UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA
BIOMÉDICA

TRABAJO FIN DE MÁSTER

DEVELOPMENT OF A COMPUTER VISION BASED ROBOTIC GUIDANCE SYSTEM FOR THE NEUROREHABILITATION OF BRAIN INJURY PATIENTS

SANTIAGO ROS DOPICO

2022

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

MÁSTER UNIVERSITARIO EN INGENIERÍA BIOMÉDICA

TRABAJO FIN DE MÁSTER

DEVELOPMENT OF A COMPUTER VISION BASED ROBOTIC GUIDANCE SYSTEM FOR THE NEUROREHABILITATION OF BRAIN INJURY PATIENTS

Autor
SANTIAGO ROS DOPICO

Tutores
PABLO ROMERO SOROZÁBAL
Dr. D. ÁLVARO GUTIÉRREZ MARTÍN

2022

Resumen

Las lesiones cerebrales traumáticas suponen un gran reto económico, social y sanitario en todo el mundo. La llamada "*pandemia silenciosa*" afecta a millones de individuos cada año, dando lugar a una creciente población de pacientes que viven con importantes discapacidades directamente relacionadas con este trastorno y que, a consecuencia de ellas, presentan dificultades para la realización de actividades básicas de la vida diaria y para la reintegración. La pérdida de movilidad derivada de esta afección está considerada por muchos como la pérdida de actividad más significativa. Por lo tanto, la rehabilitación es esencial tanto en las fases tempranas como en las crónicas de la recuperación ya que la fisioterapia intensiva produce mejoras significativas en las funciones motoras, como la fuerza muscular, que con frecuencia limitan la independencia de estos individuos.

Las últimas decadas han atestiguado un amplio y rápido desarrollo de la robótica en el ámbito de la rehabilitación dada su capacidad de proporcionar un entorno de entrenamiento estandarizado, de ofrecer un apoyo adaptable al estado actual del paciente y de aumentar la intensidad y la dosis de la terapia. Dado que el éxito de la misma viene determinado, en gran medida, por la motivación y la participación activa de los pacientes, este Trabajo de Fin de Máster propone un sistema de guiado basado en visión artificial para los sistemas robóticos de asistencia a la marcha. Su finalidad es dotar a estos dispositivos de la capacidad de reconocer objetos o personas en su entorno de manera que sean capaces de guiar el movimiento del paciente hacia un objetivo físico, además de proporcionar una fuerza supletoria, dando así al usuario un incentivo para caminar.

Para lograr este objetivo, se ha desarrollado un entorno de detección y reconocimiento de objetos, que puede adaptarse a cualquier caso de uso específico, a través de la implementación de múltiples técnicas de procesamiento de imágenes digitales. Desde los métodos clásicos de segmentación hasta las últimas tendencias en inteligencia artificial, se han elaborado y evaluado en diversas condiciones experimentales un amplio conjunto de algoritmos con el fin de detectar objetos en tiempo real. Además, se han confeccionado un mecanismo de cálculo de coordenadas y un controlador cinemático inverso para el despliegue del sistema de navegación en un andador inteligente con soporte parcial del peso y tracción motorizada tan solo con una cámara USB y un ordenador de placa única.

Palabras clave: Lesión cerebral traumática, rehabilitación, robótica, visión artificial, inteligencia artificial.

Abstract

Traumatic brain injury presents a major economic, social and health challenge worldwide. The so-called "*silent pandemic*" afflicts millions of individuals every year, giving rise to a growing population of patients living with significant disabilities directly related to this disorder who struggle with basic activities of daily living, community participation and reintegration. The loss of mobility derived from this condition is considered by many as the most significant loss of activity. Therefore, rehabilitation is essential during both early and chronic stages of recovery, with intensive physical therapy yielding significantly better motor function outcomes such as muscle strength, which frequently limits these patient's self-independence.

The past decades have witnessed vast and rapid developments of robots for the rehabilitation of sensorimotor deficits given their ability to supply a standardised training environment, to provide adaptable support to the patient's actual state and to increase therapy intensity and dose. As the therapy's success is in large part determined by the active physical and cognitive engagement of patients and their motivation, this Master's Thesis proposes a computer-vision based guidance system for assistive walking robots that aims to grant these devices the ability to recognise objects or people in their surroundings so that they are capable of guiding the patient's movement towards a physical target as well as providing a helping force, thus giving the user a compelling incentive to walk.

To achieve this goal, an object detection-recognition framework, that can be adapted to any specific use case, has been developed. A collection of different digital image processing techniques, from classical segmentation methods to the latest trends in artificial intelligence, are implemented and evaluated under different experimental conditions for the purpose of real-team detection. A coordinate finding mechanism and a robotic inverse kinematic controller have also been elaborated for the deployment of the navigation system on a partial body weight-supported and traction-powered assistive walker with a USB camera and a single-board computer.

Keywords: Traumatic brain injury, rehabilitation, robotics, computer vision, artificial intelligence.

Contents

Resumen **iii**

Abstract **iv**

Contents **v**

List of Figures **viii**

List of Tables **x**

List of Acronyms **xi**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Traumatic brain injury . . . 2
1.2.1 Impact . . . 2
1.2.1.1 Demographic . . . 2
1.2.1.2 Socioeconomic . . . 3
1.2.2 Causes . . . 4
1.2.3 Pathophysiology . . . 4
1.2.4 Consequences . . . 5
1.2.5 Therapeutics . . . 6
1.3 Neurorehabilitation and Neuroplasticity . . . 7
1.4 Project scope and objectives . . . 9
1.5 Document layout . . . 10

2 State of the Art **11**
2.1 Robotics . . . 11
2.1.1 Present-day solutions . . . 11
2.1.2 Architecture . . . 13
2.1.3 Control System . . . 14
2.1.4 Robotic Operating System (ROS) . . . 14
2.1.5 Robotics in medicine . . . 15
2.2 Artificial Intelligence . . . 16
2.2.1 Computer vision . . . 16
2.2.2 AI, ML and DL . . . 17
2.2.3 Supervised learning algorithms . . . 17

2.2.3.1 K-nearest neighbours (KNN) . . . 18
2.2.3.2 Random forest (RF) . . . 18
2.2.3.3 Support vector machine (SVM) . . . 18
2.2.4 Unsupervised learning algorithms . . . 19
2.2.4.1 K-means . . . 19
2.2.5 Artificial neural networks (ANNs) . . . 19
2.2.6 Development pipeline . . . 20

3 Classical Object Detection and Segmentation 21
3.1 Materials and methods . . . 21
3.2 Classical image segmentation . . . 22
3.3 Colour-based object detection . . . 23
3.4 Edge-based object detection . . . 26
3.5 Region-based object detection . . . 30
3.6 Limitations . . . 31

4 Artificial Intelligence Based Recognition 32
4.1 Image classifiers into object detectors . . . 32
4.1.1 Image pyramid and sliding window approach . . . 33
4.1.2 Selective search . . . 34
4.1.3 Non-maximum suppression (NMS) . . . 34
4.2 ML vs DL classification . . . 35
4.2.1 Feature descriptors for ML . . . 35
4.2.1.1 GLCM . . . 35
4.2.1.2 LBP . . . 36
4.2.1.3 HOG . . . 36
4.2.1.4 SIFT . . . 36
4.2.2 CNNs for DL . . . 37
4.3 End-to-end object detection . . . 37
4.3.1 YOLO . . . 38
4.4 Implementation . . . 38
4.4.1 Dataset generation . . . 38
4.4.2 Training . . . 38
4.4.3 Testing . . . 40

5 Deployment on a Robotic Rehabilitation System 45
5.1 Object coordinate calculation . . . 45
5.1.1 3D position estimation . . . 45
5.1.2 Camera calibration . . . 47
5.1.3 Implementation . . . 48
5.2 Trajectory planning . . . 50
5.3 Materials . . . 51
5.3.1 *Jetson Nano* . . . 51
5.3.2 *Swalker* . . . 52
5.4 ROS integration . . . 53

6 Conclusions and Future Developments **55**
6.1 Conclusions 55
6.2 Future developments 56

A Ethical, economic, social and environmental impact **72**
A.1 Introduction 72
A.2 Description of relevant impacts related to the project 73
A.3 Conclusion 74

B Economic Budget **75**

List of Figures

1.1 Age-standardised incidence of traumatic brain injury per 100.000 population by location for both sexes (2016) [24]. 3

2.1 Selection of state-of-the-art robot systems. *(a)* Ghost Robotic's *Vision-60* robot [48] *(b) Ameca* Humanoid Robot AI Platform [49] *(c)* NASA's *VIPER* rover [50] *(d) Ocean One*: The Humanoid Remotely Operated Vehicle (ROV) [51]. 12
2.2 Basic architecture of robotics [55]. 13
2.3 The *da Vinci* surgical system for robotic-assisted surgery [64]. 15
2.4 Position of ML and DL within the area of AI [72]. 17
2.5 ANN structure: input layer, hidden layers and output layer [84]. . . . 20

3.1 USB Camera [93]. 22
3.2 Colour-based object detection algorithm. 23
3.3 Colour-based object detection test. *(a)* Original frame *(b)* HSV conversion *(c)* Colour-thresholding mask *(d)* Close-open noise removal *(e)* Contour extraction *(f)* Bounding rectangle over detected object. . 25
3.4 Colour-based object detection test under different lighting conditions (*low*, *medium* and *high*). *(a)* Original frame *(b)* Colour-thresholding mask after noise removal *(c)* Bounding rectangle over detected object. 26
3.5 Edge-based object detection algorithm. 27
3.6 Edge-based object detection test. *(a)* Original frame *(b)* Gray scale conversion and Gaussian smoothening *(c)* Edge detection by Canny algorithm *(d)* Raw template *(e)* Canny-processed template *(f)* Correlation matrix *(g)* Bounding rectangle over detected object. . . . 28
3.7 Edge-based object detection test under different lighting conditions (*low*, *medium* and *high*). *(a)* Original frame *(b)* Edge detection by Canny algorithm *(c)* Bounding rectangle over detected object. 29
3.8 Region-based object detection *(a)* Accurate detection for small elements *(b)* Inconsistent detection for same object depending on its size in the image. 31

4.1 Traditional approach to converting image classifiers into object detectors *(a)* Image pyramid [124] *(b)* Sliding window. 33
4.2 Selective search algorithm for automatic region proposal [127]. 34
4.3 NMS *(a)* Original classifier output *(b)* Output after NMS. 35

4.4 Training dataset *(a)* Original layout *(b)* Final classification dataset after data augmentation: positive (*top*) and negative (*bottom*) samples. 39

5.1 Geometry of image formation [155]. 46
5.2 Camera calibration *(a)* Checkerboard pattern with world coordinate system axes *(b)* Corner detection. 48
5.3 Object coordinate calculation test *(a)* Short distance: $\{X_w, Z_w\} = \{$*-12, 60* $\}$ *cm (b)* Medium distance: $\{X_w, Z_w\} = \{$*16.5, 90* $\}$ *cm (c)* Long distance: $\{X_w, Z_w\} = \{$*34.5, 120* $\}$ *cm.* 50
5.4 Motion planning: α turn followed by a straight line trajectory. 51
5.5 Hardware deployment *(a) Jetson Nano* single-board computer [160] *(b)* 64 GB micro SD card [161]. 52
5.6 The *Swalker* robotic platform for early mobilisation and ambulation of individuals with limited range of motion and strength in their lower limbs [162]. 53
5.7 Final algorithm that generates a motion trajectory from an input image in real-time. 54

List of Tables

4.1 Evaluation metrics of ML and DL classifiers. Colour code: accuracy, sensitivity, specificity and ROC AUC score. 41
4.2 Colour code: mAP (conf $>$ 0.5, IoU $>$ 0.5), mAP (conf $>$ 0.8, IoU $>$ 0.5), mAP (conf $>$ 0.5, IoU $>$ 0.8) and mean detection time (s). . . . 43

B.1 Economic budget for human resources. 75
B.2 Economic budget for material resources. 75
B.3 Total costs. 76

List of Acronyms

AGV: Automated Guided Vehicles.

AI: Artificial Intelligence.

AMR: Autonomous Mobile Robot.

ANN: Artificial Neural Network.

AUC: Area Under Curve.

BLOB: Binary Large OBject.

CFR: Case Fatality Rate.

CNN: Convolutional Neural Network.

CNS: Central Nervous System.

CP: Continuous-Path.

CSIC: Consejo Superior de Investigaciones Científicas.

DL: Deep Learning.

DRS: Disability Rating Scale.

GCS: Glasgow Coma Scale.

GLCM: Gray Level Co-ocurrence Matrix.

GPU: Graphical Processing Unit.

HIC: High-Income Country.

HoG: Histogram of Oriented Gradients.

HRQoL: Health-Related Quality of Life.

IoU: Intersection over Union.

LBP: Local Binary Pattern.

LMIC: Low and Medium-Income Country.

mAP: Mean Average Precision.

ML: Machine Learning.

NMS: Non-Maximum Suppression.

NPC: Non-Playable Character.

PTP: Point-To-Point.

RF: Random Forest.

ROC: Receiver Operating Characteristic.

ROI: Region Of Interest.

ROV: Remotely Operated Vehicle.

SIFT: Scale-Invariant Feature Transform.

SSD: Single Shot MultiBox Detectors.

TBI: Traumatic Brain Injury.

UK: United Kingdom.

US: United States.

WHO: World Health Organization.

YLD: Year of Healthy Life Lost due to Disability.

Chapter 1

Introduction

Traumatic brain injury (TBI) exerts a **devastating impact** on society. It constitutes a major cause of chronic disability worldwide, with 5.3 million people living with disability resulting from TBI in the US alone (over 2% of the US population), where brain injury is suffered by someone every 15 seconds [1]. Those who suffer brain injury may be left with behavioural, cognitive and executive function **sequelae** lasting days, weeks, years or their entire lifetime [2]. This affects the performance of individual tasks and social interaction and development to a point where the patient may stop taking care of themselves (dressing, eating, walking) and suffer loss of family, work and social environment [3]. Rehabilitation is **essential** after TBI treatment, with widespread evidence proving that early intervention of rehabilitation training yields significantly better treatment outcomes. As a result, this Master's Thesis will focus on developing a computer vision based robotic guidance system to enhance active physical and cognitive engagement of patients during therapy, as well as their motivation, which are all crucial factors for recovery.

1.1 Motivation

TBI presents a major economic, social and health challenge worldwide [4]. Despite the fact that a certain percentage of TBI cases never reach medical care, and thus overall rates for TBIs are most likely underreported, evidence suggests that this condition continues to afflict **millions** of individuals around the world on an annual basis [5][6]. Improved clinical guidelines and significant technological advancements in current treatment regimens have led to a lower rate of related deaths and, consequently, a growing population of individuals living with **significant disabilities** directly related to this disorder [6].

A large proportion of individuals with TBI sustain long-term physical, cognitive, and emotional **impairments** that have a profound impact on their everyday level of functioning, community participation and reintegration [7]. The psychosocial factors associated to this loss of functionality and occupational performance include loss of self-esteem, depression and loss of quality of life [8]. In fact, participation in daily life activities and work is identified by patients, their families and healthcare professionals as one of the **most important** outcomes of TBI-rehabilitation [9].

The value a person places on a particular outcome acts as a decisive factor in its accomplishment [10]. Therefore, directing patient treatment towards goals of paramount importance to the patient heavily contributes towards the therapy's success as this is in large part determined by the patient's **motivation** [11]. For many individuals with TBI, loss of mobility is the most significant loss of activity [12]. As such, the rehabilitation process must target the enhancement of motor impairments and functions, area in which rehabilitation robots have already proven their **effectiveness** [13]. Consequently, as a result of their implementation, the effect of TBI both on patients and their families will be reduced as it depends not only on the injury's severity, but also on the **quality** of the rehabilitation services provided [14].

1.2 Traumatic brain injury

TBI, also known as acquired brain injury, is a broad term that describes a vast array of injuries that happen to the brain when it is damaged by a sudden, external, physical assault [15] [16]. It may happen when there is a blow, bump, or jolt to the head, in which case it is a **closed head** injury, or when an object penetrates the skull, in which case it is referred to as a **penetrating** injury [15]. There are various forms of TBI ranging from mild alterations of consciousness to an unrelenting comatose state and death [6].

1.2.1 Impact

TBI, the so-called **"*silent pandemic*"**, is a leading cause of disability in all regions of the globe, with approximately 69 million individuals sustaining a TBI each year worldwide [17] [18] [19]. The frequency of brain injury is currently **higher** than that of any other disease, including notorious diseases such as breast cancer, AIDS, Parkinson's disease and multiple sclerosis, as it affects all age groups and both genders [20]. Accounting for an estimated global incidence rate of 351–939 cases per 100.000 population, it contributes to death and disability more than **any other traumatic insult** [19] [17]. These staggering figures come as a result of the persistent rise in the prevalence of this condition during the last decades, in large part due to the the increased motorisation and urbanisation in low and medium-income countries (LMICs) which has created new and multiple risks of TBI [21] [19]. In fact, the World Health Organization (WHO) had already predicted by the mid-2000s that TBI would surpass many diseases as the **major cause** of death and disability and become the third largest cause of global disease burden by 2020 [22] [23].

1.2.1.1 Demographic

The incidence, prevalence and expected duration of disability from TBI differ between global regions [18]. Overall, the highest incidence rates are found in central Europe, eastern Europe and central Asia, as can be observed in Figure 1.1 [24]. However, proportionally, LMICs experience nearly **three times** more cases of traumatic brain injury than high-income countries (HICs), with Southeast Asian and

Western Pacific regions showcasing the greatest overall burden [21] [18]. Discrepancy is also witnessed according to other geographic and demographic factors such as **rural** and **urban** areas, which have been proven to experience significantly dissimilar TBI incidence rates in various countries, including China and the United States (US) [21].

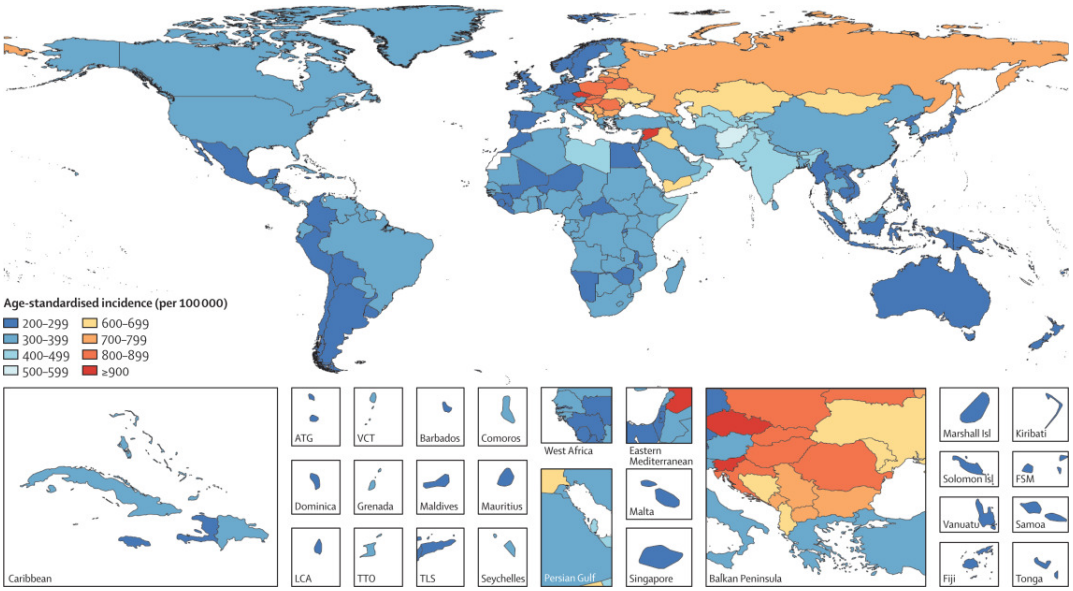


Figure 1.1: Age-standardised incidence of traumatic brain injury per 100.000 population by location for both sexes (2016) [24].

Age-related TBI differences demonstrate three main age groups with the highest prevalence: early childhood (0-4 y), late adolescence or early adulthood (15-24 y) and elderly (> 65 y) [21] [6]. TBI is of particular importance in children and adults younger than 35 years, groups in which this pathology constitutes the **leading** cause of long-term disability [23]. However, in terms of the lesion's repercussions, moderate and severe TBI are most common in individuals aged above 15 years [23]. In reference to gender, the global trend illustrates that most individuals with TBI are **men**, with a prevalence rate between 1.5 and 2.5 times that of women excluding the 8^{th} and 9^{th} decades of life [23] [21].

1.2.1.2 Socioeconomic

Besides being a personal tragedy, TBI is also a public socioeconomic problem [14]. These injuries do not only cause health loss and disability for individuals and their families, but also represent a **burden** to health-care systems and economies [24]. Moreover, there is an increased risk of job loss when incurring a TBI, resulting in a decline of productivity that constitutes a **larger share** of the total societal costs associated to this condition than direct health care costs [25]. In addition, emotional distress and decreased quality of life among caregivers and close family members have been reported several times as well as **caregiver burden** [26]. Furthermore, studies

have shown how family members report personality change, which may very well affect the marital relationship, leading to separation or divorce rates in the years following injury as high as 49% [27].

The magnitude of the economic cost associated to this condition has led to its recognition by the WHO as a **"*critical public health problem*"** to worldwide healthcare systems. In the case of the US, where a TBI occurs every 15 seconds, an estimated 5.3 million Americans are living today with long-term disabilities directly related to TBI, costing the country between $56 and $76.5 billion per year considering costs for disability and loss of productivity [20] [22] [21]. Total annual economic costs of TBI in Europe were estimated to be **€33 billion** ($36.8 billion) in 2010, corresponding to €8,809 ($9,820) per patient, as opposed to $13,000 in the US [14]. However, the previously mentioned transition in LMICs towards motorisation and urbanisation has rendered these nations more **susceptible** to higher socioeconomic burden given their insufficient health care and poor preventive strategies [21].

1.2.2 Causes

TBIs are mainly caused by external kinetic forces to the head, which occur frequently in the context of road traffic collisions, interpersonal violence, work environments, subsequent to falls and during sporting activities [19]. Overall, **falls** and **motor vehicle accidents** are the two leading causes of TBI [6]. Falls in particular are the main cause of TBI and years of healthy life lost due to disability (YLDs) attributable to TBI, accounting for more than **50%** of the age-standardised incidence in some regions such as central Europe [24].

The primary causes of TBI also vary by age, socioeconomic factors, geographic region and political circumstances (i.e. conflict areas) [21]. The occurrence of TBIs among the younger population is mainly due to collisions in the road environment, while falls account for a higher proportion of TBIs among older people [19]. The proportion of head injuries following road traffic collisions and TBIs secondary to these events is greatest in LMICs, specially Africa and Southeast Asia (56%), whilst injuries related to **violence** present the highest incidence in South America, the Caribbean and Sub Saharan Africa [17] [21]. Furthermore, TBI exhibits a close link with **alcohol consumption** whereby head injury incidence in acutely intoxicated patients can reach figures as high as 65% in certain countries such as the United Kingdom (UK) [21].

1.2.3 Pathophysiology

Despite recent advances, our knowledge on the pathophysiology of TBI and its underlying mechanisms remains limited [28]. The magnitude of the TBI epidemic is matched only by the sheer **complexity** of the cerebral pathophysiology involved, as all of its intrinsic factors, including injury severity, type and location or the individual's age and gender, contribute towards producing **unique** brain pathologies, meaning that no two TBIs are the same [20]. The importance of this field of research lies

in the fact that there are secondary effects ensuing the initial traumatic event that often progress slowly over months to years, thus providing a **window** for therapeutic interventions that can be acted upon if their processes are properly understood [28].

