ye, Qing Zhang, Liang Wang, Jiejie Zhu, Ruigang Yang, and Juergen Gall. A survey on human motion analysis from depth data. In *Time-of-flight and depth imaging. sensors, algorithms, and applications*, pages 149–187. Springer, 2013.

[161] Noor Adnan Ibraheem and Rafiqul Zaman Khan. Survey on various gesture recognition technologies and techniques. *International journal of computer applications*, 50(7), 2012.

[162] Prashan Premaratne. *Human computer interaction using hand gestures*. Springer Science Business Media, 2014.

[163] Sushmita Mitra. Senior member, ieee, and tinku acharya, senior member, ieee,"gesture recognition: A survey". *IEEE Transactions On Systems, Man, And Cybernetics—part C: Applications And Reviews*, 37(3):311, 2007.

[164] M. Al-Rajab. "hand gesture recognition for multimedia applications.". In *ProQuest Dissertations Publishing*. School of Computing, University of Leeds, 2008.

[165] Sangheon Park, Sunjin Yu, Joongrock Kim, Sungjin Kim, and Sangyoun Lee. 3d hand tracking using kalman filter in depth space. *EURASIP Journal on Advances in Signal Processing*, 2012(1):36, 2012.

[166] Amar Aggoun, Emmanuel Tsekleves, Mohammad Rafiq Swash, Dimitrios Zarpalas, Anastasios Dimou, Petros Daras, Paulo Nunes, and Luí Ducla Soares. Immersive 3d holoscopic video system. *IEEE MultiMedia*, 20(1):28–37, 2012.

[167] P Supplies. Holga 120-3d stereo camera. `https://www.freestylephoto.biz/194120-Holga-120-3D-Stereo-Camera`, view in 2021.

[168] P Wadekar. gesture recognition. https://www.slideshare.net/PrachiWadekar/gesture-recognition-21207480, view in 2021.

[169] T Deyle. Low-cost depth cameras (aka ranging cameras or rgb-d cameras) to emerge in 2010. http://www.hizook.com/blog/2010/03/28/low-cost-depth-cameras-aka-ranging-cameras-or-rgb-d-cameras-emerge-2010, view in 2020.

[170] tequipment.net. Flir e60 infrared compact thermal camera. https://www.tequipment.net/FLIRE60.html, view in 2020.

[171] V Turk. the vr controller of the future could be your own hands. https://www.vice.com/en_us/article/bmv5za/the-vr-controller-of-the-future-could-be-your-own-hands, view in 2020.

[172] Single Camera. Single camera live projection package. https://www.hawaiicamera.com/rent/single-camera-live-projection-package, view in 2020.

[173] R M Ltd. "single camera shooting. http://www.toptelly.co.uk/single-camera-unit, view in 2020.

[174] U Brunel. Holoscopic 3d vision. https://www.brunel.ac.uk/research/Projects/Horoscopic-3D-Vision, view in 2020.

[175] Mohammad Swash et al. *Holoscopic 3D imaging and display technology: Camera/processing/display*. PhD thesis, Brunel University London., 2013.

[176] Amar Agooun, Obaidulah Abdul Fatah, Juan C Fernandez, Caroline Conti, Paulo Nunes, and Luís Ducla Soares. Acquisition, processing and coding of 3d holoscopic content for immersive video systems. In *2013 3DTV Vision Beyond Depth (3DTV-CON)*, pages 1–4. IEEE, 2013.

[177] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. *Light field photography with a hand-held plenoptic camera*. PhD thesis, Stanford University, 2005.

[178] Jonathan Sachs. Digital image basics. *Digital Light Color*, 1999, 1996.

[179] Melanie Cofield. Digital imaging basics. *Information Technology Lab School of Information The University of Texas at Austin, Summer*, 2005.

[180] Daisuke Sugimura, Takuya Mikami, Hiroki Yamashita, and Takayuki Hamamoto. Enhancing color images of extremely low light scenes based on

rgb/nir images acquisition with different exposure times. *IEEE Transactions on Image Processing*, 24(11):3586–3597, 2015.

[181] Vikas Kumar Mishra, Shobhit Kumar, and Neeraj Shukla. Image acquisition and techniques to perform image acquisition. *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, 9(01):21–24, 2017.

[182] Vivek Agarwal. Research on data preprocessing and categorization technique for smartphone review analysis. *International Journal of Computer Applications*, 975:8887, 2015.

[183] Alberto Martin and Sabri Tosunoglu. Image processing techniques for machine vision. *Miami, Florida*, pages 1–9, 2000.

[184] Gajanan K.Kharate S.Ghotkar. A novel approach for image segmentation in real time hand gesture recognition for hci. *International Journal of Human Computer Interaction*, 2012.

[185] A.Ghotkar. A novel approach for image segmentation in real time hand gesture recognition for hci. *International Conference(ICSCI-2011),Pentagram Research, Hyderabad*, Jan-2011.

[186] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 2002.

[187] Igor Aleksander and John Taylor. *Artificial Neural Networks, 2: Proceedings of the 1992 International Conference on Artificial Neural Networks (ICANN-92) Brighton, United Kingdom, 4-7 September, 1992*. Elsevier, 2014.

[188] Bernd Fritzke. Growing cell structures–a self-organizing network in k dimensions. In *Artificial Neural Networks*, pages 1051–1056. Elsevier, 1992.