As mentioned, the damage to neuronal tissues associated with TBI falls into two categories: **primary injury**, which is directly caused by mechanical forces during the initial insult, and **secondary injury**, which refers to further tissue and cellular damages following the primary insult [28]:

- **Primary brain injuries**. The immediate mechanical impact to the brain, involving acute and irreversible damage to the parenchyma, can be both focal, in which case the damage is limited to the injury site, or diffuse, also affecting surrounding tissues [29]. **Focal** brain damage is frequently accompanied by evidence of skull fracture, localised contusion and a concentrated necrotic area of neuronal and glial cells with compromised blood supply, causing the occurrence of hematoma, epidural, subdural and intracerebral hemorrhages [28]. In the most severe form of TBI, the entirety of the brain is affected by a **diffuse** type of injury and swelling [6]. In this scenario, strong tensile forces damage neuronal axons, oligodendrocytes and blood vasculature, leading to brain edema and ischemic brain damage which can trigger cognitive deficits, behavioural changes and hemiparesis depending on the severity of the injury [28].

- **Secondary brain injuries**. The biochemical, cellular and physiological events that occur during primary injury often progress into delayed and prolonged secondary damages which can last from hours to years [28]. TBI is a complex dynamic process that initiates a multitude of **cascades** of pathological cellular pathways that contribute to secondary injuries, including: excitotoxicity, mitochondrial dysfunction, oxidative stress, lipid peroxidation, neuroinflammation, axon degeneration and apoptotic cell death [20]. As a whole, these result in an imbalance between cerebral blood flow and metabolism, inflammatory and apoptotic processes and edema formation, all of which can render **survival** after TBI difficult due to inadequacy in attention, cognition, severe depression, processing of information as well as progression towards other forms of neurodegenerative diseases [30].

1.2.4 Consequences

Symptoms, which vary depending on the type and severity of the injury and the damaged brain area, may appear right away or several days or even weeks later and evolve over time [29]. Currently, the severity of TBI is categorised based on the **Glasgow Coma Scale (GCS)**, in which patients are scored on the basis of clinical symptoms and the resulting overall score classifies their injury as mild, moderate or severe [20]. The risk of sustaining mild TBI, which constitutes between 70 and 90% of all cases, is more than **18 times** greater than the risk for moderate to severe injuries, but these can still result in long-term cognitive and behavioural deficits and might even be associated with increased risk of neurodegenerative diseases such as

Alzheimer's or Parkison's [23] [20]. Symptoms of mild to moderate TBI can include headaches, dizziness, nausea and amnesia, although they usually resolve within days to weeks after the insult [20].

Of all types of injury, those to the brain are among the most likely to result in death or permanent disability [21]. The Case Fatality Rate (CFR) is in large part determined by **age** and injury **severity**, ranging from 0.9 to 7.6 per 100 TBI patients and from 29 to 55 per 100 severe injury patients [20] [21]. Regarding mortality, out of those who die from this condition 68% do so before reaching a hospital and fatalities are considerably higher in LMICs [21] [19]. Of the two million Americans that are annually treated as a result of TBI, an estimated 56,000 individuals die whilst 80,000 individuals are estimated to be discharged from the hospital with some TBI-related impairment and need **assistance** with activities of daily living [19].

As aforementioned, patients who have been diagnosed with a TBI are often affected by long-term disabilities including cognitive and physical impairments, behavioural changes, impaired attention and psychological problems such as depression [19]. Many studies have also linked TBI to sleep disturbances, chronic pain and loss of communication skills, all of which disrupt the ability to engage in **daily activities** within the home and community, and thus, negatively impact the Health-Related Quality of Life (HRQoL) [31]. Mobility is a major domain affected by this condition, with people affected by TBI-associated impaired mobility being more likely to experience falls and to be discharged to a long term care facility [32]. These motor as well as sensory deficits are **wide-ranging** and may include motor programmes that are either ineffective or absent, impaired motor memory (especially for motor sequences and postural alignment), impaired feedback and feed forward mechanisms, ataxia, dysmetria, dysdiadochokinesis, and intention tremor [33].

1.2.5 Therapeutics

In the US and Europe, the increased public awareness on this epidemic due to the publicity received by injured athletes and military personnel has uncovered the **lack** of treatment options for a crisis that affects millions [20]. LMICs constitute a particular testament to this acknowledgement, as 80% of individuals living with TBI-related impairments are estimated to live in these countries yet merely **2%** of these have access to rehabilitation services [21]. However, the most successful measures in decreasing TBI-related impairments have been proven to be **preventive** strategies including more rigorous safety measures, legislative changes, educating the general population, improved emergency and neuro-trauma services, and the implementation of evidence-based guidelines in treating survivors [21].

Treatment modalities vary extensively based on the severity of the injury and range from daily cognitive therapy sessions to radical surgery such as bilateral decompressive craniectomies [6]. To date, **hyperbaric oxygen therapy**, defined as the inhalation of 100% oxygen under the pressure greater than 1 atmosphere absolute, is one of

the most important clinical therapies for TBI, with researches suggesting a derived reduction in mortality and enhancement in functional outcomes [2]. Other approaches that have also been reported to exert potential effects on TBI treatment including: noninvasive brain stimulation (transcranial magnetic or direct current stimulation) to alter neuronal excitability, functional electrical stimulation to replace or correct lost function in limbs and organs, computer-aided training combined with audial and visual stimulations for the engagement of different components of impairment (e.g. memory, attention, visual perception, etc.) and behavioural, emotional, and family therapies, crucial for emotional stability and self-confidence [2].

1.3 Neurorehabilitation and Neuroplasticity

Stroke and TBI are two of the most prevalent neurological conditions affecting the central nervous system (CNS) and are the most common disorders for which patients receive inpatient neurological rehabilitation [32]. **Neurorehabilitation** is the Health Sciences discipline dealing with recovery from brain injury sequelae, defined as "*a systematic, functionally oriented service of therapeutic activities that is based on assessment and understanding of the patient's brain-behavioural deficits*" [3]. It maintains a **multidisciplinary** approach where different clinical therapeutic perspectives like neuropsychology, physiotherapy, occupational therapy and speech/language therapy work toward biopsychosocial recovery with field-specific actions [3].

The aim of neurorehabilitation is to improve outcome of function after damage to the CNS, mainly muscle weakness which frequently limits self-independence, through intensive physical therapy [34]. Therefore, despite involving multiple disciplines, they all work collectively towards the ultimate goal of enhancing an individual's capacity to process and use incoming information so as to allow **increased functioning** in everyday life [3]. It is also worth noting that, since "*normal*" movement can only be rarely restored after CNS injuries of this calibre, the objective of rehabilitation is to enable "*simpler*", less well-organised movements to achieve optimal outcome in mobility and **independence** during activities of daily living rather than re-establishing these "*normal*" movement patterns [34].

For over two millennia, rehabilitation of people with neurological damage was based on the recovery of the physical structures of the body without consideration for **mental processes**; with the arrival of the cognitive paradigm during the latter half of the last century, however, the theoretical and scientific bases of neurorehabilitation have been linked to the knowledge developed in cognitive neuropsychology and cognitive neuroscience [3]. In this way, recovery of sensorimotor function after CNS damage is based on the exploitation of **neuroplasticity** according to neurophysiological and clinical insights and evidence from multiple studies both in primates and humans [34]. Neuroplasticity can be viewed as a general umbrella term that refers to the brain's ability to modify, change and adapt both structure and function throughout life in response to experience [35].

Since studies have shown that the major cause of death after TBI is neuronal death and rupture of blood vessels, nerve regeneration and angiogenesis play key roles in functional recovery [2]. Research in neuroscience has shown that the brain and spinal cord retain a remarkable ability to **adapt**, even after injury, through the use of practised movements [36]. Therefore, therapy-induced recovery is mediated by neuroplasticity, and the goal of rehabilitation is thus to maximally exploit neuroplasticity in order to achieve an optimal outcome for the individual patient [34]. However, neuroplasticity is **limited**, with most patients reaching a plateau after recovering approximately 70–80% of the initial impairment, thus suggesting that most of the observed recovery is spontaneous, particularly on upper limb function for which there is no evidence of significant training effects [34].

So far there is no clear understanding of the principles underlying effective neurorehabilitation approaches [37]. Current practice for motor recovery during physical therapy is based on the theory that **repeated mass practice** will lead to the recovery of motor function [38]. It is much like a relearning process exploiting preserved sensorimotor circuits where the relearning can be optimised by providing appropriate proprioceptive stimuli with the goal of **maximally engaging** preserved neural circuits [34]. The extent of recovery depends on the severity of CNS damage and the individual neural capacity of a patient to regain a function [34]. In general, therapeutic protocols can be readily described by the following aspects: the body part trained (e.g., the legs), the tools or machines used for the training (e.g., a treadmill), the activity performed (e.g., walking), and when the therapy commences (e.g., during the acute phase after a stroke) [37].

Rehabilitation is essential after TBI treatment, with studies proving that **early intervention** of rehabilitation training yields significantly better treatment outcomes such as higher disability rating scale (DRS) scores [2]. Repetitive, high dose, task specific training during the acute stages of recovery has been found to enhance beneficial neuroplasticity, accelerate functional recovery and the restoration of healthy gait, and lead to better outcomes during the chronic stages of recovery [38]. However, neurorehabilitation can still be beneficial even **years** after an injury or illness event, with long-term and recurrent therapy helping individuals maintain or advance their functional status [39]. Simultaneously, this practise enables the scientific community to collect valuable data which allows inferring about the principles of brain organisation and the mechanisms of learning new functions or relearning lost ones [37].

There are several factors that must be considered when devising rehabilitation protocols. Firstly, rehabilitation needs change over the course of an illness and as patients adjust to their post-acute environment [39]. These protocols must be **individualised** to each patient, with the common goal of developing the patient's ability to function within his or her unique social and physical environment through therapeutic interventions, education, support, and environmental modifications [39]. Furthermore, active physical and cognitive engagement of patients during therapy are

crucial for recovery, as is motivation, which can be enhanced through feedback about movement performance [34]. Finally, it is also beneficial to take into account that the recovery of motor function is dependent on the interrelationship between dosing, intensity and task specific practice, with recent research indicating that the **amount** of practice in the specific task is more critical than the difficulty and variations of task practice when learning new gait patterns [38].

Physical therapists may not always be able to provide enough high dose, task specific repetitive gait training during the acute stages of recovery where maximum physical assistance is required [38]. Therefore, current practices result in **variable** recovery of motor function, and may also cause residual gait deviations and reduced functional ambulation [38]. Research is focused on increasing the dose administered to individuals to enhance recovery during early stages, with rapid and vast developments in the past decades of **robots** for the rehabilitation of sensorimotor deficits after damage to the CNS [34]. Therapy robots, sometimes called rehabilitators, are machines or tools for rehabilitation therapists that allow patients to perform practice movements aided by the robot [36]. When appropriately applied, robot-assisted therapy can provide a number of advantages over conventional approaches, including a standardised training environment, adaptable support to the actual state of patients and the ability to increase therapy intensity and dose, while reducing the physical burden on therapists [34]. Nevertheless, limitations in functionality and **high costs** continue to largely restrict the availability of rehabilitation robots [36].

1.4 Project scope and objectives

The aim of this Master's thesis is to design and implement a guidance system for assistive walking robots in order to provide them with the ability to **recognise** their environment and hence navigate adequately through it. By means of its application, the selection and labelling of surrounding objects or people will be enabled so that the physician can **direct** the patient towards different physical targets that must be reached during the rehabilitation process. As a result, the patient will be given a compelling incentive to walk and the robot's functionality will be enhanced as it will be capable of guiding the patient's movement as well as providing a helping force. In this way, the expected outcome is to increase the patient's active **engagement** and **motivation** during therapy so as to improve motor recovery, as has been previously discussed throughout this chapter.

To achieve this goal, a computer-vision based approach will be pursued by means of a camera and micro-processor that will both be integrated into a pre-existing robotic system: the ***Swalker*** robotic platform that promotes early weight bearing and mobilisation during the rehabilitation of musculoskeletal diseases. This project will focus mainly on the development of an object detection-recognition **framework** that will allow the identification and localisation of any desired object in the patient's surroundings by selecting, in a simple, accessible and time-efficient manner, the solution that presents the best behaviour for the specific object at hand from an

extensive collection of pre-defined algorithms. Therefore, the following objectives can be set for the accomplishment of this task:

- Analysis and familiarisation with **state-of-the-art technologies** in the fields of robotics and computer vision.
- Investigation and implementation of digital image processing and video analysis techniques for the development of real-time, **object detection-recognition** algorithms.
- Validation of the multiple proposed solutions under different **experimental conditions** for their comparison and inspection of their advantages and limitations.
- Design of an object **co-coordinate finding** mechanism and robotic forward-inverse kinematic controller to implement the guidance system from vision sensor feedback.
- Integration of the developed software into a **robotic operating system** for the hardware deployment of the navigation system.
- Evaluation of **project outcomes** and definition of future lines of work.

1.5 Document layout

In accordance with the previously established set of objectives, this Master's Thesis has been organised into the following chapters:

- **Chapter 2**. The latest advancements and current trends in robotics are presented both in the field of medicine and beyond, alongside detailed explanations on robotic architectures, motion controllers and operating systems. Artificial intelligence (AI) and computer vision tools are also introduced as well as their development pipelines.
- **Chapter 3**. This chapter is dedicated to the exploitation of classical image segmentation techniques, based on thresholds, edges or regions, for the purpose of real-time object detection. Different approaches are pursued and assessed with the aim of reviewing their strengths and weakness in order to select the best possible alternative for the robotic guidance system at hand.
- **Chapter 4**. The application of AI-based technologies to the processing of information contained in digital images is investigated for the completion of object recognition tasks. A considerable cohort of varied solutions are examined, with special emphasis on the comparison between one-stage and two-stage methods which prioritise different outcomes.
- **Chapter 5**. Coordinate calculation and motion planning strategies are elaborated for the deployment of the devised object detection-recognition framework on the *Swalker* robotic platform.

Chapter 2

State of the Art

Artificial intelligence (AI) and robotics are both rapidly evolving fields. On one hand, **AI** is currently being implemented for a myriad of different purposes including personalised shopping, fraud prevention, facial recognition and the creation of smart, human-like non-playable characters (NPCs) to interact with users in video games [40]. On the other, **robots** present a wide variety of use cases that make them the ideal technology for the future, with applications ranging from co-bots in manufacturing plants and autonomous vehicles to surgical assistants or landmine detectors in war zones [41]. **Artificially intelligent robots** are ultimately the bridge between robotics and AI; these are robots which are controlled by AI algorithms, enabling them to perform more complex tasks as opposed to non-intelligent robots, whose limited functionality often constraints their use to applications that only require carrying out a repetitive series of movements [42].

2.1 Robotics

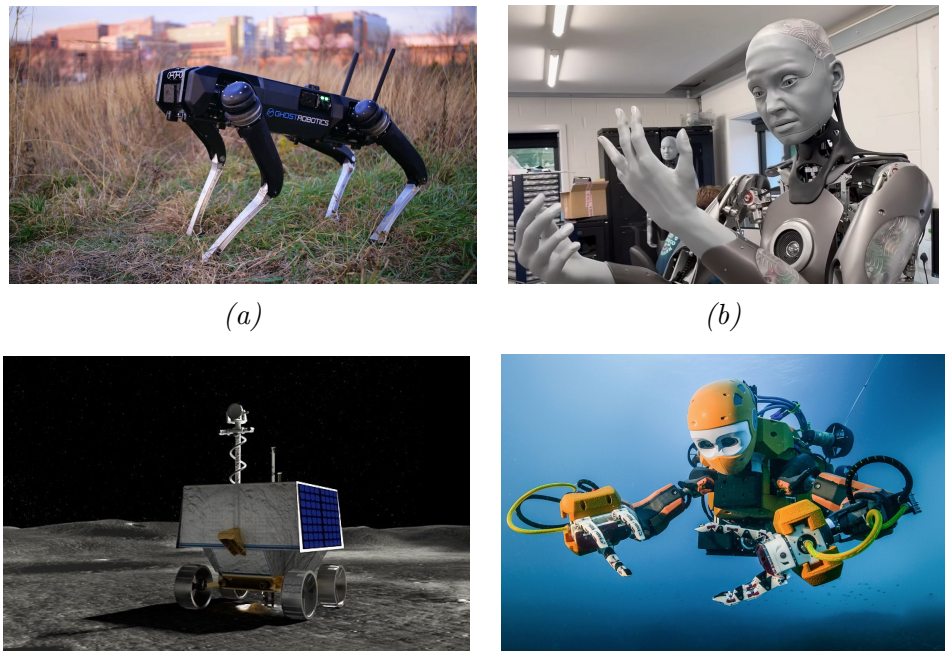
Robotics is an interdisciplinary sector of science and engineering dedicated to the design, construction and use of mechanical robots [41]. There is no exact definition on what a robot is, but by general agreement it is considered a programmable machine that **imitates** the actions or appearance of an intelligent creature, usually a human [43]. It is an exciting time to work in robotics, with plenty of interesting challenges arising in designing machines that intelligently **interact** with both humans and their environment, and a range of techniques and insights from engineering, computer science, physics, biomechanics, psychology and other fields are available to help solve them [44].

2.1.1 Present-day solutions

From carefully harvesting crops to assembling automobiles and delivering medications, robotics solutions are enhancing **productivity**, improving **safety** and enabling greater **flexibility** in a variety of industries [45]. These devices are generally indicated for tasks requiring programmable motions, particularly where those motions should be quick, strong, precise, accurate, untiring, and/or via complex articulations [43]. The number of robots in use worldwide has already multiplied **three-fold** over the

past two decades, with trends suggesting an even faster growth over the next 20 years that is set to boost productivity, economic growth and lead to the creation of new jobs in yet-to-exist industries [46].

Robots can be classified into multiple categories according to numerous criteria such as their **physical configuration** (Cartesian, cylindrical, polar or joint-arm), the **mobility** of the robot's base, which can be fixed (e.g. manufacturing robots) or mobile, or their **control system** [47]. While robotic applications vary greatly, current instruments can be generally grouped into six categories: **autonomous mobile robots (AMRs)**, which move throughout the world while making near real-time decisions, **automated guided vehicles (AGVs)**, that rely on tracks, predefined paths or operator oversight rather than traversing environments freely, **articulated robots**, meant to emulate the functions of a human arm, **humanoids**, which perform human-centric functions and often take human-like forms, **cobots**, designed to function alongside or directly with humans, and **hybrids** of any of the previous categories [45].



(a) *(b)*

(c) *(d)*

Figure 2.1: Selection of state-of-the-art robot systems. *(a)* Ghost Robotic's *Vision-60* robot [48] *(b)* *Ameca* Humanoid Robot AI Platform [49] *(c)* NASA's *VIPER* rover [50] *(d)* *Ocean One*: The Humanoid Remotely Operated Vehicle (ROV) [51].

Some examples of the latest advancements in this field of study have been chosen to capture the current trends and to highlight the wide range of different applications these solutions can enjoy. Ghost Robotics specialises in quadruped robots, including the ***Vision-60*** robot illustrated in Figure 2.1 (a), made for unstructured natural

environments that cannot be traversed by traditional wheeled or tracked robots such as caves, mines, forests and deserts [52]. ***Ameca***, exhibited in Figure 2.1 (b), is the world's most advanced human shaped robot and has been designed specifically as a platform for human-robot interaction through the implementation of smooth, lifelike motion and advanced facial expression capabilities [53]. The ***VIPER*** rover (Figure 2.1 (c)), scheduled to be launched to the southern polar region of the moon in November 2023, will use a variety of instruments to search for water, ice and other resources as well as to map and register terrain [52]. Robotics has also opened up new possibilities in the field of underwater research with ROVs such as the ***Ocean One*** displayed in Figure 2.1 (d), maneuvered and overlooked in real-time by human operators, that enable the exploration of the ocean floor which is, in its majority, too hostile for human explorers [51].

2.1.2 Architecture

The most important factor that distinguishes robot architectures from other software structures is the need to interact **asynchronously**, in real time, with an uncertain, often **dynamic**, environment at varying temporal scopes ranging from millisecond feedback control to minutes, or hours, for complex tasks [54]. Therefore, a common feature of robot architectures is the **modular** decomposition of systems into simpler, largely independent pieces connected by communicating processes, as this design enables each component to handle interactions with the environment asynchronously while minimising interactions with one another, thus decreasing overall system complexity and increasing **reliability** [54].

The basic architecture of automated robotics can be divided into modules that include data **collection**, environment **perception** and understanding, **decision making** and decision **execution**, as shown in Figure 2.2. The data collected from sensors like cameras are then processed and interpreted by advanced algorithms such as motion or path planning algorithms, whose outputs later determine the decisional messages which are finally are passed onto the actuator hardware systems where they are executed [55]. The communication between processes is usually carried out through message passing either in the ***client–server*** style, in which a message request from the client is paired with a response from the server, or in the ***publish-subscribe*** paradigm, which reduces the impact of missing or out-of-order messages by broadcasting them asynchronously in such a way so that all modules that have previously indicated an interest in such messages receive a copy [54].

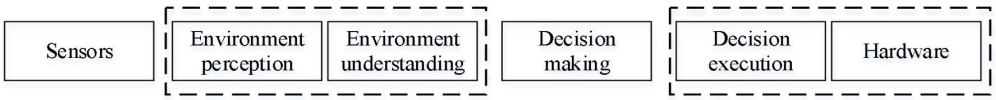


Figure 2.2: Basic architecture of robotics [55].

2.1.3 Control System

The logic that continuously reads from sensors and accordingly updates the actuator commands so as to achieve the desired robot behaviour is the robot's **control system** or **controller** [56]. Examples of control objectives include: ***motion*** *control*, when a robot arm moves along a specified trajectory, ***force*** *control*, where the aim is to apply specific forces to an object in the environment, ***hybrid motion-force*** *control*, to control the motion in some directions and the forces in others, for example, when a gripper opens a door, and ***impedance*** *control*, as when a robot is employed to render a virtual environment [56] [57].

Within motion control, the control method that will be dealt with in this Master's Thesis, there are three main approaches [47]:

1. **Point-to-point (PTP)**: the robot is capable of moving between memory recorded locations, but it does not control the path to get from one point to the other.

2. **Continuous-path (CP)**: the robot is capable of performing movements along a controlled path. All the points along the path must be stored explicitly in the robot's control memory, which is usually achieved by manually displacing the robot through the desired path while the controller unit stores a large number of individual point locations (*teach-in*). Straight-line motion is the simplest example for this type of robot, although continuous-path controlled robots also have the capability of following smooth curve paths defined by the programmer.

3. **Controlled-path robot**: the control equipment can generate paths of different geometry such as straight lines, circles and interpolated curves with a high degree of accuracy at any point along the specified path. Only the start and finish points and the path definition function must be stored in the robot's control memory.

2.1.4 Robotic Operating System (ROS)

Writing software for robots is a **challenging** task, with different types of robots having wildly varying hardware and with an extensive amount of required code, starting from driver-level software and continuing up through perception, abstract reasoning and beyond [58]. Since the necessary breadth of expertise is well beyond the capabilities of any single researcher, robotics software architectures must also support large-scale software **integration** efforts [58]. Many software platforms, sometimes called middlewares, have been proposed with the purpose of easing the construction of robot systems by introducing modular and adaptable features [52]. Over time, some of them have grown to become rich ecosystems of utilities, algorithms and sample applications; however, few rival the ROS in its **significance** on the maturing robotics industry [52].

ROS is an open-source, meta-operating system for robots that provides hardware abstraction, low-level device control and message-passing between processes, among many other services usually expected from an operating system, as well as tools and libraries for obtaining, building, writing, and running code across multiple computers [59]. Its biggest strength is its ability of connecting **nodes** together, pieces of software that take care of a small subset of tasks, such as reading a sensor or controlling a servo, and that can be written in any programming language [60]. An asynchronous publish-subscribe message-passing framework is employed whereby nodes that have information to share will broadcast it using **topics** so that only those nodes that are interested in the information receive it [52] [60]. The multiple advantages offered by ROS have led to its establishment as the standard in robotics programming, quickly becoming the equivalent of *Windows* for PCs or *Android* for mobile phones as any programme that runs on ROS can be shared among many different robots [61].

2.1.5 Robotics in medicine

Medical robotics is causing a **paradigm shift** in therapy, with AI, miniaturisation and computer power contributing towards the rise in the design and use of robots in this field [62] [63]. Medical robots were first introduced around 35 years ago when an industrial robot and computed tomography navigation were used to insert a probe into the brain to obtain a biopsy specimen [63]. Nowadays, new uses for medical robots are created regularly, as in the initial stages of any **technology-driven revolution**, while the use of already existing solutions becomes more consolidated, as is the case with Intuitive Surgical's *da Vinci* system (Figure 2.3), which was already used in 80% of radical prostatectomies in the U.S. just nine years after the system became available on the market [62].

Figure 2.3: The *da Vinci* surgical system for robotic-assisted surgery [64].

The greatest impact of medical robots has been in **surgeries**, where outcomes such as patient trauma or hospital stay can be improved through the precise and accurate manipulation of the necessary tools with robotic assistance [62] [63]. These instruments present a wide variety of **features** that range from 3D vision, tremor filtration and haptic feedback for tactile sensation to infrared eye-tracking, image

guided navigation systems, 4 to 7 degrees of freedom, integrated seats with enhanced ergonomics or polarised glasses [65]. The main **benefits** offered by these devices are, for surgeons, a greater range of motion and dexterity, visualisation of highly-magnified and high-resolution images of the operating field and better access to the area being operated on; for patients, this translates into less risk of infection, lower blood loss and fewer blood transfusions, less pain and quicker return to daily routine [66].

In the area of rehabilitation, two kinds of robots can be distinguished: **assistive** robotic systems, designed to provide more autonomy to people with disabilities by aiding every day tasks such as eating or shaving, and **rehabilitation** systems, that are similar to assistive systems but are designed to facilitate recovery by delivering therapy and measuring the patient's progress [62]. The topic of **exoskeletons** is of particular interest given the number of devices currently being studied as well as purchased by facilities for rehabilitation purposes, emerging as an advantageous tool for disabled individuals [67]. The decades since the introduction of the first powered exoskeletons for therapeutic applications in the 1970s, systems which used pneumatic, hydraulic or electromagnetic actuators for position servocontrol in order to increase patient stability, have seen an **explosion** of novel rehabilitation robots for both the upper and lower extremities [34].

2.2 Artificial Intelligence

AI is a leading technology of the current age of the *Fourth Industrial Revolution*, with the capability of incorporating human behaviour and intelligence into machines or systems [68]. It is a branch of computer science which involves developing algorithms that are able to tackle **cognitive functions** such as learning, perception, problem-solving, language-understanding and/or logical reasoning in order to complete tasks which would otherwise require human intelligence [42]. There are various types of AI including **analytical**, **functional**, **interactive**, **textual** and, most relevantly to the pursued application in this project, **visual** AI, capable of recognising, classifying and sorting items as well as converting images and videos into insights [68].

2.2.1 Computer vision

Computer vision is a field of AI that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs, and take actions or make recommendations based on that information; if AI enables computers to think, computer vision enables them to **see**, **observe** and **understand** [69]. Even though early experiments in computer vision started in the 1950s, its deployment has not grown exponentially until fairly recently, with its market expected to reach $48.6 billion in 2022 [69]. It has become a significant part of everyday life partly due to the vast amount of visual data generated nowadays, with countless images and videos from the built-in cameras of our mobile devices alone [70]. **Any task** enabled by the human sight can be transferred onto machines through computer vision, thus

creating endless applications such as **facial recognition** for police work or payment portals, detection of lane markings and traffic signals in **self-driving cars** or accurate translations of signs in foreign languages [70].

2.2.2 AI, ML and DL

AI, machine learning (ML) and deep learning (DL) are three prominent terminologies used interchangeably nowadays to represent intelligent systems or software [68]. In general, AI is an **umbrella** term that refers to any methodology that combines human behaviour and intelligence into machines or systems, whereas ML is a way of **learning from data** or experience through the application of AI in the form of algorithms to enable automated tasks [71] [68]. ML models are often made up of a set of rules, procedures or sophisticated "*transfer functions*" that can be used to discover interesting data patterns or anticipate behaviours [68]. Finally, DL is a specialised category of ML where the structure of AI algorithms is **layered** and more powerful, creating what are known as artificial neural networks (ANNs) [71]. The relationship between the three areas of study is depicted in Figure 2.4.

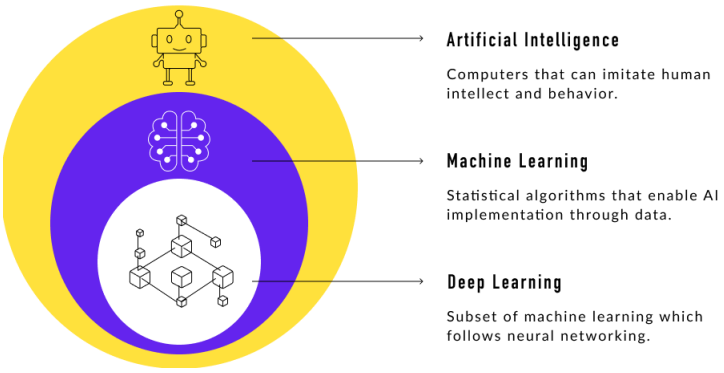


Figure 2.4: Position of ML and DL within the area of AI [72].

2.2.3 Supervised learning algorithms

There are two basic approaches within ML. **Supervised** learning is defined by its use of labeled datasets, comprised of inputs and their corresponding correct outputs, to train algorithms that solve either **classification** or **regression** problems [73]. The first require assigning data into specific categories whilst the second enable understanding the relationship between dependent and independent variables, commonly being used to make projections such as sales revenues for businesses [73]. Image classification is one of the most important applications of these algorithms [74]. It must be performed on the basis of a **vector of parameters** which characterises the visual content of the input images and, thus, enables their mapping to an N-dimensional space, with N being the number of parameters, where they can be more easily categorised. The supervised learning algorithms that will be implemented in this project are presented below.

2.2.3.1 K-nearest neighbours (KNN)

The KNN classifier is by far the most **simple** classification and regression ML algorithm. It is referred to as a "*lazy learner*" because it doesn't perform any training; it simply stores the training data, without performing any calculations, and doesn't build a model until a query is performed on the dataset [75]. It uses **proximity** to make classifications or predictions about the grouping of an individual data point, working off the assumption that similar points can be found near one another in the previously mentioned N-dimensional space [76]. For classification problems, which make up for most of its applications, a class label is assigned to a given data point on the basis of a **majority vote**, that is, the label that is most frequently represented among its k-nearest neighbours [76]. In order to determine whether a data point is a neighbour or not, a distance metric must be calculated between a given point and its closest fellows [75].

2.2.3.2 Random forest (RF)

Decision trees, the most powerful and popular tool for classification and prediction, are the building blocks of the random forest model [77] [78]. These are flowchart-like tree structures where each internal node denotes a test on a feature or attribute, each branch represents an outcome of the test and each leaf node (terminal node) holds a class label [77]. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute and then repeating this process for the subtree rooted at the new node until some leaf node is reached, which provides the classification of the instance [77]. A RF, like its name implies, consists of a large number of individual decision trees that operate as an **ensemble** in which each tree outputs a class prediction and the class with the most votes becomes the model's prediction [78]. The fundamental concept behind RFs is a simple but powerful one: the ***"wisdom of crowds"***, as a large number uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions [78].

2.2.3.3 Support vector machine (SVM)

SVMs are a set of supervised ML methods used for classification, regression and outlier detection [79]. The objective of the support vector machine algorithm is to find a **hyperplane** in the aforementioned N-dimensional space that distinctly classifies the input data [80]. Hyperplanes are decision boundaries that enable the attribution of entries to different classes depending on the side of the plane they fall on, with the optimal hyperplane being the one that presents the maximum **margin**, i.e. distance between data points of both classes [80]. Support vectors are data points that are closer to the hyperplane and influence its position and orientation in order to maximise the classifier's margin according to a pre-defined loss function [80]. Several studies have reported that SVMs are generally capable of delivering higher performance in terms of classification **accuracy** than the other data classification algorithms [74].

2.2.4 Unsupervised learning algorithms

The other main approach in ML is **unsupervised** learning, which employs algorithms to analyse unlabeled datasets, only containing inputs and no associated outputs, with the aim of discovering hidden patterns or data groupings without the need for human intervention [81]. These are **clustering** problems, consisting of identifying homogeneous subgroups within the data such that data points in each cluster are as similar as possible and data points in different clusters as different as possible, according to a similarity measure such as euclidean or correlation based distance [82]. In this case, inferences are made from datasets using only input vectors without referring to known, or labelled, outcomes and so the model's performance cannot be evaluated [83]. The only unsupervised learning algorithm that will be implemented in this project is presented below.

2.2.4.1 K-means

This algorithm tries to **partition** the dataset into K pre-defined, distinct and non-overlapping subgroups or *clusters* where each data point belongs to only one group [82]. A set of centroids, representing the location of each cluster's centre, are randomly selected and iterative calculations are performed to optimise their positions by means of an **expectation-maximisation** approach [83]. The actions that are repetitively carried out in each iteration until the centroids have stabilised are: allocating each data point to the cluster with the closest centroid, updating the centroid coordinates to the average of the all data points that belong to each cluster and computing the sum of the squared distances between data points and all centroids [82]. In this way, the within-cluster sum of the squared distances is minimised, ensuring that each cluster subgroup possesses the optimal **homogeneity** [83].

2.2.5 Artificial neural networks (ANNs)

Although ANNs can be exploited both for supervised and unsupervised learning problems, in this Master's Thesis the former approach will be pursued. The name and structure of these DL algorithms are inspired by the **human brain**, mimicking the way that biological neurones signal to one another [84]. Processing elements, also known as artificial neurones or perceptrons, are **connected** to each other, thus constituting the nodes of the network, and are typically arranged in a layer or vector [85]. The output of one layer serves as the input to the next and possibly other layers, with a given neurone being connected to all or a subset of the neurones in the subsequent layer, simulating the synaptic connections of the brain [85]. In this way, ANNs are comprised of **node layers** which can take the form of an input layer, hidden layers or an output layer, as depicted in Figure 2.5 [84].

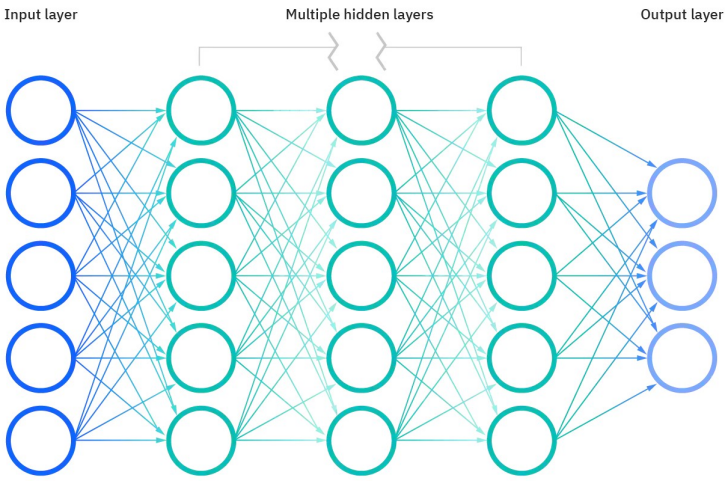


Figure 2.5: ANN structure: input layer, hidden layers and output layer [84].

Each neurone can be thought of as its **own** linear regression model, performing a simple computation on the input data it receives, and has an associated threshold so that it is only activated and, consequently, allowed to send data to the next layer of the network, when its output exceeds a certain boundary [84]. The connections between neurones are also assigned a **weight** that is adjusted during the learning process to alter the strength of the signal carried by that connection according to a specified learning rule until the ANN performs the desired task correctly [86]. When applied on an unseen observation, the network is not only capable of predicting its associated class label, but also of returning the **probability** of the observation belonging to each possible class.

2.2.6 Development pipeline

Contrary to what may seem, identification and **collection** of data, along with their suitable **preparation**, are the most important steps in an AI algorithm's lifecycle since these can only be as good as the quality of the data used for their development [87]. Following data collection and data preparation, the third phase in the AI development pipeline is the proper creation of an intelligent decision-making process, performed following three basic steps: **modeling**, which involves deciding on the algorithm or layers of algorithms to employ in order to interpret the data, **training**, which entails processing large amounts of data through the AI model in iterative test loops while monitoring accuracy to ensure an appropriate model behaviour, and **inference**, which refers to the deployment of the AI model into its real-world use case [71].

Chapter 3

Classical Object Detection and Segmentation

Object detection is a computer vision technique that aims to replicate the human ability to recognise and locate objects of interest in images or videos within a matter of moments [88]. Therefore, it can be defined as "*the task of detecting instances of objects of a certain class within an image*" [89]. With this kind of **identification** and **localisation**, object detection can be used to count objects in a scene and determine and track their precise locations, all while accurately labeling them [90]. This chapter is dedicated to the study of the numerous classical digital image processing and video analysis techniques that can be exploited for the development of a **real-time**, object **detection-recognition** algorithm. Different approaches to this task will be presented and assessed with the aim of selecting the best possible alternative for the robotic guidance system at hand.

3.1 Materials and methods

In order to achieve the aforementioned purpose, the Open Source Computer Vision Library (*OpenCV*) will be employed. Launched by Intel in 1999 with the aim of advancing vision research and disseminating vision knowledge, *OpenCV* is an optimised and portable library of programming functions available for free [91]. The newest release contains more than 2500 optimised algorithms, is used extensively around the world, with over 2.5 million downloads and 40 thousand people in the user group, and is implemented in both academic and commercial applications [91].

This library provides a simple interface that enables video capture from a file or device and **frame-by-frame manipulation** [92]. In this MSc Thesis, live stream was collected from the USB camera displayed in Figure 3.1. It possesses an acquisition speed of 30 frames per second (fps), automatic correction in low light conditions and full 360^{o} rotatory movement [93]. Once the connection with this device is established by means of a *Python* script, an infinite loop extracts each frame from the live feed, applies different image segmentation techniques and, finally, displays the resulting outcome.

Figure 3.1: USB Camera [93].

3.2 Classical image segmentation

In computer vision, **segmentation** is the process of partitioning a digital image into multiple segments (sets of pixels, also known as super pixels) to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse [94]. Therefore, it can be defined more precisely as the process of assigning a label to every pixel in an image such that pixels with the same label are **similar** with respect to some characteristic or computed property, for instance colour, intensity, or texture [94]. The combination of a segmentation technique followed by classification, where the separated homogeneous regions are assigned to particular classes, is regarded as the **elementary component** of computer vision [95].

A wide range of segmentation techniques are available and can be classified into two main categories: **classical** segmentation methods, mainly edge based, region based and threshold based, and **AI** based strategies, mainly ML and DL [95]. Even if these last technologies have pushed the limits of what was possible in the domain of digital image processing, that is not to say that traditional computer vision techniques have become obsolete [96]. In fact, these still pose a **better solution** for some problems while simultaneously overcoming the many challenges AI brings, including, among others, computing power and quantity of inputs [96].

The following sections will review the classical segmentation methods which are less resource-intensive as opposed to AI techniques. These are **thresholding** based segmentations, where an optimum threshold separates the histogram into classes while minimising intra-class variance and maximising inter-class variance, **edge** based segmentations, which depend on local changes in image intensity, and **region** based segmentations, that rely on seed points from which the regions grow by adding neighbouring pixels according to their intensity [95].

3.3 Colour-based object detection

Thresholding is the simplest way to segment images by dividing the image pixels into different groups concerning their intensity values, usually obtaining binary images where the pixel values below the selected threshold are set to 0 (background pixels) and the ones above said limit are set to 1 (foreground pixels) [97]. In computer vision, this procedure is employed to detect **monochromatic objects** by filtering all of the pixel values within a specific range of colour so that every object whose colour falls within the specified range is changed to white and the rest of the image is left black [98]. The optimised algorithm developed for this purpose is displayed in Figure 3.2.

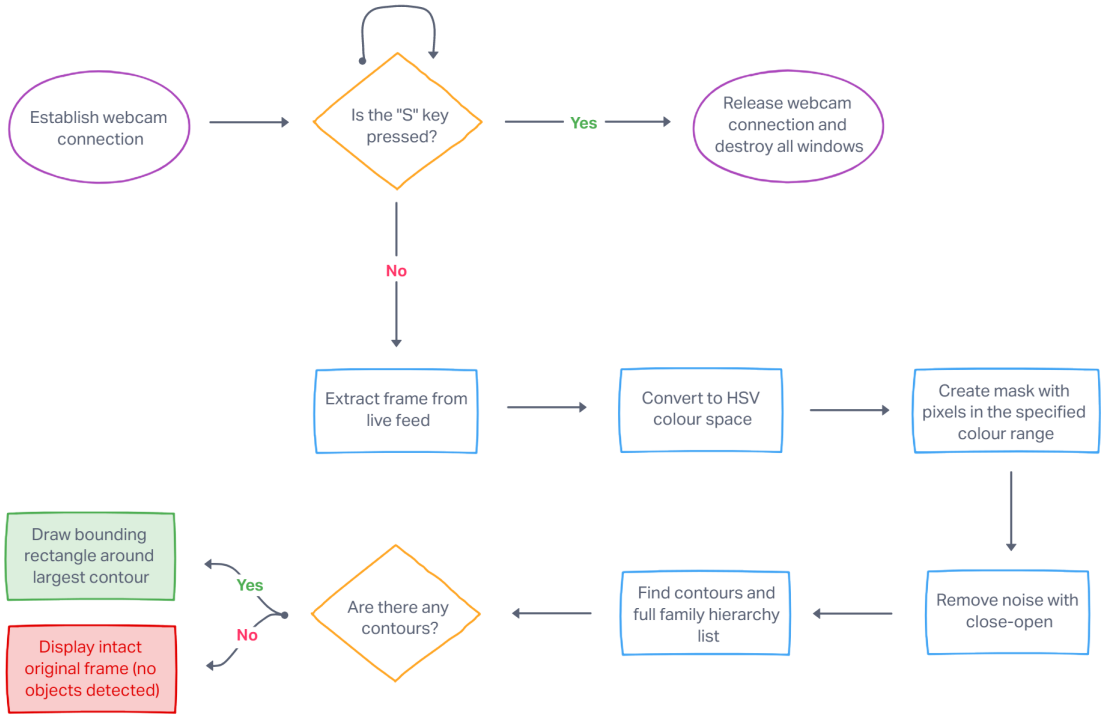


Figure 3.2: Colour-based object detection algorithm.

The Red, Green and Blue (RGB) colour space, widely used in digital image display and optical instruments, is not sensitive to human visual perception or statistical analysis [99]. When a colour pixel-value is adjusted in this colour space, intensities of red channel, green channel and blue channel are modified, meaning colour, intensity and saturation of a pixel are **not involved** in colour variations and so these are difficult to observe in complex colour environments or content [99]. The Hue, Saturation and Value (HSV) format is a non-linear transform from the RGB space that describes perceptual colour relationship more accurately, with **hue** denoting the property of colour (e.g. blue, green, red, etc.), **saturation** denoting its perceived intensity and **value** denoting its perceived brightness [99]. Therefore, converting the colours in the image from the standard RGB space to the HSV scale is

highly recommended when performing the task at hand in order to facilitate **feature extraction** by masking a specific colour out of the frame [98].

Next, the lower and upper limits of the desired colour range are specified and fed as arguments to the thresholding function, which will output a **binary mask** where all the values from the image source that lie within said range are set to 255 and the rest to 0. In order to obtain a better result, the noise and artifacts in the mask are reduced using the **opening** and **closing** operators. Both are morphological methods, image processing techniques based on the shape and form of objects, that consist of applying a structuring element to an input image such that the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbours [100]. While opening eliminates thin protrusions and thus the internal noise in the image, closing removes small holes, smoothens contours and fuses narrow breaks [101].

Finally, the contours in the mask are extracted to perform a **shape analysis** and distinguish between the possibly multiple segments in the scene that have been detected to present the same colour as the object of interest to be identified. Contours are defined as the lines joining all the points along the boundary of an image that possess the same intensity [102]. For this calculation, a *simple chain* contour approximation method, that removes all redundant points and thereby compresses the contour to minimise memory usage, and a *tree* contour retrieval mode, that creates a full family hierarchy list for all contours, are employed [102] [103]. The area of each contour is then obtained and compared in order to select the **largest** one for the computation of the coordinates and geometrical properties of a bounding rectangle that will be displayed in the original frame.

This algorithm's effectiveness was **tested** by setting the task of detecting a green water bottle placed amongst numerous objects (Figure 3.3 (a)). For this purpose, the colour range parameter was delimited by [45, 100, 50] and [75, 255, 255], the two HSV-expressed thresholds within which all the greens in the world lie [104]. Moreover, the kernel for the opening and closing operations was a 7×7 8-bit integer matrix which, as shown in Figure 3.3 (d), successfully removes the pixels in the mask that don't correspond to the target object [105]. All in all, the developed colour-based object detection algorithm provides encouraging results since the object of interest was **successfully distinguished** from its surroundings, as Figure 3.3 (f) exhibits. Nevertheless, the entire object's area is not captured by the obtained contour, displayed in Figure 3.3 (e), which may result **problematic** when calculating the object's coordinates as this variable will be used to determine its location in terms of depth within the field of view.

Further testing under different environmental conditions uncovered additional limitations. As it can be observed in Figure 3.4, different **lighting** settings heavily influence this algorithm's outcome. In the case of medium lighting, the green water bottle is correctly detected and its entire area is covered by the bounding rectangle.

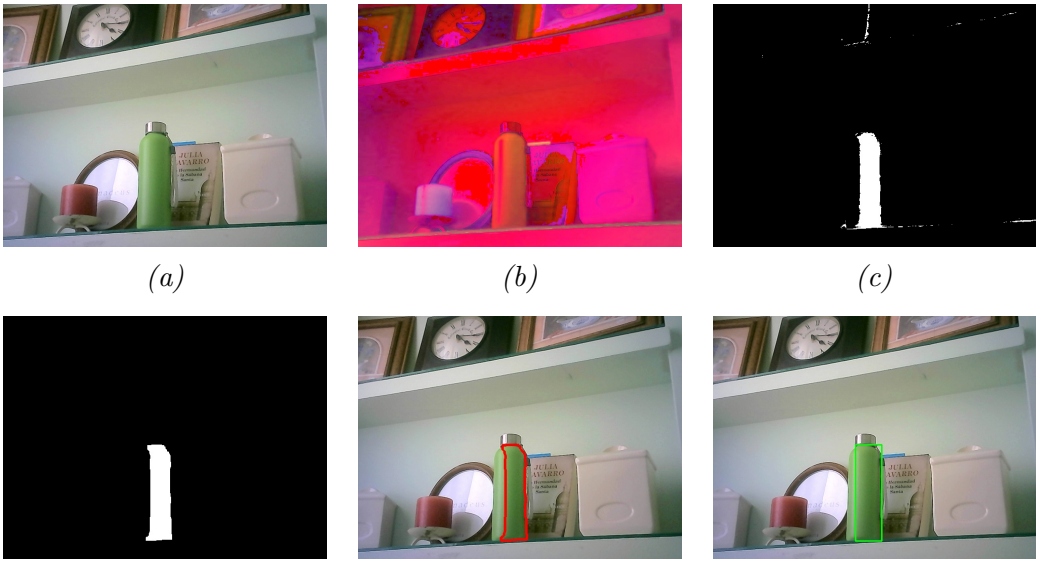


Figure 3.3: Colour-based object detection test. *(a)* Original frame *(b)* HSV conversion *(c)* Colour-thresholding mask *(d)* Close-open noise removal *(e)* Contour extraction *(f)* Bounding rectangle over detected object.