[189] MATLAB. What is deep learning?: How it works, techniques and applications,. `https://www.mathworks.com/discovery/deep-learning.html`, view in 2021.

[190] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning cambridge, 2017.

[191] Shuchao Pang, Juan José del Coz, Zhezhou Yu, Oscar Luaces, and Jorge Díez. Deep learning to frame objects for visual target tracking. *Engineering Applications of Artificial Intelligence*, 65:406–420, 2017.

[192] Simulink. Convolutional neural network -matlab and simulink. https://uk.mathworks.com/solutions/deep-learning/convolutional-neural-network.html, view in 2021.

[193] MathWorks. "trainnetwork,"learn about convolutional neural networks. https://uk.mathworks.com/help/deeplearning/ug/introduction-to-convolutional-neural-networks.html, view in 2021.

[194] Y Heaton, IG Jeff, Y Bengio, and A Courville. Deep learning, genetic programming and evolvable machines. *Nature*, 19(1–2):305–307, 2017.

[195] Mohammad Waseem. How to implement classification in machine learning. https://www.edureka.co/blog/classification-in-machine-learning/, view in 2021.

[196] A. L. C. Barczak, N. H. Reyes, M. Abastillas, A. Piccio, and T. Susnjak. A new 2d static hand gesture colour image dataset for asl gestures. *Research Letters in the Information and Mathematical Sciences*, 15:12–20, 2011. Dataset on accessed 25 July 2021.

[197] Raimundo F Pinto, Carlos DB Borges, Antônio Almeida, and Iális C Paula. Static hand gesture recognition based on convolutional neural networks. *Journal of Electrical and Computer Engineering*, 2019, 2019.

[198] Akash. Image data set for alphabets in the american sign language. https://www.kaggle.com/grassknoted/asl-alphabet, April view in 2021. Dataset on accessed 15 July 2021.

[199] tecperson. Drop-in replacement for mnist for hand gesture recognition tasks. https://www.kaggle.com/datamunge/sign-language-mnist, October view in 2020. Dataset on accessed 25 July 2021.

[200] Steve Lawrence, C Lee Giles, Ah Chung Tsoi, and Andrew D Back. Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1):98–113, 1997.

[201] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[202] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[203] P. Ghamisi, Y. Chen, and X. X. Zhu. A self-improving convolution neural network for the classification of hyperspectral data. *IEEE Geoscience and Remote Sensing Letters*, 13(10):1537–1541, Oct 2016.

[204] Nicholas Becherer, John Pecarina, Scott Nykl, and Kenneth Hopkinson. Improving optimization of convolutional neural networks through parameter fine-tuning. *Neural Computing and Applications*, 31:3469–3479, Nov 2017.

[205] Liguo Zhou, Rong Zhu, Yimin Luo, Siwen Liu, and Zhongyuan Wang. Improving convolutional neural networks via compacting features. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2946–2950, 2018.

[206] Jie Feng, Lin Wang, Haipeng Yu, Licheng Jiao, and Xiangrong Zhang. Divide-and-conquer dual-architecture convolutional neural network for classification of hyperspectral images. *Remote Sensing*, 11(5):484, 1–26, 2019.

[207] Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. Transforming auto-encoders. In Timo Honkela, Włodzisław Duch, Mark Girolami, and Samuel Kaski, editors, *Artificial Neural Networks and Machine Learning –ICANN 2011*, pages 44–51, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[208] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3856–3866. Curran Associates, Inc., 2017.

[209] Aryan Misra. Capsule networks: The new deep learning network. `https://towardsdatascience.com/capsule-networks-the-new-deep-learning-network-bd917e6818e8`, view in 2020.

[210] Huayu Li. Cognitive consistency routing algorithm of capsule-network. *CoRR*, abs/1808.09062, 2018.

[211] Mohammed Amer and Tomás Maul. Path capsule networks. *CoRR*, abs/1902.03760, 2019.

[212] Zakaria Jaadi. Explanation of principal component analysis (pca). `https://builtin.com/data-science/step-step-explanation-principal-component-analysis`, view in 2021.

[213] Rohith Gandhi. Support vector machine — introduction to machine learning algorithms. `https://towardsdatascience.com/support-vecto`

r-machine-introduction-to-machine-learning-algorithms-934a444fca47, view in 2021.

[214] Czako Zoltan. Svm and kernel svm. https://towardsdatascience.com/svm-and-kernel-svm-fed02bef1200, view in 2021.

[215] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[216] Y. LeCun, C. Cortes, and C. J.C. Burges. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/, view in 2021.

[217] Metin Bilgin and Korhan Mutludoğan. American sign language character recognition with capsule networks. In *2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 1–6. IEEE, 2019.

[218] Murat Taskiran, Mehmet Killioglu, and Nihan Kahraman. A real-time system for recognition of american sign language by using deep learning. In *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, pages 1–5, 2018.

[219] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Multi-modal deep hand sign language recognition in still images using restricted boltzmann machine. *Entropy*, 20(11), 2018.

[220] Monu Verma, Ayushi Gupta, and Santosh Kumar Vipparthi. One for all: An end-to-end compact solution for hand gesture recognition, 2021.

[221] Y. Zhao and L. Wang. The application of convolution neural networks in sign language recognition. In *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, pages 269–272, 2018.