However, when the lighting is slightly increased or decreased with respect to this scenario, the water bottle is no longer identified; instead, either another object in the scene is mistaken for said item or none are distinguished. This suggests that the detection quality of manual thresholding algorithms is **unpredictable** under varying environmental conditions and so the coordinates and dimensions of the retrieved contours can sustain significant changes.

The main downside of the simple thresholding technique of having to manually determine the threshold in advance can be overcome through **automatic thresholding** methods such as the Otsu method, mixture models or other histogram shape-based procedures. Nevertheless, these are only useful when the image's histogram presents clear peaks corresponding to each object and when the number of objects in the scene is known in advance, which is not always the case in computer vision applications [106]. Another alternative is **adaptive thresholding**, an approach that can be employed even when the image does not contain distinct peaks and in which, unlike global thresholding, different threshold values are computed for each fractional region so as to reduce the influence of illumination gradients, another problem of the simple algorithm deployed [107]. However, its use is restricted to separating desirable foreground image objects from background pixels, and is not able to distinguish between different foreground objects as is requested of this navigation system.

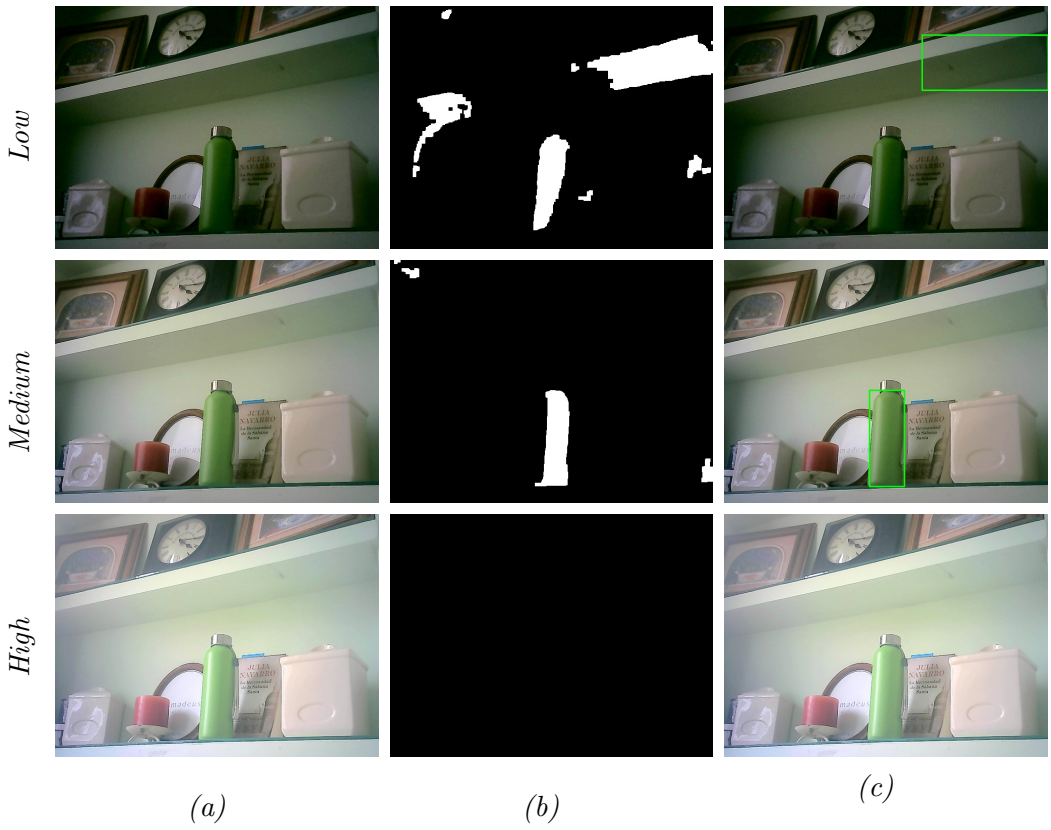


Figure 3.4: Colour-based object detection test under different lighting conditions (*low*, *medium* and *high*). *(a)* Original frame *(b)* Colour-thresholding mask after noise removal *(c)* Bounding rectangle over detected object.

3.4 Edge-based object detection

The problem of image edge detection is based on a discontinuity search for abrupt changes in pixel intensity values and plays an important role in computer vision systems [108]. Edge detection is the process of finding edges in an image, revealing **structural information** which could correspond to an object's boundaries, shadowing or lighting conditions or boundaries of "parts" within an object [109]. Using these outlines, contours can then be applied to extract the actual objects from the regions or quantify the shapes so that they can be later identified [110]. In order to implement this practice in the object detection task of this Master's thesis, the algorithm presented in Figure 3.5 has been deployed.

Edge detection requires the computation of **image gradients**, formally defined as directional changes in image intensity [110]. This task will be performed by means of the ***Canny* edge detector**, a multi-step algorithm introduced by John F. Canny in 1986 and regarded by many as the standard for edge detection [111]. It has been chosen because of its ability to produce single pixel thick, continuous edges, to detect strong and weak edges and its insusceptibility to noise interference [112].

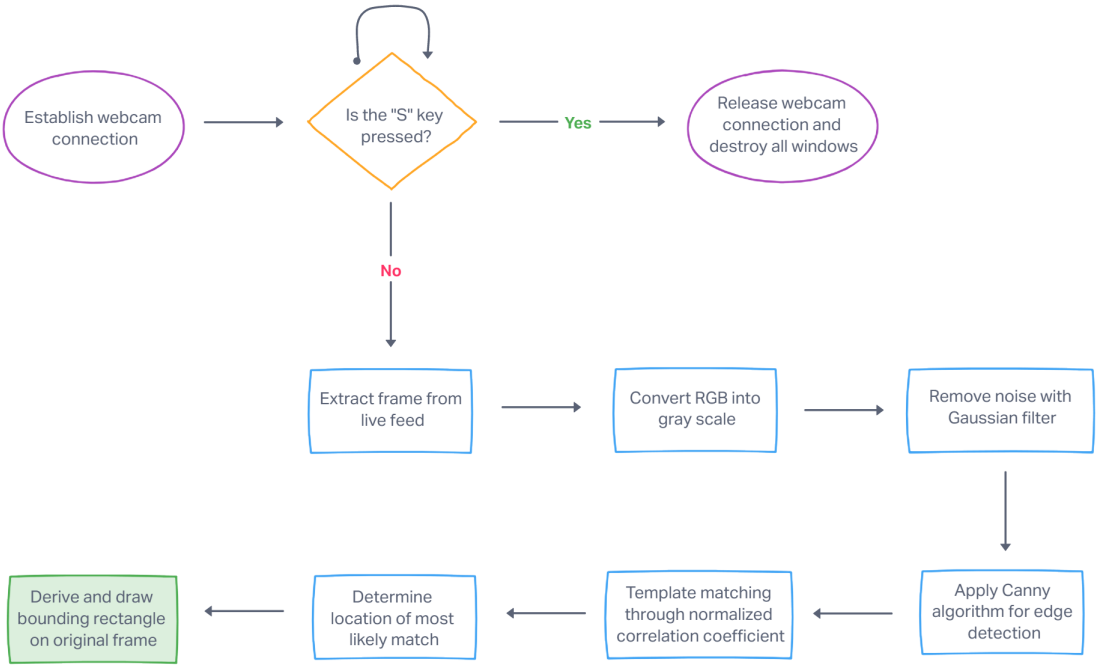


Figure 3.5: Edge-based object detection algorithm.

The frames must be first converted from the RGB colour space to gray scale to, not only reduce the computational cost associated to the subsequent processing operations, but also ensure less **noise** during the edge detection process [113]. Then, **Gaussian smoothing** is applied for a reduction in the amount of high-frequency content as another means of noise removal, as the gradient magnitude is quite susceptible to this phenomenon [110]. This filter iterates through every pixel in the image with a 5 x 5 kernel conducting a weighted average according to the Gaussian distribution, thus conferring greater importance to central pixels in the mask [113].

The image is then fed to the *Canny* algorithm. This multi-step process also includes Gaussian filtering, followed by the calculation of the gradient magnitude and orientation using the ***Sobel*** method (equivalent to first derivative estimation), non-maxima suppression as an edge-thinning process and, finally, hysteresis thresholding to remove regions that technically aren't edges [110]. Therefore, the only parameters demanded by this module are the maximum and minimum limits for the **hysteresis thresholding** phase, which were determined by comparing the results obtained with three different cases: a *wide* threshold, a *mid-range* threshold and a *tight* threshold, finally keeping the intermediate scenario [110].

The edge maps provided by the *Canny* algorithm enable the distinction between foreground objects and the background. Therefore, an **additional criterion** is required to distinguish the object of interest from the rest once the edge features have been identified. In this case, a **template matching** procedure has been pursued as suggested by [112]. This reasoning has been adapted to the available tools by employing a method that slides the template image over the input image, calculating

the correlation response between the template and each patch of the processed frame [114]. Finally, the location of the **maximum correlation** value is found and, based on its coordinates, a bounding rectangle is constructed and drawn on the original frame to highlight the object detected.

As with the colour-based object detection technique, this algorithm was tested by setting the task of detecting a green water bottle placed amongst numerous objects. The results obtained are presented in Figure 3.6. The output from the *Canny* edge detector (Figure 3.6 (c)), as previously hypothesised, successfully **isolates** the foreground objects from the background, thus simplifying the template matching procedure. The template was obtained by capturing a front-view image of the water bottle, shown in Figure 3.6 (d), applying the *Canny* algorithm and then resizing the output to a size similar to that of the object in the original frame. The final template, exhibited in Figure 3.6 (e), is employed in the template matching process yielding the normalised correlation coefficient matrix displayed in Figure 3.6 (f), whose maximum values are illustrated in white and are located in the bottom middle section where the object is located. Hence, the object is **correctly outlined** in Figure 3.6 (g).

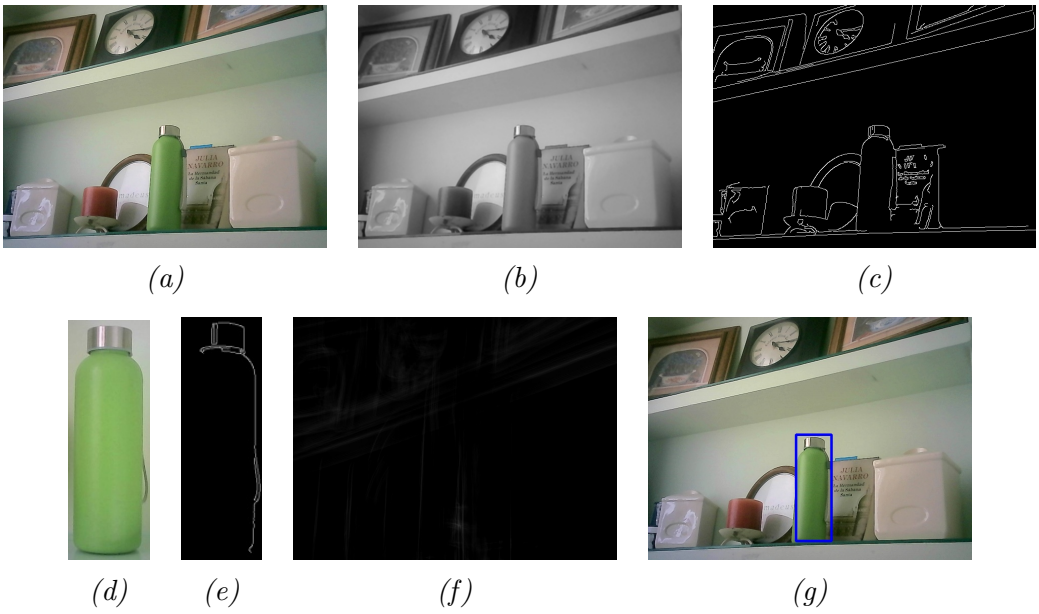


Figure 3.6: Edge-based object detection test. *(a)* Original frame *(b)* Gray scale conversion and Gaussian smoothening *(c)* Edge detection by Canny algorithm *(d)* Raw template *(e)* Canny-processed template *(f)* Correlation matrix *(g)* Bounding rectangle over detected object.

With the algorithm's effectiveness proven under the standard environmental conditions, it's dependence on the scene's lighting was studied to analyse the importance of this parameter in this technique's performance and compare its **robustness** with that of the previous algorithm. In order to do so, it was fed the

same *low*, *medium* and *high* lighting frames used in the previous light-dependence test, providing the results exhibited in Figure 3.7. As it can be seen, even though the *Canny* algorithm's outputs vary slightly depending on the lighting, the edge-based object detection is not affected by this variable and so it can be considered to be **independent** of this parameter.

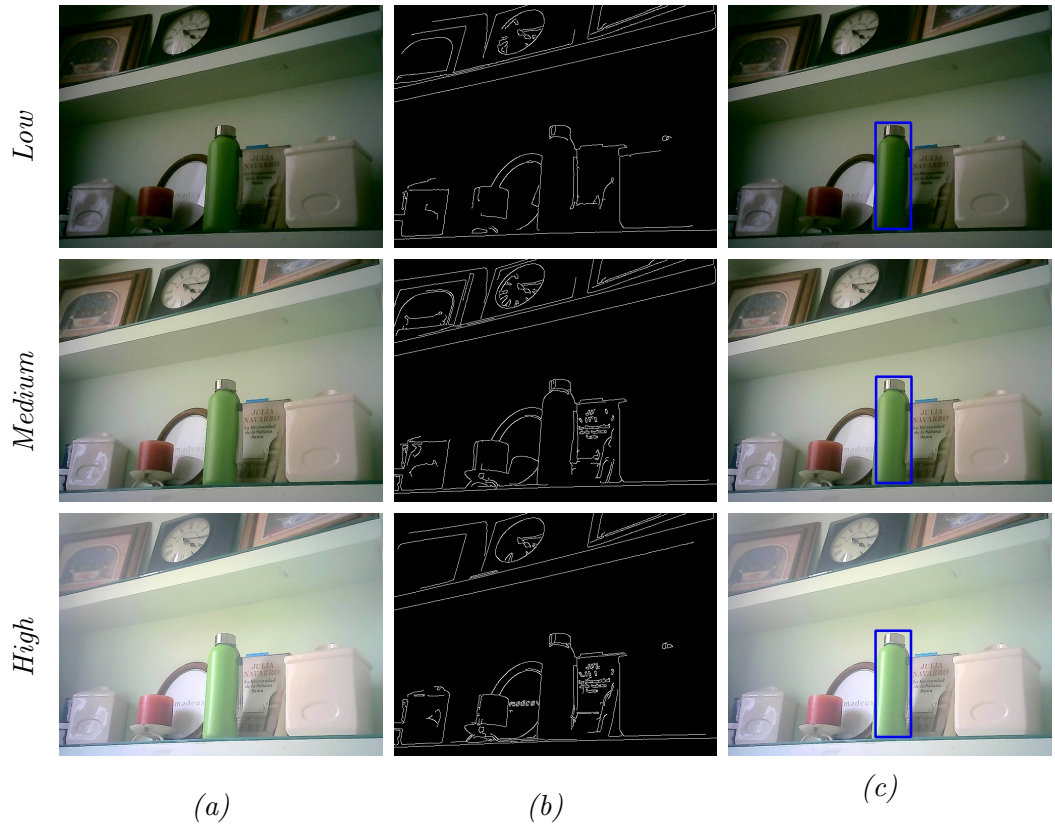


Figure 3.7: Edge-based object detection test under different lighting conditions (*low*, *medium* and *high*). *(a)* Original frame *(b)* Edge detection by Canny algorithm *(c)* Bounding rectangle over detected object.

Nevertheless, the exploitation of the template matching procedure limits this algorithm's independence from variability in input images. For instance, the **scale** of the template must be similar to that of the object in the frame for there to be a sufficiently high correlation between the two. A study was carried out to determine this algorithm's tolerance to differences between template and object sizes by varying the template's size until the water bottle was no longer recognised. The results indicate that an adequate recognition requires a template size **at most** 20% over or 15% under the object's size in the input image, which may result in an incorrect detection as the robot approaches or moves away from the object of interest.

Although robustness to scaling variations can be improved employing a **multi-scale** approach in which the input image is repeatedly re-scaled in search for the largest correlation coefficient, template matching can similarly fail due to changes in

rotation, viewing or non-affine transformations [115]. Even if the template employed in the above, successful tests presents a different viewing angle to that of the water bottle in the input images, as it can be observed in Figure 3.6 (d) and Figure 3.6 (a), a similar behaviour in terms of this factor as with the scale can be expected, with changes in viewing angle greater than a certain proportion causing inevitable **detection errors**. Therefore, template-matching is only applicable when templates are fairly rigid and well-defined via an edge map and when variations in translation and scaling, at most, are expected [115].

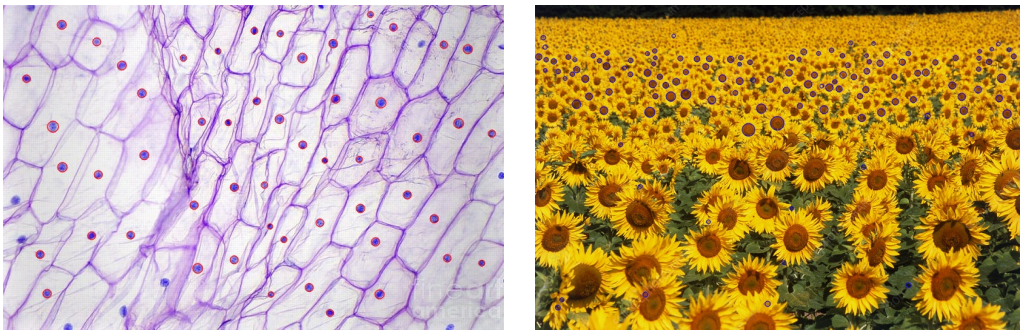
3.5 Region-based object detection

A **region** can be described as a group of connected pixels exhibiting similar properties such as intensity or colour [116]. This type of segmentation method looks for similarities between adjacent pixels, grouping together into unique regions those that possess similar attributes according to **predefined rules** [116] [117]. It tends to work well for difficult imagery, as it is adaptive and less susceptible to the effects of partial occlusion, adjacency, noise and ambiguous boundaries [117]. Region-based techniques are further classified into 2 types based on the approaches they follow. **Region growing** methods start with some pixel as the seed and add those adjacent pixels that abide by the predefined similarity rules to its region [116]; in **region splitting and merging** methods, the whole image is first taken as a single region that is repetitively subdivided as long as the generated sub-regions don't follow the pre-established rules [116].

Although these segmentation methods are not commonly employed for the purpose of object detection, *OpenCV* does provide a simple **binary large object (BLOB)** detector, where BLOBs are the equivalent of regions: groups of connected pixels that share some common property [118]. This algorithm converts the input image into multiple binary images by applying several thresholds, extracting connected components from every binary image through contour identification, and calculating the coordinates of their centres to then group close centres from different images together [119]. From these groups, the final centre coordinates of each detected BLOB and their corresponding radii are returned as locations and sizes of keypoints [119]. The final BLOBs can then be **filtered** according to different criteria including colour, size, circularity, convexity or inertia ratio in order to extract the region or objected that is searched [118].

As opposed to the aforementioned classical segmentation algorithms, this region-based object detection technique will not be discussed in detail in this chapter since the results yielded by this procedure were far from satisfactory. As shown in Figure 3.8, the BLOB detector's output was found to be **heavily dependent** upon the morphology, both in shape and size, of the regions of interest. Despite yielding fairly accurate detections for small elements, for instance, the cell nuclei displayed in Figure 3.8 (a), when exposed to objects of a larger scale the algorithm was **uncapable** of identification regardless of the filtering criteria specified, even when the rest of the

attributes, such as colour and shape, were maintained. This can be corroborated in Figure 3.8 (b), where similar sunflowers are captured at different distances from the camera and only the ones at the back, of smaller apparent size in the image, are recognised by the detector. A similar effect was observed in terms of shape, whereby **non-circular** objects were extremely difficult to discern. Instead, region proposal will be examined in the next chapter as its use is essential to many AI-based object detection algorithms.



(a) *(b)*

Figure 3.8: Region-based object detection *(a)* Accurate detection for small elements *(b)* Inconsistent detection for same object depending on its size in the image.

3.6 Limitations

The classical image segmentation methods reviewed in this section for the purpose of object detection have proven that one of the major drawbacks of simple computer vision algorithms is **parametrisation**: many techniques require setting parameters or initial conditions for each situation [113]. Frequently, these parameters are exclusive to a specific lighting or perspective, as is the case with the colour-based and edge-based object detection algorithms presented, respectively. This also entails certain limitations of particular importance in computer vision approaches, such as **previous knowledge** requirements about the object to be detected and its surroundings (e.g. colour or lighting temperature). As a result, it complicates finding a **unique** solution for different situations and therefore, something that works correctly under certain experimental conditions does not necessarily work under others.

All in all, the constraints endured by classical segmentation methods originate from the two common underlying **assumptions** they are built upon: "*the object of interest is uniform and homogeneous with respect to some characteristic*" and "*adjacent regions differ significantly*" [94]. These are rarely met in real-world applications, deriving in their inability to **adapt** to real-world changes that can be caused by variations in the object itself (e.g. different colour or texture) or in the environmental factors, most importantly shadow and highlight bands which cause non-uniform changes in the appearance of objects, violating the homogeneity assumption [94].

Chapter 4

Artificial Intelligence Based Recognition

Artificial vision is contained within the field of AI as it makes use of its different algorithms, techniques and methods in order to achieve the processing of information contained in digital images [120]. These techniques generally fall into either **ML**-based approaches, where it becomes necessary to first define the features a selected classifier will then use to sort the images, or **DL**-based approaches, which are capable of carrying out the desired task without having to specifically define features [121]. The state-of-the-art object detection algorithms typically leverage ML or DL methodologies through two main types of approaches: **one-stage** methods, which prioritise inference speed, and **two-stage** methods, which prioritise detection accuracy [89]. Both these categories will be studied and compared in the present chapter of this Master's Thesis.

Two-stage detectors first filter out the regions that have a high probability of containing an object from the entire image, phase which is known as **region proposal**, and then feed the candidate bounding boxes to a classifier which extracts features in order to assign each box's corresponding classification score [122]. In contrast, a one-stage detector predicts bounding boxes in a **single step** without using region proposals, leveraging a grid box and anchors to localise the region of detection in the image and constraint the shape of the object [123]. Though one-stage detectors are more **efficient** due to their straightforward architectures, two-stage detectors still take the lead in **accuracy** [123]. This is mainly due to the fact that, by sampling a sparse set of region proposals, two-stage detectors filter out most of the negative proposals and can afford the extraction of richer features as only a small number of proposals are processed [122].

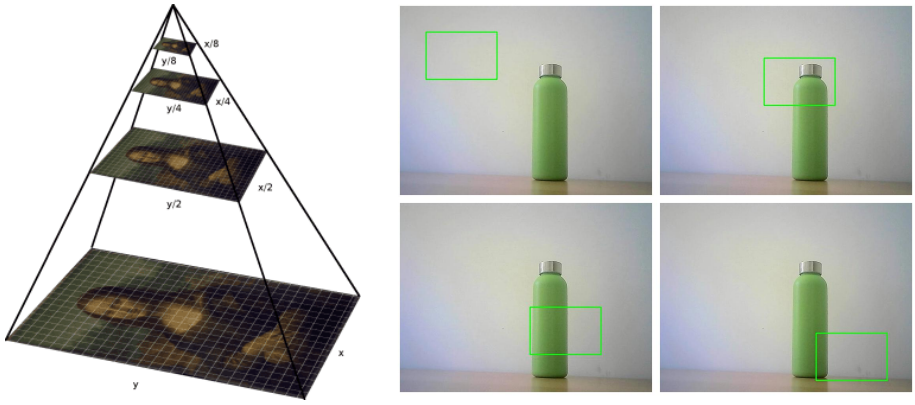
4.1 Image classifiers into object detectors

When performing image classification, an input image is given a **class label**, that is meant to characterise the contents of the entire image or, at least, the most dominant, visible contents of the image, and the probability associated with the class label prediction [124]. Object detection, on the other hand, is not only capable of

inferring what is in the image (i.e., class label), but also of identifying **where** in the image the object is located by means of bounding box coordinates [124]. Object detection networks are available but are more complex, more involved and take multiple orders of magnitude and more effort to implement compared to traditional image classification [124]. Fortunately, there are several computer vision techniques that can be leveraged to **convert** any image classifier into an object detector by means of a two-stage approach.

4.1.1 Image pyramid and sliding window approach

Operating with an image of constant size is the usual practice in image processing; however, when searching for an object, the size at which said target will be present in the image is **unknown** and so a set of the same image with different resolutions must be created so that the object can be searched for in all of them and found regardless of its scale [125]. When arranged into a stack where the highest resolution image, at its original size, is located at the bottom and the lowest resolution image at the top, as shown in Figure 4.1 (a), these sets of images resemble pyramids, as suggested by their name [125]. At each subsequent layer, the image is progressively **sub-sampled** and optionally smoothed via Gaussian blurring until some stopping criterion is met, normally when a minimum size has been reached [124].



(a) *(b)*

Figure 4.1: Traditional approach to converting image classifiers into object detectors *(a)* Image pyramid [124] *(b)* Sliding window.

These multi-scale image representations must then be split into **regions**, each of which will be fed to the classifier in order to decide whether the object of interest is present in any of them. These regions are generated by means of a **sliding window**, a rectangular region of fixed width and height that slides from *left-to-right* and *top-to-bottom* within an image, extracting in each step the region of interest (ROI) of the image captured within its area [126]. In combination with image pyramids, sliding windows enable the localisation of objects at different **locations** and multiple **scales** of the input image.

4.1.2 Selective search

Region proposal algorithms constitute an alternative to the image pyramid and sliding window approach that avoids the high **computational cost** associated to looping over each image pyramid layer and inspecting every location in the image with a sliding window, and that also reduces the **sensitivity** to parameter choices such as pyramid scale and sliding window size, that can lead to significant variations in the yielded results [127]. The general idea behind these algorithms is to inspect the image and attempt to find regions that likely contain an object accurately and in a fast and efficient way, so that these "*candidate proposals*" can then be fed to a downstream classifier which labels them, thus completing the object detection framework [127].

Selective search is an automatic region proposal algorithm that operates by over-segmenting an image into superpixels which are then merged together in a hierarchical fashion based on similarity measures in order to find regions that could contain an object [128]. This procedure is exhibited by Figure 4.2, where the bottom layer of the pyramid is the original over-segmentation generated by means of a superpixel algorithm, regions are joined together in the middle layer and, eventually, the final set of proposals, displayed in the top layer, are formed [127]. The five key similarity measures for merging are: **colour**, **texture**, **size**, **shape** and a final **meta-similarity** measure that acts as a linear combination of the aforementioned properties [128].

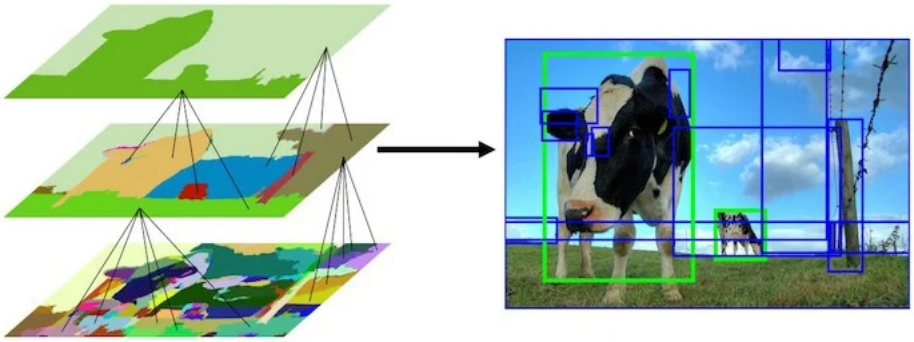
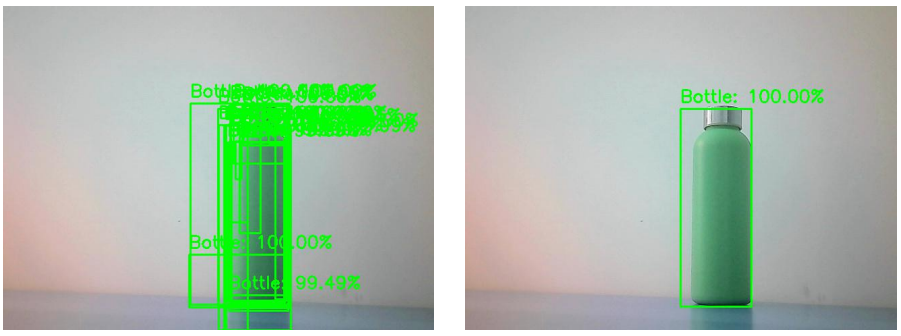


Figure 4.2: Selective search algorithm for automatic region proposal [127].

4.1.3 Non-maximum suppression (NMS)

In the object detection pipeline, once the candidate regions for the object of interest have been identified, for instance, with either of the approaches detailed above, they are then fed to a **classifier** which assigns foreground/background scores depending on the features computed in each region [129]. Neighbouring windows normally present similar scores to some extent and can therefore all be considered as candidate regions if they surpass the threshold score established for detection, which leads to **hundreds** of proposals and, thus, a large number of bounding boxes surrounding the image, as can be observed in Figure 4.3 (a). While each detection may in fact be valid, the

classifier must not report back multiple objects in the image when there is clearly just one. This problem gave rise to NMS, a technique which **filters** the proposals based on some criteria [129]. The result of its application is demonstrated by Figure 4.3 (b).



(a) *(b)*

Figure 4.3: NMS *(a)* Original classifier output *(b)* Output after NMS.

4.2 ML vs DL classification

The key challenge in creating ML classifiers is achieving **robustness** to variations in illumination, pose and occlusions in the image [130]. In this way, rather than being trained on the pixel-based representation of the image, an intermediate representation, commonly known as **feature vector** or **descriptor**, is employed [130] [131]. This simplification only extracts the image information that is useful for classification and that is invariant to small changes in illumination and occlusions, reducing an image of size *width* x *height* x *3 (channels)* into an array of length n [130] [131]. Whilst earlier ML classifiers require manually deciding which characteristics of the image are most important for the task at hand, DL approaches perform this selection **automatically**, taking the task of feature extraction out of the developer's hands [132]. Both of these approaches are contrasted and presented below for the purpose of image classification.

4.2.1 Feature descriptors for ML

The main feature descriptors employed in the field of computer vision are: the gray level co-ocurrence matrix (GLCM), the local binary pattern (LBP), the scale-invariant feature transform (SIFT) and the histogram of oriented gradients (HOG) [121]. Once these are extracted from an image, they must be paired with an **AI algorithm**, such as those presented in Section 2.2.3 and Section 2.2.4, in order to carry out the classification problem.

4.2.1.1 GLCM

The GLCM function characterises the texture of an image by means of calculating how often a **pair** of pixels with specified values and spatial relationship occurs [133].

In this way, a square matrix, whose size depends on the image's range of intensities (e.g. 256 x 256 for an 8-bit channel), is constructed in which each entry *GLCM[i,j]* holds the count of the number of times the corresponding pair of intensities *(i, j)* **appears** in the image in the defined direction and distance [134]. Once the GLCM is calculated, **texture properties** can be extracted from the matrix, such as correlation, energy, homogeneity, contrast, or dissimilarity, to represent the textures in the image [134]. To construct the GLCMs, a series of parameters are required which must be stipulated through rigorous analysis for the extraction of the most important textural information with lesser number of correlated features [133].

4.2.1.2 LBP

LBPs are very descriptive and efficient yet computationally simple grayscale texture operators that have become a popular approach specially in applications involving challenging real-time settings such as face recognition or visual inspection [135]. Rather than computing a global representation of texture as is the case with the GLCM, a **local** representation is constructed by comparing each pixel with its surrounding neighbourhood of pixels [136]. For every pixel in the grayscale image, a LBP value is calculated by **thresholding** it against its selected neighborhood of size r; each neighbour is given a value of 0 or 1 depending on whether its intensity is smaller or greater-than-or-equal to that of the centre pixel being evaluated [136]. These are then converted to decimal to obtain a single value per pixel and built into a **histogram** that describes the entire image, yielding its feature vector [135].

4.2.1.3 HOG

The HOG descriptor focuses on the **structure** or the shape of an object, counting occurrences of gradient orientation in the localised portion of an image [137]. To calculate a HOG descriptor, the horizontal and vertical gradients must be first calculated by filtering the image using kernels such as the Sobel operators; this removes a considerable amount of non-essential information, highlighting **outlines** (i.e. sharp changes in intensity) in their respective directions [131]. At every pixel, the gradient has a magnitude and a direction, but rather than depicting individual gradients, the image is subdivided into fixed-sized **cells** and a histogram is computed with both variables for each patch in order to obtain a more compact representation that is also much less sensitive to noise [131]. The final HOG feature vector is obtained by concatenating all the histograms from every block in the image once these are subjected to normalisation in order to reduce the influence of lighting variations [131]. All in all, the HOG is better than any edge descriptor as it uses magnitude as well as **angle** of the gradient to compute the features [137].

4.2.1.4 SIFT

SIFT is a feature detection algorithm which enables the location of local features in an image, commonly known as "**keypoints**", whose major advantage is that they are not affected by the size or orientation of the image [138]. This is achieved by constructing a scale space, a collection of images with different scales generated from

the same one, which are subjected to feature **enhancement** by means of the difference of Gaussian (DoG) technique that acts as an edge detector [139]. Next, the important keypoints are found by calculating the local maxima and minima and then removing low contrast keypoints and those that lie very close to the edge [139]. At this stage, a set of **scale-invariant**, stable keypoints have been selected which are finally assigned an orientation so that they are invariant to rotation by computing a histogram for magnitude and orientation of each pixel in the image [138].

In the world of natural language processing (NLP), multiple documents can be compared by counting the occurrences of each word in the corpus of all words, thus converting each document into a histogram of word counts that can be used as a feature for ML [140]. Every picture can be thought of as a document of "**visual words**", or SIFT descriptors, with each "word" representing a part or feature of the image such as an eye or a finger, and so the *bag of words* model can be extended to classify images instead of text documents [140]. Descriptors representing the same real-world feature must be grouped together as these may present variations over different images; this gathering can be performed mathematically with a **clustering** algorithm, where the descriptors are grouped into K different codewords [140].

4.2.2 CNNs for DL

Convolutional neural networks (CNNs) are frequently used for the task of image classification as this term is used to describe an architecture for applying ANNs to two-dimensional arrays [132] [141]. CNNs may be conceptualised as a system of **connected feature detectors** with non-linear activations that is able to take raw data, without the need for an initial separate preprocessing or feature extraction stage, and perform both the feature extraction and classification tasks naturally within a single framework [141]. Neurones that are located earlier in the network are responsible for examining small windows of pixels and detecting simple, **basic** features such as edges and corners [132]. These outputs are then fed into neurones in the intermediate layers, which learn to recognise particular spatial combinations of previous features at progressively larger spatial scales, generating "**patterns of patterns**" in a hierarchical manner [132] [141]. These are then used to make a final judgment about whether the image contains the desired object [132].

4.3 End-to-end object detection

The classification algorithms introduced in the previous section can all be exploited as object detectors by applying two-stage techniques such as those detailed in Section 4.1. However, this type of network is not **end-to-end trainable**. Precisely, the reason why DL-based object detectors such as Faster R-CNN or Single Shot MultiBox Detectors (SSDs) perform so well is because they are end-to-end trainable as the region proposal task is carried out **internally** [124]. This means that any error in bounding box predictions can be made more accurate through back propagation and updating the weights of the network [124].

4.3.1 YOLO

Amongst single-stage object detectors, the YOLO algorithm, which is an abbreviation for the term "*You Only Look Once*", has excelled due to its speed, accuracy and learning capabilities which enable **real-time** detection applications [142]. Other approaches such as the aforemetioned Faster R-CNNs or SSDs have solved the challenges of data limitation and modelling in object detection, but do not possess YOLO's ability to detect multiple objects in an image in a **single algorithm run**, yielding a superior performance [142]. It works by dividing the image into N grids, each having an equal dimensional region of S x S and each being responsible for the detection and localisation of the object it contains [143]. Correspondingly, these grids predict bounding box coordinates relative to their cell coordinates, along with the object label and probability of the object being present in the cell [143]. This process greatly **lowers** the computation as both detection and recognition are handled by cells in the image [143].

4.4 Implementation

4.4.1 Dataset generation

In ML, datasets are split into two subsets. The first subset is known as the **training data** and is the portion of the actual dataset that is fed into the ML model to discover and learn patterns [144]. The other subset is known as the **testing data** and it is unseen data that is used to evaluate the model's performance and/or progress the algorithm's training by adjusting or optimising it for improved results [144]. In this case, the entire dataset was constructed by manually acquiring *640* x *480* resolution images with the USB camera displayed in Figure 3.1. The chosen target for detection was the same water bottle employed for the implementation of the classical segmentation algorithms, capturing said object under **different environmental conditions** such as natural and artificial lighting, varying lighting brightness and temperature (at different times of day), diverse backgrounds and numerous distances from the object and viewing angles. The images were then annotated by means of the *LabelImg* tool to register the object's ground-truth coordinates within each image.

4.4.2 Training

The training dataset was comprised of 74 images of the object in as many different conditions as possible, as shown in Figure 4.4 (a), which contributed towards the aim of subjecting the models to a considerable amount of **variability** in order to optimise their robustness. The ML classifiers destined for the purpose of image classification require 2 types of data: **positive** images, containing the object that is to be detected, and **negative** images, that don't contain the object at all. Therefore, the original dataset had to be transformed to obtain such an arrangement by using the previously, manually-specified coordinates of each object in the image in order to **crop** the ROIs containing the object, which are considered positive samples, and the rest as negative samples.

Taking advantage of this necessary step, a **data augmentation** technique was introduced so as to increase the training dataset's size. Selective search was employed to propose regions where there could be an object and only those which presented an **intersection over union** (IoU) greater than 70% with the corresponding ground-truth bounding box were introduced into the positive set. The IoU is an evaluation metric used to measure the accuracy of an object detector on a particular dataset as it represents the ratio of the area of overlap between a predicted bounding box and its respective ground-truth bounding box with respect to the area of union, that is, the area encompassed by both [145]. The resulting images were then resized to maintain a fixed, constant size and reduce their computational load, taking into consideration the preservation of their average aspect ratio. With a maximum number of 30 positive and 10 negative images to be extracted from each original image, a total of 1.374 and 781 positive and negative samples were generated, respectively. A subsample of the final training dataset for classification is displayed in Figure 4.4 (b).

(a) *(b)*

Figure 4.4: Training dataset *(a)* Original layout *(b)* Final classification dataset after data augmentation: positive (*top*) and negative (*bottom*) samples.

Firstly, the training of all the supervised learning classifiers presented in Section 2.2.3 was carried out based on the manual extraction of the feature descriptors previously introduced in this Chapter. Each classifier was trained with each type of descriptor with the aim of discovering the **best performing pairings**. To do so, the training procedure was standardised and, after various iterations of model evaluation, the optimal approach for each descriptor was attained. The GLCM was extracted in four different directions: $\theta = \{0°, 45°, 90°, 135°\}$ and with a distance of 1 *px*, the LBP was computed with 24 circularly symmetric neighbour set points and a radius of *8 px*, and the HOG descriptor was acquired with *5* x *7 px* cells and *10* x *21 px* blocks. The best results were found to be obtained with a simple preprocessing consisting of grayscale conversion with the exception of the SIFT descriptor, whose performance was discovered to be enhanced through histogram normalisation and equalisation, along with the removal of the training image resizing process which resulted in lower image resolution and thus complicated the identification of keypoints.

With all features extracted from the training set of images, the collected data could then be fed to the CNN, RF and SVM classifiers. In the case of the SIFT feature, an additional step of training a *Kmeans* classifier was also required to cluster the extracted keypoints into 5 different codewords that would enable the classification of images, as in NLP. The performance of these ML algorithms can be very **sensitive** to how cost, kernel or other types of parameters are set [74]. As a result, extensive cross validation was conducted in order to determine their optimal parameter settings, process which is commonly referred to as **model selection** [74]. This hyperparameter tuning was performed by means of the **grid search** technique, which enables exhaustively testing all possible hyperparameter configurations the developer is interested in optimising [146]. The grid was constructed with different weighting functions and algorithms to compute the nearest neighbours, with varying numbers of tree estimators and minimum number of samples required to split nodes, and with varying kernels and tolerances, among other factors, for the KNN, RF and SVM classifiers, respectively. The optimal parameter settings were determined using a **k-fold cross-validation** process of 10 folds and 3 repetitions.

Finally, the remaining DL models were trained. In the case of the CNN for classification, a *MobileNetV2* architecture was selected and an optimised preprocessing technique for this particular architecture, provided by *tensorflow*, was employed. An **image generator** was constructed for data augmentation on the fly by performing operations such as rotations, zoom, horizontal flips or width shifts, and callbacks including **early stopping**, to avoid overfitting, **reduced learning rate on plateau**, to enhance model performance, and **model checkpoints**, to only save the best model during training, were implemented. The quantity that was chosen as the metric to be monitored in order to regulate the behaviour of these callbacks was the **validation loss**, which required splitting the training set with an 80-20 split to extract a validation dataset and selecting a loss function, binary cross-entropy. In the case of YOLO, the training images were those originally captured, before the data augmentation procedure (Figure 4.4 (a)), and online imagespace and colourspace augmentations were also applied to create 3 random images from each input image [147]. Following official guidelines and recommendations on training with small custom datasets, **pretrained** YOLOv5 weights were utilised [148].

4.4.3 Testing

The different combinations of supervised learning classifiers and feature descriptors were first evaluated on the basis of image **classification**. To do so, a 20% subset of the aforementioned training dataset obtained through data augmentation was employed, which was naturally excluded from the training process. In this way, a test dataset completely independent from both the training and testing stages of image classification would be later employed for the evaluation and comparison of these classifiers as object detectors with respect to the YOLO detector. In this first scenario, the chosen evaluation **metrics** to assess the models' performance were: accuracy, sensitivity, specificity and ROC AUC score, all of which are shown in Table 4.1.

		Classifer			
		KNN	*RF*	*SVM*	*CNN*
Feature	*GLCM*	0.853 0.718 0.931 0.875	0.923 0.795 0.996 0.963	0.637 0.0 1.0 0.647	0.98 ± 0.02
	LBP	0.935 0.865 0.974 0.969	0.944 0.872 0.985 0.986	0.529 0.981 0.273 0.849	0.95 ± 0.06
	HOG	0.881 0.673 1.0 0.899	0.951 0.897 0.982 0.995	0.965 0.962 0.967 0.997	0.994 ± 0.006
	SIFT	0.853 0.718 0.931 0.865	0.747 0.75 0.745 0.819	0.639 0.004 1.0 0.629	0.9983 ± 0.0006

Table 4.1: Evaluation metrics of ML and DL classifiers. Colour code: accuracy, sensitivity, specificity and ROC AUC score.

Accuracy is the ratio of correct predictions to the total number of input samples [149]. For **sensitivity** and **specificity**, the confusion matrix was constructed, with the first being the proportion of positive data points that are correctly considered as positives and the second the proportion of negative data points that are correctly considered as negative [149]. Finally, the area under the receiver operating characteristic curve (**AUC ROC**), one of the most widely used metrics for binary classification evaluation, is the probability that the classifier will rank a randomly chosen positive example higher than a randomly chosen negative example [149].

The testing procedure was performed on **three** trained models of each feature-classifier combination, with the mean values for each evaluation metric displayed in Table 4.1. As it can be observed, the only model that presented performance **variations** across different training instances was the CNN, where the standard deviation is included as well as the mean value, due to the stochastic nature of its training process. Overall, the results obtained are encouraging, with most classifiers presenting reasonably good behaviours. These outcomes reveal that the same data, in this case the various features extracted from the images, yield **different results** depending on the ML model employed. As expected, the CNN exhibits excellent results across all 4 metrics, particularly in terms of accuracy and ROC AUC score, arguably the most important variable, where it assumes the leading position amongst all models. Interestingly, however, the other models do not fall far behind, specially the well known HOG-SVM pairing, that also performs very well across all 4 variables.

This suggests that manual feature extraction is still a viable option and may even prove to be a **better** solution in some cases than DL considering the computational resources involved. Nevertheless, finding the appropriate features to extract, the best model to employ and the optimal parameter settings is **crucial** to the development of a high-quality performance algorithm, as this investigation demonstrates.

These classifiers were then converted into object detectors using the aforementioned image pyramid and sliding window approach and the selective search methodology. For the first scenario, a custom function was developed to loop over each image, generating increasingly downsized copies according to a **scale factor** received as a parameter, finally set at 10% for each iteration, until a minimum size was reached, corresponding to the object's smallest expected dimensions. Moreover, for each layer of the pyramid, several **ROIs** were extracted in accordance with the window size and window step size specified by the user, finally fixed at *90* x *280 px* and *8 px*, respectively. These ROIs were then fed to the classifying models after the adequate preprocessing and feature extraction techniques had been applied and their coordinates in terms of the original image were stored in order to later possess the ability to **localise** positive detections. On the other hand, the selective search strategy was carried out with *OpenCV*'s utilities, establishing a maximum number of 200 proposals from each image on which to perform inference.

To test and compare the image classifiers and the YOLO detector, an independent dataset was generated by acquiring 25 images of the water bottle in environments which were not included in the training set, again varying factors such as lighting conditions to introduce high variability and, therefore, extract **reliable** conclusions. These images were also labeled with the same tool in order to gather the ground-truth bounding box coordinates which were required for the evaluation procedure. In this case, the main metric that was chosen to assess the models' behaviour was the **mean average precision** (**mAP**), which considers both the category and the location of the classifications. To calculate it, a custom function was created that first computes the IoU between the ground-truth and the predicted bounding boxes to decide which predictions are true positives and which are false positives depending on whether an established **IoU threshold** is surpassed or not. With this information, the precision-recall curve can then be constructed, with precision being the ratio between the correctly classified positive samples to the total number of samples classified as positive, and recall being equivalent to sensitivity [150]. After proper smoothing, the AUC is calculated by means of the 11-point **interpolation** technique introduced in the PASCAL VOC challenge, obtaining the so-called average precision (AP), which is replaced by the mAP when the detection involves multiple categories [150].

The results obtained from this testing phase are exhibited in Table 4.2. As it can be seen, all the feature descriptors presented for the purpose of image classification were included, but only paired with the ML classifier which yielded the **best** performance results in the previous evaluation stage. The mAP (or AP in this case as there is only 1 class) was calculated for various confidence and IoU thresholds to investigate each

algorithm's performance in terms of its **classification** and **localisation** capabilities separately. In this way, the "standard" mAP was calculated considering all positive detections whose confidence surpassed a 0.5 probability threshold, the bare minimum to not be considered a background ROI, and whose IoU overlap score with the ground-truth bounding boxes was of at least 0.5, the usual norm for "good" predictions [150]. Building on this basis, two more metrics were computed: a mAP score for predictions with a confidence level greater than 0.8, to test the models' classification accuracy, and a mAP score for predictions with an IoU greater than 0.8, to test the models' ability to correctly localise positive samples within the image. Furthermore, the **mean time** required for the detection process on a single image was also recorded and incorporated as an evaluation metric, as this information is of extreme importance when pursuing real-time applications.

		Region proposal	
		Image pyramid	*Selective search*
AI model	*GLCM + RF*	0.0560 0.0335 0.00521 83 ± 2	0.0542 0.0285 0.00864 8 ± 1
	LBP + RF	0.0691 0.273 0.0128 5 ± 1	0.0611 0.125 0.0 5 ± 1
	HOG + SVM	0.239 0.361 0.0200 1.8 ± 0.8	0.216 0.112 0.0585 5 ± 2
	SIFT + KNN	0.0404 0.0267 0.00869 400 ± 100	0.0526 0.0598 0.0103 10 ± 7
	CNN	0.220 0.0370 0.0125 6.1 ± 0.9	0.318 0.407 0.0 6 ± 2
	YOLOv5	0.271 0.213 0.210 0.14 ± 0.05	

Table 4.2: Colour code: mAP (conf > 0.5, IoU > 0.5), mAP (conf > 0.8, IoU > 0.5), mAP (conf > 0.5, IoU > 0.8) and mean detection time (s).

Amongst the image classifiers repurposed as object detectors, significant behavioural dissimilarities can be observed. In the case of the feature descriptor and ML classifier pairings, a significant performance difference in the "standard" mAP **cannot** be observerd between the two region proposal approaches, but when the confidence threshold is increased, the mAP is significantly better with the image pyramid technique. However, the CNN presents the **opposite** behaviour: when a greater confidence level is demanded, the selective search approach constitutes a much better alternative, with a significant difference of an order of magnitude. Therefore, decisive conclusions cannot be extracted from this data, remaining unclear which region proposal technique provides better performance results in two-stage object detection algorithms as this may depend on the ML model in question. The same can be inferred in terms of the time required for each detection, with some models presenting almost the exact same time intervals across both approaches (e.g. *LBP + RF* and *CNN*), while others exhibit enormous **disparities**, notoriously the *SIFT + KNN* combination which stands in the order of minutes when an image pyramid and sliding window strategy is employed.

In terms of the comparison between one-stage and two-stage object detectors, the results corroborate the hypotheses encountered in the bibliographic sources cited at the beginning of this Chapter, with two-stage detectors exhibiting better **accuracy** and one-stage algorithms greater **speed**. With respect to accuracy, the findings show how the *YOLOv5* detector is outperformed both in the standard mAP as well as in the confidence-enhanced mAP by the CNN, and surpassed by the *HOG + SVM* and *LBP + RF* models in terms of the latter metric. Interestingly, however, when the IoU thresold is increased, the *YOLOv5* detector showcases a better response than the other AI models, experiencing a much smaller decrease than the rest with respect to the other mAP values, remaining considerably competitive whilst the others provide near-null results. This suggests that two-stage methods are superior when **distinguishing** between the object of interest and the rest of the image, but one-stage techniques carry out a more desirable **localisation** of positive samples. This seems reasonable considering that the region proposal is carried out internally in these networks and so the weights updated during training will also aim to reduce errors in bounding box predictions. Nevertheless, the biggest and most significant disparity between both categories relies in the time required for detection, with one-stage detectors in the order of *ms* and two-stage algorithms in the order of several seconds, difference which proves to be **pivotal** when pursuing real-time detection.

Chapter 5

Deployment on a Robotic Rehabilitation System

Having completed the investigation and evaluation of an extensive framework of diverse digital image processing techniques for real-time object detection applicable to **any specific use case**, the design of a coordinate finding mechanism and an inverse kinematic controller must be accomplished in order to supply the robot with the ability to **approach** the identified targets. As the popular strategy of modular decomposition, introduced in Section 2.1.2, has been applied to reduce the robot architecture's complexity and increase its reliability, the integration of the software developed throughout this MSc Thesis into ROS will also be required. In this way, a structure consisting of simple, independent nodes each assuming a **subset** of the navigation system's tasks, for instance, object detection or motion planning, will be utilised. Therefore, the developed vision sensor feedback functionality will need to **communicate** with other processes which will pass the planned motion trajectories onto the actuator hardware systems of a partial body weight-supported and traction powered walker where they will be executed.

5.1 Object coordinate calculation

Once the object of interest has been detected in the scene through any of the previously presented object-recognition algorithms, its **location** in the environment must be determined so that the appropriate trajectory to reach it can be devised. Hence, the next step in the implementation of a guidance system from vision sensor feedback must be the design of an object **co-coordinate finding** strategy. This task is not simple since the implemented arrangement relies on a single camera and photogrammetric systems require at least two projections of an object, that is, from two simultaneous photographs taken from different perspectives, in order to determine its 3D space coordinates [151].

5.1.1 3D position estimation

Achieving 3D perception with a single camera is possible with the sole condition of knowing the **size** of the object in the picture whose 3D location is to be estimated

[152]. There are multiple techniques with different degrees of complexity from the very straightforward **triangle similarity** to the complex, yet more accurate, use of the **intrinsic parameters** of the camera model [153]. The first option sets up a simple mathematical relationship between the object's known width W, its apparent width in pixels P and the perceived focal length of the camera F, from which the distance D to the object can be determined with a simple calibration step (Equation 5.1) [153]. However, a more **accurate** relationship between a 3D point in the real world and its corresponding 2D projection (pixel) in the image can be established through several camera parameters or coefficients that can be estimated by performing a proper camera calibration [154].

$$F = (P \cdot D)/W \tag{5.1}$$

Camera calibration is the process of understanding the **intrinsic** and **extrinsic** parameters of the camera [152]. These describe how the three different coordinate systems involved in the geometry of image formation, exhibited in Equation 5.1, are related: the **world** coordinate system, the **camera** coordinate system and the **image** coordinate system [155]. The extrinsic parameters are rotation and translation matrices that describe the relation between a given 3D coordinate system in the world with the position of the camera, and the intrinsic parameters are internal parameters of the camera or lens system (e.g. focal length, optical centre, radial distortion coefficients, etc.) that govern the relation between the 3D coordinate of the object and the 2D pixel coordinate of the image captured by the camera [153] [152].

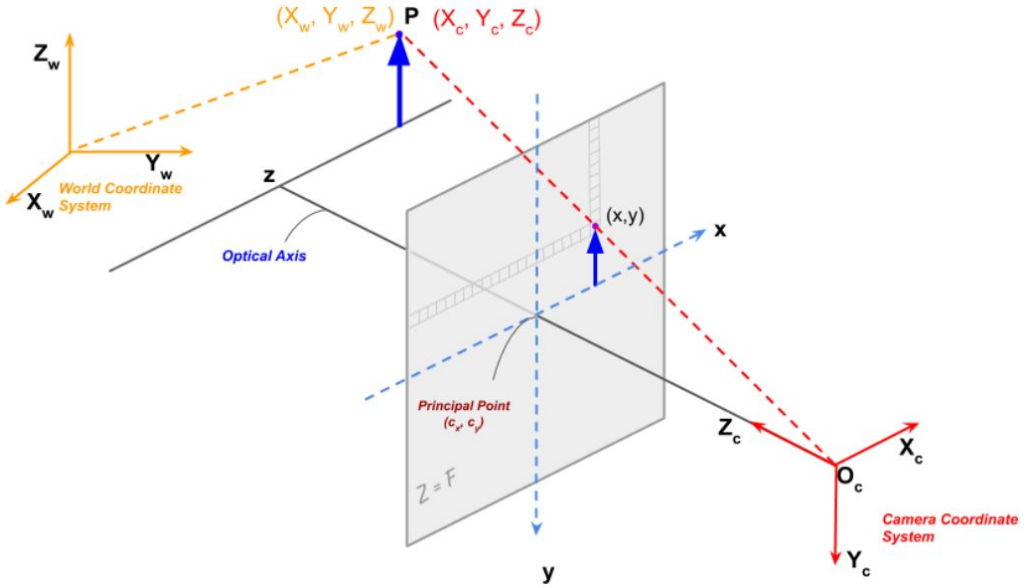


Figure 5.1: Geometry of image formation [155].

The first step in finding the projection of a 3D point onto the image plane is transforming the point from the world coordinate system to the camera coordinate system using the extrinsic parameters: the rotation matrix **R** and the translation vector **t** [154]. The camera coordinate system is defined as a 3D Cartesian coordinate system where the origin is located at the **focus point** of the camera and the Z-axis is the **optical axis** of the camera, as shown in Figure 5.1 [152]. Then, the point is projected onto the **image plane** using the camera's intrinsic parameters and so the 3D point in world coordinates (X_w, Y_w, Z_w) is related to its projection (u, v) through the projection matrix **P** as follows:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{5.2}$$

With:

$$u = \frac{u'}{w'} \tag{5.3a}$$

$$v = \frac{v'}{w'} \tag{5.3b}$$

$$\mathbf{P} = \mathbf{K} \times [\mathbf{R}|\mathbf{t}] \tag{5.3c}$$

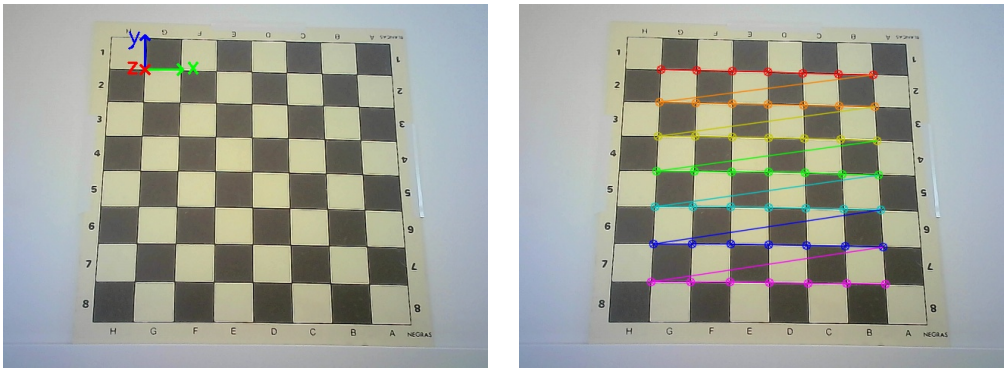
$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3d}$$

Where **K** is the intrinsic matrix, $[\mathbf{R}|\mathbf{t}]$ is the extrinsic matrix, f_x and f_y are the x and y focal lengths, c_x and c_y are the x and y coordinates of the optical centre in the image plane and γ is the skew between the axes.

5.1.2 Camera calibration

The goal of the calibration process is to find the 3×3 matrix **K**, the 3×3 rotation matrix **R** and the 3×1 translation vector **t** using a collection of images with points whose 2D image coordinates (u, v) and 3D world coordinates (X_w, Y_w, Z_w) are **known**. There are different types of camera calibration methods; in this case, geometric clues in the scene will be used by capturing several images of an object or pattern of known dimensions from different view points and orientations [154]. For this purpose, the corners of the squares in a **checkerboard** pattern will be located since these are distinct and easy to detect in an image, their sharp gradients in both directions are ideal for localisation and their world coordinates are easily defined by taking a single point as reference and defining the rest with respect to it since they all lie on the same plane and are equally spaced [154].

In this way, a chessboard was fixed to a wall in the camera's surroundings and the world coordinate system's origin was established in its **top-left corner**, as shown in Figure 5.2 (a). Multiple pictures of this pattern were taken from different positions and orientations maintaining the same coordinate system. The pixel coordinates (u, v) of each of the chessboard's internal corners were determined with sub-pixel accuracy, as can be observed in Figure 5.2 (b), using histogram equalisation and adaptive thresholding for image binarisation as well as several computational load optimisation strategies. Finally, the camera's intrinsic matrix and **distortion** coefficients were calculated employing each corner's 2D and respective 3D coordinates as well as the rotation matrix and translation vector for each camera position and orientation, since these last parameters are **unique** for each perspective.



(a) *(b)*

Figure 5.2: Camera calibration *(a)* Checkerboard pattern with world coordinate system axes *(b)* Corner detection.

5.1.3 Implementation

Taking into consideration all the information presented in this section and the project's requirements, a suitable object coordinate finding strategy was devised. Since the rotation matrix **R** and the translation vector **t** that relate the camera coordinate system to the world coordinate system **vary** according to the camera's position and orientation, these cannot be applied as the camera will be in **constant movement** as it approaches an object of interest and, thus, these parameters will not be valid for any possible configuration. In this case, determining the object's position solely in terms of the camera coordinate system seems like a better option since this reference system will always be centered in the camera, thus removing the uncertainty derived from the camera's movement, and **simplifying** the trajectory calculation from the camera to the object as the former is the origin $(0, 0, 0)$ of the system.

Following this line of reasoning, **R** can be taken as the identity matrix **I** and **t** as $[0\ 0\ 0]^T$ given that the world and camera coordinate systems are now **equivalent**. The resulting relationship between world and image coordinates is described by

Equation 5.4, where S is a scaling factor. When expanded, this association gives rise to **three** simultaneous equations with **four** unknown variables (S, X_w, Y_w, Z_w), thus rendering the system unsolvable. However, Z_w can also be obtained from Equation 5.1, where X_w and Y_w can be inferred from Equation 5.5 (a) and Equation 5.5 (b) respectively, since $S = Z_w$. In this way, the relationships derived from both the *triangle similarity* and the camera's intrinsic parameters are **combined** to estimate the object's 3D location.

$$S \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{5.4}$$

$$X_w = \frac{S\,(u - c_x)}{f_x} \tag{5.5a}$$

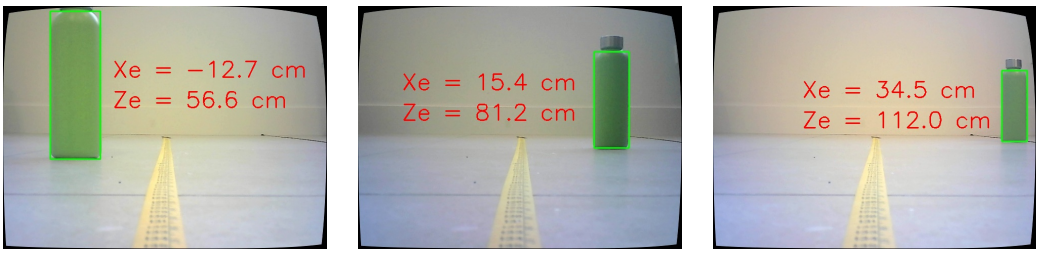
$$Y_w = \frac{S\,(v - c_y)}{f_y} \tag{5.5b}$$

Consequently, an additional calibration step was required in order to determine the camera's **perceived focal length** F, which was derived from Equation 5.1 by taking multiple pictures of an object of known width W at different distances D from the camera. With this datum, the object's coordinates can now be calculated by first **undistorting** the image with the distortion coefficients obtained through the camera calibration process, then calculating the **distance** to the object (Z_w) by means of the *triangle similarity*, subsequently **segmenting** the image in order to find the pixel coordinates of the object's centre and, finally, determining X_w and Y_w with the camera's intrinsic parameters.

The developed procedure's effectiveness was tested by placing an object at different positions with respect to the camera and **comparing** its estimated location with its actual coordinates, determined by means of a measuring tape. Both of these variables are displayed in Figure 5.3 for each of the multiple scenarios that were tested at varying distances from the camera (*short*, *medium*, *long*) so as to analyse the possible **influence** of this variable on the quality of results. The presented images are undistorted and the object was manually segmented in this case in order to remove the dependence of the object's apparent width in pixels P on the accuracy of the segmentation method employed, which may vary depending on various factors as has been previously discussed in prior chapters.

Given that the robot's task is to approach an object by moving on a plane, only **two** of its 3D coordinates are required to characterise its position for trajectory planning. Therefore, in terms of the camera's coordinate system, the object's Y_w coordinate, which represents its relative "*height*" with respect to the camera, is **irrelevant** since the robot will not dispose of this degree of freedom and so only the estimated and actual coordinates in the XZ plane are specified. As it can be observed, the

estimations (X_e, Z_e) are considerably reasonable for all scenarios, regardless of the object's distance from the camera, with an error of just 0.6 ± 0.6 cm in X_w and 7.0 ± 3.0 cm in Z_w.



(a) *(b)* *(c)*

Figure 5.3: Object coordinate calculation test *(a)* Short distance: $\{X_w, Z_w\} = \{-12, 60\}$ *cm (b)* Medium distance: $\{X_w, Z_w\} = \{16.5, 90\}$ *cm (c)* Long distance: $\{X_w, Z_w\} = \{34.5, 120\}$ *cm.*

5.2 Trajectory planning

Path planning is the task of finding a continuous, collision-free path connecting a system from an **initial** to a **final** goal configuration in a given environment by determining and evaluating plausible trajectories [156]. This is one of the most important functions of any navigational technique as its performance directly decides whether the robot's task is **successful** or not [157] [55]. The optimal path to follow can be decided based on **constraints** and conditions with respect to time, distance and energy, among other factors, with multiple algorithms existing in order to identify safe, efficient, collision-free and least-cost travel paths from origin to destination with varied applicability determined by the system's kinematics, the environment's dynamics, robotic computation capabilities, and sensor-and other-sourced information availability [156].

A custom **controlled-path** motion control system will be applied for the computation of the trajectory to the object's previously calculated world coordinates. Developed by Álvaro Sala Ayala from ROBOLABO [1], this ROS server node receives as parameters the angle α the *Swalker* must **turn** to reach the new, desired configuration, as shown in Figure 5.4, and the wheel, left or right, that must perform such movement [158]. In this way, the robot's current position and the object's 2D coordinates are interpreted as a triangle, with the robot first performing a turn corresponding to the angle formed between both positions, and then moving in a **straight line** along the hypotenuse that joins them in the shortest distance. Therefore, the desired movement is achieved through continuous petitions to this node with $\alpha = \arctan X_e/Z_e$.

[1] https://robolabo.etsit.upm.es

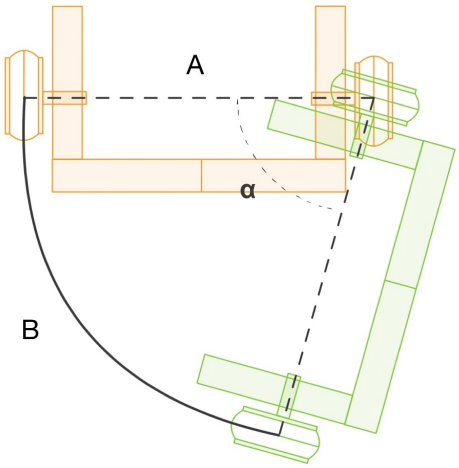


Figure 5.4: Motion planning: α turn followed by a straight line trajectory.

5.3 Materials

Aside from the USB camera employed in the previous Chapters for the performance of the video analysis, object detection techniques, two additional materials are required for the navigation's system's deployment: a **hardware device** on which to set up the ROS node that processes the camera's live feed, and a **rehabilitation robot** on which to test the system's correct functioning.

5.3.1 *Jetson Nano*

The hardware deployment of the object detection navigation system was performed on a **single-board computer** due to their low cost, small weight and high computing capability, thus constituting the best option for the incorporation of the developed system into the preexisting robotic device. The *Raspberry Pi* and the *Jetson Nano* boards share a lot of common properties and are arguably the most popular devices for the sort of application pursued in this Thesis. Given that the biggest difference between the two lies in their **graphics capabilities**, with the *Jetson Nano* possessing a much more capable graphical processing unit (GPU), the final choice was to opt for said tool, displayed in Figure 5.5 (a), as the GPU's parallel processing ability considerably enhances the **speed** of the computations involved in ML model training and inference [159].

The official operating system for the *Jetson Nano*, the Linux4Tegra based on Ubuntu 18.04, was burnt onto a microSD card, illustrated in Figure 5.5 (b). This is a **key component** of this single-board computer as it acts as both the boot device and the main storage, and, in this case, it was also configured to provide **swap space** given the *Nano*'s limited 2GB physical memory RAM. Although the aforementioned operating system is designed to run on NVIDIA hardware, including GPUs, some initial configuration alterations were carried out to obtain CUDA support for certain applications required for this navigation system, such as *OpenCV*.

(a) *(b)*

Figure 5.5: Hardware deployment *(a) Jetson Nano* single-board computer [160] *(b)* 64 GB micro SD card [161].

5.3.2 *Swalker*

The relentless growth of individuals with a limited **range of motion** and strength in their lower limbs, mainly due to the increasing older population in developed countries, has made technology pursue new challenges to improve this collective's quality of life and independence. With the increased use of robotic devices in the last decades, specially those focused on neurorehabilitation of disorders, one of the newest applications has been the development of robotic platforms to **rehabilitate** musculoskeletal diseases. The *Swalker* platform, whose main functional goal is to facilitate these individuals' early **mobilisation** and ambulation using a safety walker frame, is found at the forefront of this movement [162]. Early intervention with this partial body weight-supported and traction powered walker will aim to reduce the high morbidity and increase the **independence** after therapy [162].

The *Swalker* frame, illustrated in Figure 5.6, consists of a **T-shaped** structure supported on four wheels and two adjustable parallel grab bars for greater comfort and safety perception. Two gear motors equipped with encoders are coupled to the back wheels to enable **motorised traction** while providing speed information, and a mechanical mechanism involving two cylinder-piston actuators activated by a hydraulic pump, controlled by an electric motor, provides the **body-weight support**. The user's stability is achieved by an adaptable trunk harness, alongside the aforementioned parallel bars. Several sensors are also included to provide relevant outcome measures, in particular, hip range of motion, weight supported by the robot and gait speed, which aid in the determination of optimum rehabilitation **dosage** and **progress**. A control unit is incorporated to capture the information from the sensors and control the actuators accordingly. Finally, a graphical therapist interface, running on a conventional tablet, is designed to control the *Swalker*, monitor the registered parameters and **display** a patient database where all the information about the therapy sessions is stored.

Figure 5.6: The *Swalker* robotic platform for early mobilisation and ambulation of individuals with limited range of motion and strength in their lower limbs [162].

5.4 ROS integration

The workflow of the final software, that takes an image from the camera's live feed as the input and outputs the motion trajectory that must be performed to reach an object in the robot's surroundings, is exhibited in Figure 5.7. This algorithm is deployed on a **ROS 2 node** on the *Jetson Nano*, which broadcasts the parameters that characterise the movement to be performed so that the node responsible for translating those commands to the actuators can receive this information by subscribing to said topic. *Rospy*, the Python client library for ROS, is applied to set up the node and the topic as well as the **communication** with other nodes, as it enables the specification of numerous factors such as the message instance, the queue size or the rate of publishing.

As it can be observed, as soon as the node is initialised, a connection is established with the camera, connected through the USB port, by means of the *OpenCV* client. While the node is operating, an infinite loop performs **continuous** frame-by-frame manipulation of the live feed supplied by the camera, which is placed on the *Swalker*'s structure. After a frame is extracted, it is first undistorted using the camera's distortion coefficients obtained through the calibration process, as this will contribute towards a greater accuracy both in the model's inference and in the 2D world coordinate estimation. The next step involves further **preprocessing** of the captured image, this time requiring different techniques depending on the AI model employed for objection detection, as the exact same procedure employed during the training stage, including dependencies, will have to be **reproduced** in order to obtain accurate results. As presented in the previous chapter, this could entail a wide range of different digital image processing techniques ranging from a simple grayscale conversion to more sophisticated histogram equalisation or normalisation methodologies.

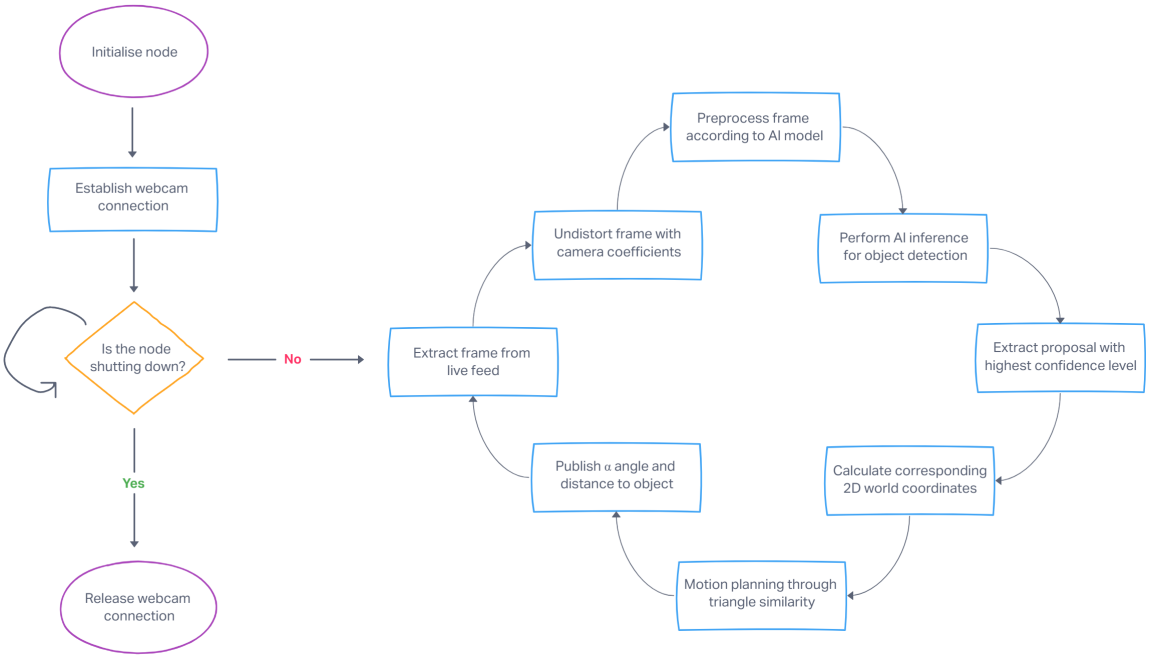


Figure 5.7: Final algorithm that generates a motion trajectory from an input image in real-time.

Once the image is adequately prepared, it must be fed to the AI algorithm of choice for the recognition and localisation of the object it has been trained to identify in the specific use case. Depending on the chosen architecture, a previous stage of region proposal might be needed. The model will return a **variable** number of bounding box proposals with an associated probability of containing the object. Given that the robot can only approach a single target at a time, a filtering criteria must be implemented to ensure that only one of the candidate regions progresses onto the motion planning phase; in this case, the **confidence** level related to each proposal is employed to select the ROI that possess the highest probability of containing the object. The front and lateral distances to the object are then calculated based on the proposed ROI's bounding box coordinates and the trajectory to that spatial point is determined following the procedures explained in previous sections, with the final outcome being the publication of the angle turn and straight line distance required to reach the target. If no object instance is detected, both variables receive null values to indicate to the subscribing node that **no movement** is required. Whenever the node receives a shut down request, this loop is interrupted and the webcam connection is released to ensure a proper programme termination.

Chapter 6

Conclusions and Future Developments

6.1 Conclusions

The devastating consequences brought upon worldwide communities by TBI, an unknown condition to most people, have been highlighted in order to underline the importance of taking action and **improving the care** of those affected given the considerable socioeconomic impact this could entail. As discussed throughout this MSc Thesis, the loss of functionality and occupational performance derived from the multiple impairments associated with this condition triggers a wide spectrum of **far-reaching repercussions** that range from economic burden to emotional distress and decreased quality of life among family members, aside from the natural psychosocial effects on the patients themselves including loss of self-esteem and depression. With rehabilitation proven to be essential to the recovery of motor function outcomes, which frequently limit these patients' self-independence, an improvement of the currently available physical therapies could be the **key** to unlocking a greater quality of life, not only for the afflicted individuals, but also for those closest to them.

Considering the vast and rapid developments of robots in the field of rehabilitation and their effectiveness in **ameliorating recovery**, a computer vision based guidance system is proposed to grant these devices the ability to recognise objects or people in their surroundings so that they are capable of guiding the patient's movement towards a physical target, aside from simply providing a helping force, thus promoting the patient's active participation and **motivation**. This approach to enhancing the therapy's success and adapting therapeutic interventions to each patient has required the application of a wide range of different digital image processing techniques for the purpose of real-time object recognition and localisation. After an extensive evaluation of these techniques, the designed and implemented vision sensor feedback has been effectively deployed on an assistive walking device employing a modular node based on ROS that communicates with the other modules of the *Swalker* control system to successfully perform the required task, consequently fulfilling **all the objectives** set out at the beginning of the project.

Special emphasis has been placed on the analysis and familiarisation with state-of-the-art technologies in the field of **computer vision**, with the initial investigation of simpler, more traditional techniques being essential to the achievement of a profound understanding of image processing, which is definitely necessary before progressing to the most advanced solutions that are actually applied in professional settings. Overall, the object detection results obtained may seem slightly disappointing at first sight. Understanding that the art of training high-quality AI models isn't so much in the code, but more in **gathering the data** to train the model with, is crucial in the development of these solutions. Not only is quality data required, but also lots of it, as this exerts a huge impact on the results. Considering the small size of the training and testing datasets employed in this application, given the need to manually acquire and label the images, the amount of variability introduced in both the training and testing datasets has been exceedingly large, yielding subpar performance results. However, satisfying results, along with the retrieval of robust models, should be expected with this methodology when using **significantly larger datasets**.

6.2 Future developments

As the first step of a broad and ambitious project encompassing many different areas of technological expertise and complexity, there are multiple functionalities that can be **incorporated** or **improved** in order to enhance both the patient's experience and the therapy's effectiveness. As previously stated, the performance of the developed object detection algorithms is not sufficiently robust so as to ensure an adequate and **reliable** functioning of the vision sensor feedback. Therefore, the most immediate improvement would be the generation of a larger training database either through manual acquisition or by obtaining a large number of images containing a particular object from a public data repository. However, this last strategy could result problematic as the training images would be acquired with different camera characteristics than the ones employed for inference. The models' behaviour could also be enhanced by specifying a **particular environment** where the detection would be carried out in order to narrow down the environmental conditions under which the model will operate, and so the variability of factors, such as lighting, during training could be reduced to more reasonable levels.

Other areas of refinement could be the employment of AI strategies for a more accurate calculation of the object's **spatial position**, rather than performing an estimation based on the *triangle similarity*, or the exploitation of a stereo vision system, capable of computing depth information using two cameras. In terms of new features that could be incorporated, **object tracking** is a widely implemented technique in the field of computer vision which, as its name suggests, enables tracking detected objects as they move around frames in a video, interpreting them as a set of trajectories with high accuracy [163]. Its application, once an initial detection is performed, would enable updating the robot's trajectory according to the object's movement without having to carry out the entire detection process, translating into a lower computational cost and saving time. Another interesting approach,

specially in terms of patient engagement and motivation during therapy, would be the incorporation of **virtual reality** so that the patient can also approach objects in an immersive virtual environment with the *Swalker*'s aid.

Finally, the most important action that needs to be carried out to complete this project is its **clinical validation**. As in any engineering activity, the feedback obtained from potential users is crucial in creating a successful product as it directs the project's focus towards the aspects that are most important to those individuals that will actually end up using the proposed solution. Taking into consideration the field in which this project is encompassed, that is the health sector, this validation phase is **particularly important** when pursuing the incorporation of the developed functionalities into rehabilitation protocols so that they can reach the target patients as this sector is specially demanding on proving a solution's effectiveness, given the potential risk of faulty devices on people's health. As such, the elaborated navigation system should be tested on a significant sample of volunteer patients to gather **clinical evidence** of its actual effectiveness in improving motor function recovery as well as to obtain **feedback**, from both patients and medical staff, to continue its development into a more complete solution which can finally reach the market and exert a considerable beneficial impact on society as a whole.

Bibliography

[1] Center for Brain Injury and Repair. Traumatic brain injury: A "silent epidemic". Perelman School of Medicine — University of Pennsylvania. [Accesed online] 06/06/2022. Available at: https://www.med.upenn.edu/cbir/silentepidemic.html#:~:text=Brain%20injury%20is%20suffered%20by,a%20severe%20brain%20injury%20annually.

[2] Dang B, Chen W, He W, and Chen G. Rehabilitation treatment and progress of traumatic brain injury dysfunction. *Neural Plast*, 2017.

[3] David Martínez-Pernía. Experiential neurorehabilitation: A neurological therapy based on the enactive paradigm. *Frontiers in Psychology*, 11, 2020.

[4] A.I. Maas, N. Stocchetti, and R. Bullock. Moderate and severe traumatic brain injury in adults. *The Lancet Neurology*, 7(8):728–741, 2008.

[5] Levin HS, Shum D, and Chan RC. *Understanding Traumatic Brain Injury.* Oxford University Press, New York (NY), 2014.

[6] Galgano M, Toshkezi G, Qiu X, Russell T, Chin L, and Zhao LR. Traumatic brain injury: Current treatment strategies and future endeavors. *Cell Transplant*, 26(7):1118–1130, 2017.

[7] Maas AIR, Menon DK, Adelson PD, Andelic N, Bell MJ, Belli A, and et al. Traumatic brain injury: integrated approaches to improve prevention, clinical care, and research. *Lancet Neurol.*, 16:987–1048, 2017.

[8] E. Jiménez Arberas, F. F. Ordoñez Fernández, and S. Rodríguez Menéndez. Psychosocial impact of mobility assistive technology on people with neurological conditions. *Disability and Rehabilitation: Assistive Technology*, 16(5):465–471, 2019.

[9] Andelic Nada, Løvstad Marianne, Norup Anne, Ponsford Jennie, and Røe Cecilie. Editorial: Impact of traumatic brain injuries on participation in daily life and work: Recent research and future directions. *Frontiers in Neurology*, 10:1664–2295, 2019.

[10] Davis RH, Alexander LT, and Yelon SL. *Learning System Design - An approach to the improvement of instruction.* McGraw-Hill Inc, New York, 1974.

[11] Taylor DP. Treatment goals for quadriplegic and paraplegic patients. *Am J Occup Ther.*, 28(1):22–9, 1974.

[12] Chiou II and Burnett CN. Values of activities of daily living. a survey of stroke patients and their home therapists. *Phys Ther*, 65(6):901–6, 1985.

[13] L. Li, S. Tyson, and A. Weightman. Professionals' views and experiences of using rehabilitation robotics with stroke survivors: A mixed methods survey. *Frontiers in Medical Technology*, 3, 2021.

[14] Norup A, Kruse M, Soendergaard PL, Rasmussen KW, and Biering-Sørensen F. Socioeconomic consequences of traumatic brain injury: A danish nationwide register-based study. *J Neurotrauma*, 37(24):2694–2702, 2020.

[15] Health Topics — MedlinePlus. Traumatic brain injury. Online. [Accesed online] 26/03/2022. Available at: https://medlineplus.gov/traumaticbraininjury.html#:~:text=Traumatic%20brain%20injury%20(TBI)%20is,This%20is%20a%20penetrating%20injury.

[16] John Hopkins Medicine. Traumatic brain injury. Online. [Accesed online] 18/03/2022. Available at: https://www.hopkinsmedicine.org/health/conditions-and-diseases/traumatic-brain-injury.

[17] Dewan M. C., Rattani A., Gupta S., Baticulon R. E., Hung Y., Punchak M., Agrawal A., Adeleye A. O., Shrime M. G., Rubiano A. M., Rosenfeld J. V., and Park K. B. Estimating the global incidence of traumatic brain injury. *Journal of Neurosurgery JNS*, 130(4):1080–1097, 2019.

[18] Bryan-Hancock C and Harrison J. The global burden of traumatic brain injury: preliminary results from the global burden of disease project. *Injury Prevention*, 16(1), 2010.

[19] A. Abio, P. Bovet, B. Valentin, T. Bärnighausen, M. A. Shaikh, J. P. Posti, and M. Lowery Wilson. Changes in mortality related to traumatic brain injuries in the seychelles from 1989 to 2018. *Frontiers in Neurology*, 12, 2021.

[20] Prins M, Greco T, Alexander D, and Giza CC. The pathophysiology of traumatic brain injury at a glance. *Dis Model Mech*, 6(6):1307–1315, 2013.

[21] Physiopedia. Epidemiology of traumatic brain injury. Online. [Accesed online] 28/03/2022. Available at: https://www.physio-pedia.com/Epidemiology_of_Traumatic_Brain_Injury.

[22] Kirsteen Burton z S. Alali, Robert A. Fowler, David M.J. Naimark, Damon C. Scales, Todd G. Mainprize, and Avery B. Nathens. Economic evaluations in the diagnosis and management of traumatic brain injury: A systematic review and analysis of quality. *Value in Health*, 18(5):721–734, 2015.

[23] Feigin VL, Theadom A, Barker-Collo S, Starkey NJ, McPherson K, Kahan M, Dowell A, Brown P, Parag V, Kydd R, Jones K, Jones A, and Ameratunga S. Incidence of traumatic brain injury in new zealand: a population-based study. *Lancet Neurol*, 12(1):53–64, 2013.

[24] S. L. James, A. Theadom, R. G. Ellenbogen, M. S. Bannick, W. Montjoy-Venning, L. R. Lucchesi, and et al. Global, regional, and national burden of traumatic brain injury and spinal cord injury, 1990–2016: A systematic analysis for the global burden of disease study 2016. *The Lancet Neurology*, 18(1):56–87, 2019.

[25] García-Altés A, Pérez K, Novoa A, Suelves JM, Bernabeu M, Vidal J, Arrufat V, Santamariña-Rubio E, Ferrando J, Cogollos M, Cantera CM, and Luque JC. Spinal cord injury and traumatic brain injury: a cost-of-illness study. *Neuroepidemiology*, 39(2):103–8, 2012.

[26] Manskow US, Friborg O, Røe C, Braine M, Damsgard E, and Anke A. Patterns of change and stability in caregiver burden and life satisfaction from 1 to 2 years after severe traumatic brain injury: A norwegian longitudinal study. *NeuroRehabilitation*, 40(2):211–222, 2017.

[27] Wood RL and Yurdakul LK. Change in relationship status following traumatic brain injury. *Brain Inj*, 11(7):491–501, 1997.

[28] Si Yun Ng and Alan Yiu Wah Lee. Traumatic brain injuries: Pathophysiology and potential therapeutic targets. *Frontiers in Cellular Neuroscience*, 13, 2019.

[29] Office of Communications — National Institutes of Health (NIH). What are common symptoms of traumatic brain injury (tbi)? Online, 2020. [Accesed online] 26/03/2022. Available at: https://www.nichd.nih.gov/health/topics/tbi/conditioninfo/symptoms.

[30] Parmeet Kaur and Saurabh Sharma. Recent advances in pathophysiology of traumatic brain injury. *Curr Neuropharmacol*, 16(8):1224—-1238, 2018.

[31] S. Alghnam, A. AlSayyari, I. Albabtain, B. Aldebasi, and M. Alkelya. Long-term disabilities after traumatic head injury (thi): a retrospective analysis from a large level-i trauma center in saudi arabia. *Injury epidemiology*, 4(1):29, 2017.

[32] Eng JJ, Rowe SJ, and McLaren LM. Mobility status during inpatient rehabilitation: a comparison of patients with stroke and traumatic brain injury. *Arch Phys Med Rehabil*, 83(4):483–490, 2002.

[33] Sameera Haffejee, Veronica Ntsiea, and Witness Mudzi. Factors that influence functional mobility outcomes of patients after traumatic brain injury. *Hong Kong Journal of Occupational Therapy*, 23(1):39–44, 2013.

[34] R. Gassert and V. Dietz. Rehabilitation robots for the treatment of sensorimotor deficits: a neurophysiological perspective. *J NeuroEngineering Rehabil*, 15(1), 2018.

[35] Voss P, Thomas ME, Cisneros-Franco JM, and de Villers-Sidani É. Dynamic brains and the changing rules of neuroplasticity: Implications for learning and recovery. *Front Psychol*, 8, 2017.

[36] David J. Reinkensmeyer. rehabilitation robot. Britannica, 2013. [Accesed online] 05/06/2022. Available at: https://www.britannica.com/technology/rehabilitation-robot.

[37] Maier M, Ballester BR, and Verschure PFMJ. Principles of neurorehabilitation after stroke based on motor learning and brain plasticity mechanisms. *Frontiers in Systems Neuroscience*, 11:13–74, 2019.

[38] Karen J. Nolan, Kiran K. Karunakaran, Kathleen Chervin, Michael R. Monfett, Radhika K. Bapineedu, Neil N. Jasey, and Mooyeon Oh-Park. Robotic exoskeleton gait training during acute stroke inpatient rehabilitation. *Front. Neurorobot.*, 14, 2020.

[39] Sarah E. Chard. Community neurorehabilitation: A synthesis of current evidence and future research directions. *The Journal of the American Society for Experimental NeuroTherapeutics*, 3(4):525–534, 2006.

[40] Avijeet Biswal. Ai applications: Top 14 artificial intelligence applications in 2022. simplilearn, 2022. [Accesed online] 12/05/2022. Available at: https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/artificial-intelligence-applications.

[41] Robotics. Builtin. [Accesed online] 12/05/2022. Available at: https://builtin.com/robotics.

[42] Ziyad Saleh. Artificial intelligence definition, ethics and standards. *Electronics and Communications: Law, Standards and Practice*, 2019.

[43] Robotics. ScienceDirect. [Accesed online] 12/05/2022. Available at: https://www.sciencedirect.com/topics/social-sciences/robotics.

[44] Robotics and the art of science. *Nat Mach Intell*, 259, 2019.

[45] Types of robots: How robotics technologies are shaping today's world. Intel. [Accesed online] 25/05/2022. Available at: https://www.intel.com/content/www/us/en/robotics/types-and-applications.html#:~:text=What%20are%20the%20main%20types,%2C%20humanoid%20robots%2C%20and%20hybrids.

[46] How robots change the world. Oxford Economics, 2020. [Accesed online] 25/05/2022. Available at: https://resources.oxfordeconomics.com/how-robots-change-the-world.

[47] Sachin Thorat. Robotics — introduction and classification of robotics. LearnMech — Mechatronics. [Accesed online] 16/05/2022. Available at: https://learnmech.com/robotics-introduction-and-classification/.

[48] Vision 60. Ghost Robotics, 2021. [Accesed online] 03/06/2022. Available at: https://www.ghostrobotics.io/vision-60.

[49] Emma Roth. A humanoid robot makes eerily lifelike facial expressions. The Verge, 2021. [Accesed online] 03/06/2022. Available at: https://www.theverge.com/2021/12/5/22819328/humanoid-robot-eerily-lifelike-facial-expressions.

[50] La tecnología de thales alenia space españa permitirá que viper, el robot que la nasa lanzará a la luna en busca de agua, se comunique directamente con la tierra. Thales, 2020. [Accesed online] 03/06/2022. Available at: https://www.thalesgroup.com/es/el-mundo/space/press-release/tecnologia-thales-alenia-space-espana-permitira-viper-el-robot-nasa.

[51] Satyam Ambast. Introduction to underwater robotics. Medium, 2020. [Accesed online] 03/06/2022. Available at: https://medium.com/reconsubsea/introduction-to-underwater-robotics-c4565502369.

[52] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall. Robot operating system 2: Design, architecture, and uses in the wild. *Science Robotics*, 7(66), 2022.

[53] Ameca: The future face of robotics. Engineered Arts. [Accesed online] 03/06/2022. Available at: https://www.engineeredarts.co.uk/es/robot/ameca/.

[54] David Kortenkamp and Reid Simmons. Robotic systems architectures and programming. In *Springer Handbook of Robotics*, pages 187–206. Springer, 2008.

[55] C. Zhou, B. Huang, and P Fränti. A review of motion planning algorithms for intelligent robots. *Journal of Intelligent Manufacturing*, 33:387—424, 2022.

[56] Kevin M. Lynch and Frank C. Park. Control system overview. Northwestern Robotics, 2018. [Accesed online] 17/05/2022. Available at: https://www.youtube.com/watch?v=mGuDXlZEoSc.

[57] Kevin M. Lynch and Frank C. Park. Hybrid motion-force control. Northwestern Robotics, 2018. [Accesed online] 17/05/2022. Available at: https://www.youtube.com/watch?v=UR0GpaaBVKk.

[58] Morgan Quigley. Ros: an open-source robot operating system. In *ICRA 2009*, 2009.

[59] ROS Wiki. What is ros? Online, 2018. [Accesed online] 13/03/2022. Available at: https://wiki.ros.org/ROS/Introduction.

[60] Tiziano Fiorenzani. What is ros (robot operating system)— introduction to the tutorials. YouTube, 2018. [Accesed online] 03/06/2022. Available at: https://www.youtube.com/watch?v=N6K2LWG2kRI.

[61] Ricardo Tellez. What is robot operating system (ros)? The Construct, 2019. [Accesed online] 03/06/2022. Available at: https://www.theconstructsim.com/what-is-ros/.

[62] Ryan Beasley. Medical robots: Current systems and research directions. *Journal of Robotics*, pages 1687–9600, 2012.

[63] Gyles C. Robots in medicine. *Can Vet J*, 60(8):819–820, 2019.

[64] Albert Cadanet. Objetivo, 63 millones: el gigante español de robots da vinci prevé crecer un 40% en 2019. PlantaDoce, 2019. [Accesed online] 21/05/2022. Available at: https://www.plantadoce.com/empresa/el-gigante-espanol-de-robots-da-vinci-preve-incrementar-un-40-sus-ventas-hasta-63-millones.html.

[65] Dr. Francisco Sánchez-Margallo (Centro de Cirugía Mínima Invasión Jesús Usón de Cáceres). Minimally invasive surgery: how technology revolutionised surgical treatments. In *Clinical Seminars*. Master of Science in Biomedical Engineering (UPM), 2021.

[66] About robotic surgery at ucla. UCLA Health. [Accesed online] 21/05/2022. Available at: https://www.uclahealth.org/robotic-surgery/what-is-robotic-surgery.

[67] Gorgey AS. Robotic exoskeletons: The current pros and cons. *World J Orthop*, 9(9):112–119, 2018.

[68] I.H Sarker. Ai-based modeling: Techniques, applications and research issues towards automation, intelligent and smart systems. *SN COMPUT. SCI.*, 3(2), 2022.

[69] What is computer vision? IBM. [Accesed online] 28/05/2022. Available at: https://www.ibm.com/topics/computer-vision.

[70] Bernard Marr. 7 amazing examples of computer and machine vision in practice. Enterprise Tech — Forbes, 2019. [Accesed online] 02/06/2022. Available at: https://www.forbes.com/sites/bernardmarr/2019/04/08/7-amazing-examples-of-computer-and-machine-vision-in-practice/#3dbb3f751018.

[71] Ai modeling: Driving intelligence in analytics. Intel. [Accesed online] 25/05/2022. Available at: https://www.intel.com/content/www/us/en/analytics/data-modeling.html.

[72] Firas Ghunaim. Ai vs machine learning vs deep learning: What's the difference. SiTech, 2022. [Accesed online] 26/05/2022. Available at: https://www.sitech.me/blog/ai-vs-machine-learning-vs-deep-learning.

[73] IBM Cloud Education. Supervised learning. IBM, 2020. [Accesed online] 25/05/2022. Available at: https://www.ibm.com/cloud/learn/supervised-learning.

[74] Durgesh Srivastava and Lekha Bhambhu. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 12:1–7, 2010.

[75] Amal Joby. What is k-nearest neighbor? an ml algorithm to classify data. Learn Hub — G2, 2021. [Accesed online] 27/05/2022. Available at: https://learn.g2.com/k-nearest-neighbor.

[76] What is the k-nearest neighbors algorithm? IBM. [Accesed online] 27/05/2022. Available at: https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.

[77] Shyamanth Besetty. Decision tree. GeeksforGeeks, 2022. [Accesed online] 06/06/2022. Available at: https://www.geeksforgeeks.org/decision-tree/.

[78] Tony Yiu. Understanding random forest. Towards Data Science, 2019. [Accesed online] 06/06/2022. Available at: https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

[79] Support vector machines. Scikit-Learn. [Accesed online] 26/05/2022. Available at: https://scikit-learn.org/stable/modules/svm.html.

[80] Rohith Gandhi. Support vector machine — introduction to machine learning algorithms. Towards Data Science, 2018. [Accesed online] 26/05/2022. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

[81] IBM Cloud Education. Unsupervised learning. IBM, 2020. [Accesed online] 25/05/2022. Available at: https://www.ibm.com/cloud/learn/unsupervised-learning.

[82] Imad Dabbura. K-means clustering: Algorithm, applications, evaluation methods, and drawbacks. Towards Data Science, 2018. [Accesed online] 28/05/2022. Available at: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

[83] Dr. Michael J. Garbade. Understanding k-means clustering in machine learning. Towards Data Science, 2018. [Accesed online] 28/05/2022. Available at: https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.

[84] IBM Cloud Education. Neural networks. IBM Cloud Learn Hub, 2020. [Accesed online] 28/05/2022. Available at: https://www.ibm.com/cloud/learn/neural-networks.

[85] Steven Walczak and Narciso Cerpa. Artificial neural networks. In *Encyclopedia of Physical Science and Technology (Third Edition)*, pages 631–645, New York, 2003. Academic Press.

[86] What is a neural network? MathWorks — MATLAB. [Accesed online] 28/05/2022. Available at: https://www.mathworks.com/discovery/neural-network.html.

[87] Jaimin Dave. A step by step guide to ai model development. Data Science Central, 2021. [Accesed online] 25/05/2022. Available at: https://www.datasciencecentral.com/a-step-by-step-guide-to-ai-model-development/.

[88] MATLAB. What is object detection? 3 things you need to know. Online. [Accesed online] 05/04/2022. Available at: https://www.mathworks.com/discovery/object-detection.html.

[89] Papers With Code: The Latest in Machine Learning. Object detection. Online. [Accesed online] 05/04/2022. Available at: https://paperswithcode.com/task/object-detection.

[90] Fritz AI. Object detection guide: Almost everything you need to know about how object detection works. Online. [Accesed online] 05/04/2022. Available at: https://www.fritz.ai/object-detection/.

[91] Ivana vCuljak, David Abram, Tomislav Pribanić, Hrvoje Dvzapo, and Mario Cifrek. A brief introduction to opencv. *2012 Proceedings of the 35th International Convention MIPRO*, pages 1725–1730, 2012.

[92] OpenCV. Getting started with videos. Online. [Accesed online] 06/04/2022. Available at: https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html.

[93] Amazon. Dewanxin webcam 1080p full hd cmos cámara web de alta micrófono reductor de ruido y corrección de automática,usb plug and play,base giratoria de 360°,para pc computadora portátil, videollamadas juegos. Online. [Accesed online] 06/04/2022. Available at: https://www.amazon.es/Micr%C3%B3fono-CorrecciC3%B3n-Autom%C3%A1tica-Computadora-Videollamadas/dp/B0891YNFQT/ref=asc_df_B0891YNFQT/?tag=googshopes-21&linkCode=df0&hvadid=435632441002&hvpos=&hvnetw=g&hvrand=13325742710400706967&hvpone=&hvptwo=&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9061040&hvtargid=pla-914054080201&psc=1&tag=&ref=&adgrpid=96472718610&hvpone=&hvptwo=&hvadid=435632441002&hvpos=&hvnetw=g&hvrand=13325742710400706967&hvqmt=&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9061040&hvtargid=pla-914054080201.

[94] K.Yogeswara Rao, Dr. Meka James Stephen, and D.Siva Phanindra. Classification based image segmentation approach. *International Journal of Computer Science And Technology*, Vol. 3, Jan. - March 2012:658–660, 2012.

[95] Manaswini Jena, Smita Mishra, and Debahuti Mishra. A survey on applications of machine learning techniques for medical image segmentation. *International Journal of Engineering and Technology*, 7:4489–4495, 2018.

[96] Joseph Walsh, Niall O' Mahony, Sean Campbell, Anderson Carvalho, Lenka Krpalkova, Gustavo Velasco-Hernandez, Suman Harapanahalli, and Daniel Riordan. Deep learning vs. traditional computer vision. 2019.

[97] Yagmur Cigdem Aktas. Image segmentation with classical computer vision-based approaches. Towards Data Science, 2021. [Accesed online] 06/04/2022. Available at: https://towardsdatascience.com/image-segmentation-with-classical-computer-vision-based-approaches-80c75d6d995f.

[98] Rajarshi Banerjee. Real time object color detection using opencv. GeeksforGeeks, 2021. [Accesed online] Last time accesed. Available at: https://www.geeksforgeeks.org/real-time-object-color-detection-using-opencv/.

[99] Jun-Dong Chang, Shyr-Shen Yu, Hong-Hao Chen, and Chwei-Shyong Tsai. Hsv-based color texture image classification using wavelet transform and motif patterns. *Journal of Computers*, 20, 2010.

[100] K. Sreedhar and B. Panlal. Enhancement of images using morphological transformation. *CoRR*, abs/1203.2514, 2012.

[101] Shivam Kumar. Difference between opening and closing in digital image processing. GeeksforGeeks, 2020. [Accesed online] 10/04/2022. Available at: https://www.geeksforgeeks.org/difference-between-opening-and-closing-in-digital-image-processing/#:~:text=Opening%20is%20a%20process%20in,then%20erosion%20operation%20is%20performed.

[102] Sourabh Kumar. Find and draw contours using opencv — python. GeeksforGeeks, 2019. [Accesed online] 10/04/2022. Available at: https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/.

[103] Contours hierarchy. OpenCV. [Accesed online] 10/04/2022. Available at: https://docs.opencv.org/3.4/d9/d8b/tutorial_py_contours_hierarchy.html.

[104] Salma Ghoneim. Object detection via color-based image segmentation using python. Towards Data Science, 2019. [Accesed online] 11/04/2022. Available at: https://towardsdatascience.com/object-detection-via-color-based-image-segmentation-using-python-e9b7c72f0e11.

[105] Detect objects of similar color using opencv in python. TechVidvan. [Accesed online] 11/04/2022. Available at: https://techvidvan.com/tutorials/detect-objects-of-similar-color-using-opencv-in-python/.

[106] Thresholding - image processing with python. Data Carpentry, 2022. [Accesed online] 12/04/2022. Available at: https://datacarpentry.org/image-processing/07-thresholding/#:~:text=Automatic%20thresholding,-The%20downside%20of&text=It%20is%20particularly%20useful%20for,background%20and%20objects%20of%20interest.

[107] What is adaptive thresholding in image processing? ICSID. [Accesed online] 13/04/2022. Available at: https://www.icsid.org/uncategorized/what-is-adaptive-thresholding-in-image-processing/.

[108] Jamil Abudhamid Mohammed Saif, Mahgoub Hammad, and Ibrahim Alqubati. Gradient based image edge detection. *International Journal of Engineering and Technology*, 8:153–156, 2016.

[109] Adrian Rosebrock. Image gradients with opencv (sobel and scharr). Pyimagesearch, 2021. [Accesed online] 13/04/2022. Available at: https://pyimagesearch.com/2021/05/12/image-gradients-with-opencv-sobel-and-scharr/.

[110] Adrian Rosebrock. Opencv edge detection (cv2.canny). Pyimagesearch, 2021. [Accesed online] 13/04/2022. Available at: https://pyimagesearch.com/2021/05/12/opencv-edge-detection-cv2-canny/.

[111] John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8:679 – 698, 1986.

[112] Mathew George and C. Lakshmi. Object detection using the canny edge detector. 2013.

[113] Luis del Valle Hernández. Detector de bordes canny cómo contar objetos con opencv y python. Blog — Visión artificial, 2018. [Accesed online] 14/04/2022. Available at: https://programarfacil.com/blog/vision-artificial/detector-de-bordes-canny-opencv/.

[114] Template matching. OpenCV. [Accesed online] 14/04/2022. Available at: https://docs.opencv.org/4.x/d4/dc6/tutorial_py_template_matching.html.

[115] Adrian Rosebrock. Multi-scale template matching using python and opencv. Pyimagesearch, 2015. [Accesed online] 14/05/2022. Available at: https://pyimagesearch.com/2015/01/26/multi-scale-template-matching-using-python-opencv/.

[116] Mrinal Tyagi. Image segmentation: Part 2. Towards Data Science, 2021. [Accesed online] 04/06/2022. Available at: https://towardsdatascience.com/image-segmentation-part-2-8959b609d268#:~:text=Region%2DBased%20Segmentation,-A%20region%20can&text=The%20similarity%20between%20pixels%20can,classified%20into%20similar%20pixel%20regions.

[117] Understanding region-based segmentation. Vision Systems Design Magazine, 1998. [Accesed online] 04/06/2022. Available at: https://www.vision-systems.com/factory/consumer-packaged-goods/article/16739413/understanding-regionbased-segmentation.

[118] Blob detection using opencv (python, c++). LearnOpenCV. [Accesed online] 04/06/2022. Available at: https://learnopencv.com/blob-detection-using-opencv-python-c/.

[119] cv::simpleblobdetector class reference. OpenCV. [Accesed online] 04/06/2022. Available at: https://docs.opencv.org/3.4/d0/d7a/classcv_1_1SimpleBlobDetector.html.

[120] J. L. Ramírez-Arias, A. Rubiano-Fonseca, and R. Jiménez-Moreno. Object recognition through artificial intelligence techniques. *Revista Facultad de Ingeniería*, 29(54), 2020.

[121] Ashish Kumar. Artificial intelligence in object detection - report. Taipei, Taiwan 10608, 2020. National Taipei University of Technology.

[122] Xin Lu, Quanquan Li, Buyu Li, and Junjie Yan. Mimicdet: Bridging the gap between one-stage and two-stage object detection. *CoRR*, abs/2009.11528, 2020.

[123] Aditya Lohia, Kalyani Kadam, Rahul Joshi, and Dr Bongale. Bibliometric analysis of one-stage and two-stage object detection. *Library Philosophy and Practice (e-journal)*, 2021.

[124] Adrian Rosebrock. Turning any cnn image classifier into an object detector with keras, tensorflow, and opencv. Pyimagesearch, 2020. [Accesed online] 22/05/2022. Available at: https://pyimagesearch.com/2020/06/22/turning-any-cnn-image-classifier-into-an-object-detector-with-keras-tensorflow-and-opencv/.

[125] Image pyramids. OpenCV. [Accesed online] 22/05/2022. Available at: https://docs.opencv.org/4.x/dc/dff/tutorial_py_pyramids.html.

[126] Adrian Rosebrock. Sliding windows for object detection with python and opencv. Pyimagesearch, 2015. [Accesed online] 22/05/2022. Available at: https://pyimagesearch.com/2015/03/23/sliding-windows-for-object-detection-with-python-and-opencv/.

[127] Adrian Rosebrock. Opencv selective search for object detection. Pyimagesearch, 2020. [Accesed online] 23/05/2022. Available at: https://pyimagesearch.com/2020/06/29/opencv-selective-search-for-object-detection/.

[128] Jasper Uijlings, K. Sande, T. Gevers, and A.W.M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104:154–171, 2013.

[129] Sambasivarao. K. Non-maximum suppression (nms). Towards Data Science, 2019. [Accesed online] 24/05/2022. Available at: https://towardsdatascience.com/non-maximum-suppression-nms-93ce178e177c.

[130] Tim Holy, Tejus Gupta, and Deepank Agrawal. Object detection using hog. JuliaImages, 2020. [Accesed online] 24/05/2022. Available at: https://juliaimages.org/ImageFeatures.jl/v0.0.3/tutorials/object_detection.html.

[131] Satya Mallick. Histogram of oriented gradients explained using opencv. LearnOpenCV, 2016. [Accesed online] 24/05/2022. Available at: https://learnopencv.com/histogram-of-oriented-gradients/.

[132] Very Engineering Team. Machine learning vs neural networks: Why it's not one or the other. Very — High-Impact IoT Product Development Company, 2018. [Accesed online] 16/06/2022. Available at: https:

//www.verypossible.com/insights/machine-learning-vs.-neural-networks#:~:text=Strictly%20speaking%2C%20a%20neural%20network,usually%20used%20in%20supervised%20learning.

[133] Saptarshi Chatterjee, Debangshu Dey, and Sugata Munshi. *Recent Trends in Computer-Aided Diagnostic Systems for Skin Diseases*. Academic Press, 1 edition, 2022.

[134] Imagefeatures — gray level co-occurrence matrix (glcm). JuliaImages. [Accesed online] 11/06/2022. Available at: https://juliaimages.org/ImageFeatures.jl/stable/tutorials/glcm/.

[135] Oscar García-Olalla, Enrique Alegre, Laura Fernández-Robles, María García-Ordás, and Diego García-Ordás. Adaptive local binary pattern with oriented standard deviation (albps) for texture classification. *EURASIP Journal on Image and Video Processing*, 2013, 2013.

[136] Adrian Rosebrock. Local binary patterns with python & opencv. Pyimagesearch, 2015. [Accesed online] 15/06/2022. Available at: https://pyimagesearch.com/2015/12/07/local-binary-patterns-with-python-opencv/.

[137] Mrinal Tyagi. Hog (histogram of oriented gradients): An overview. Towards Data Science, 2021. [Accesed online] 15/06/2022. Available at: https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f.

[138] Aishwarya Singh. A detailed guide to the powerful sift technique for image matching (with python code). Analytics Vidhya, 2019. [Accesed online] 11/06/2022. Available at: https://www.analyticsvidhya.com/blog/2019/10/detailed-guide-powerful-sift-technique-image-matching-python/#:~:text=SIFT%2C%20or%20Scale%20Invariant%20Feature,'keypoints'%20of%20the%20image.

[139] Ian London. Image classification in python with sift features. Ian London's Blog, 2016. [Accesed online] 11/06/2022. Available at: https://ianlondon.github.io/blog/how-to-sift-opencv/.

[140] Ian London. Image classification in python with visual bag of words (vbow). Ian London's Blog, 2016. [Accesed online] 11/06/2022. Available at: https://ianlondon.github.io/blog/visual-bag-of-words/.

[141] Matthew Browne and Saeed Ghidary. Convolutional neural networks for image processing: An application in robot vision. In *Lecture Notes in Computer Science*, pages 641–652, Perth, Australia, 2003. Conference: AI 2003: Advances in Artificial Intelligence, 16th Australian Conference on Artificial Intelligence.

[142] Grace Karimi. Introduction to yolo algorithm for object detection. Section, 2021. [Accesed online] 16/06/2022. Available at: https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/.

[143] Hmrishav Bandyopadhyay. Yolo: Real-time object detection explained. V7, 2022. [Accesed online] 16/06/2022. Available at: https://www.v7labs.com/blog/yolo-object-detection.

[144] Kirsten Barkved. The difference between training data vs. test data in machine learning. Obsviously.ai, 2022. [Accesed online] 16/06/2022. Available at: https://www.obviously.ai/post/the-difference-between-training-data-vs-test-data-in-machine-learning.

[145] Adrian Rosebrock. Intersection over union (iou) for object detection. Pyimagesearch, 2016. [Accesed online] 17/06/2022. Available at: https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

[146] Adrian Rosebrock. Grid search hyperparameter tuning with scikit-learn (gridsearchcv). Pyimagesearch, 2021. [Accesed online] 17/06/2022. Available at: https://pyimagesearch.com/2021/05/24/grid-search-hyperparameter-tuning-with-scikit-learn-gridsearchcv/.

[147] Augmentation. YOLOV5 Documentation. [Accesed online] 17/06/2022. Available at: https://docs.ultralytics.com/FAQ/augmentation/.

[148] Glenn Jocher. Train custom data. GitHub, 2022. [Accesed online] 17/06/2022. Available at: https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data.

[149] Aditya Mishra. Metrics to evaluate your machine learning algorithm. Towards Data Science, 2018. [Accesed online] 18/06/2022. Available at: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[150] Aditya Sharma. Mean average precision (map) using the coco evaluator. Pyimagesearch, 2022. [Accesed online] 18/06/2022. Available at: https://pyimagesearch.com/2022/05/02/mean-average-precision-map-using-the-coco-evaluator/.

[151] Vesna Stojaković. Terrestrial photogrammetry and application to modeling architectural objects. *Facta Universitatis - series : Architecture and Civil Engineering*, 6, 2008.

[152] Rajendra mayavan Sathyam. 3d position estimation of a known object using a single camera. Medium, 2021. [Accesed online] 22/04/2022. Available at: https://mayavan95.medium.com/3d-position-estimation-of-a-known-object-using-a-single-camera-7a82b37b326b.

[153] Adrian Rosebrock. Find distance from camera to object/marker using python and opencv. Pyimagesearch, 2015. [Accesed online] 22/04/2022. Available at: https://pyimagesearch.com/2015/01/19/find-distance-camera-objectmarker-using-python-opencv/.

[154] Kaustubh Sadekar and Satya Mallick. Camera calibration using opencv. LearnOpenCV, 2020. [Accesed online] 22/04/2022. Available at: https://learnopencv.com/camera-calibration-using-opencv/.

[155] Satya Mallick. Geometry of image formation. LearnOpenCV, 2020. [Accesed online] 28/04/2022. Available at: https://learnopencv.com/geometry-of-image-formation/.

[156] K. Karur, N. Sharma, C. Dharmatti, and J. Siegel. A survey of path planning algorithms for mobile robots. *Vehicles*, 3:448—468, 2021.

[157] B.K. Patle, Ganesh Babu L, Anish Pandey, D.R.K. Parhi, and A. Jagadeesh. A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4):582–606, 2019.

[158] Álvaro Sala Ayala. Diseño e implementación de un sistema de tracción basado en ros para la plataforma robótica de rehabilitación swalker. In *TFG en Ingeniería de Tecnologías y Servicios de Telecomunicación*. ETSIT UPM, 2022.

[159] Cherie Tan. Jetson nano vs raspberry pi 4: The differences. All3DP, 2021. [Accesed online] 19/06/2022. Available at: https://all3dp.com/2/raspberry-pi-vs-jetson-nano-differences/.

[160] Tienda jetson. NVIDIA. [Accesed online] 19/06/2022. Available at: https://www.nvidia.com/es-es/autonomous-machines/jetson-store/.

[161] Tarjetas microsd. PHILIPS. [Accesed online] 20/06/2022. Available at: https://www.philips.es/c-p/FM64MP45B_00/tarjetas-microsd.

[162] V. Costa, O. Ramírez, J.S. Lora-Millan, E. Urendes, E. Rocon, L. Perea, and R. Raya. Design of a robotic platform for hip fracture rehabilitation in elderly people. In *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*, pages 599–604. IEEE, 2020.

[163] Vidushi Meel. Object tracking in computer vision (complete guide). viso.ai. [Accesed online] 21/06/2022. Available at: https://viso.ai/deep-learning/object-tracking/.

Appendix A

Ethical, economic, social and environmental impact

As in any professional project, it is necessary to consider and analyse the impact that the development of this initiative has exerted on society as a whole, thus fulfilling our duty as citizens and responsible members of our communities. In order to do so, the consequences of certain aspects of the project on a number of areas, considered to be of great importance for the common good, are examined. These include ethical, socioeconomic and environmental considerations.

A.1 Introduction

TBI is a major economic, social and health challenge worldwide. The so-called "*silent pandemic*" afflicts millions of individuals every year, giving rise to a growing population of patients living with significant disabilities directly related to this disorder who struggle with basic activities of daily living, community participation and reintegration. This loss of functionality and occupational performance triggers a wide spectrum of far-reaching repercussions that range from economic burden to emotional distress and decreased quality of life among family members, aside from the natural psychosocial effects on the patients themselves including loss of self-esteem and depression.

Amongst the numerous cognitive, physical and behavioural impairments associated with this condition, many individuals with TBI consider loss of mobility to be the most significant loss of activity. Intensive physical therapy after damage to the CNS is integral to the recovery of muscle strength, which frequently limitsthese patients' self-independence. Therefore, neurorehabilitation is essential after TBI treatment, as leveraging the brain's inherent plasticity through the appropriate propioceptive stimulation during physical therapy has yielded significantly better motor function outcomes during both early and chronic stages of recovery.

The past decades have witnessed vast and rapid developments of robots for the rehabilitation of sensorimotor deficits given their ability to supply a standardised training environment, to provide adaptable support to the patient's actual state and

to increase therapy intensity and dose. As the therapy's success is in large part determined by the active physical and cognitive engagement of patients and their motivation, this Master's Thesis proposes a computer-vision based guidance system for assistive walking robots that aims to grant these devices the ability to recognise objects or people in their surroundings so that they are capable of guiding the patient's movement towards a physical target as well as providing a helping force, thus giving the user a compelling incentive to walk.

Therefore, the socioeconomic problems this initiative aims to resolve are the aforementioned repercussions that stem from TBI, which range from the loss of productivity derived from job loss to mental health issues and caregiver burden. The chosen approach to address these issues is to enhance robotic rehabilitation devices, which are already proving great effectiveness in the improvement of motor function outcomes during recovery, so that patient motivation and active participation is stimulated and, thus, mobility is increased. In turn, this will enable patients to perform daily activities, gain independence and reintegrate into their family and community lives.

A.2 Description of relevant impacts related to the project

The work developed in this Master's Thesis falls within the field of assistive robotics, which has started several ethical, social and philosophical discussions, some of which will be discussed below. This term describes a group of robots that assist individuals with physical disabilities through physical interaction, aiming to address areas and gaps in care by automating supervision, motivation and companionship aspects of one-on-one interactions with individuals from various large and growing populations.

- **Ethical impact**. This project brushes on the ongoing social debate on whether the use of robots will replace human interaction, in this case, human care. However, the robotic device employed in this particular project does not substitute medical staff, but rather facilitates the physical therapy session. There are also considerations about robots' ability to deal with moral reasoning and ethical problems. Again, since clinical staff are required to supervise the robot's functioning as well as to perform other medical expertise-requiring tasks, the robot is not expected to be left to deal with patients on its own, thus removing the need to deal with these complex scenarios. Nevertheless, the question still remains on who is responsible for the robot's actions and possible damage to the patient, specially as robots become more and more autonomous. Finally, there is the issue of privacy and data collection, of what data is collected during therapies, how and where it is stored, who has access to it and how it is used. In this particular case, the acquisition of images to train the AI models, specially if face recognition is performed, is a considerably sensitive issue.

- **Economic impact**. As discussed in the introductory chapters of this document, the economic burden attributed to this condition is enormous and has began to affect low to medium income countries the most, thus contributing

towards a greater breach in the inequalities between differently developed countries. The enhanced recovery of mobility in these patients will contribute towards the reduction in the most sizeable portion of the economic burden associated with TBI: loss of productivity. By recovering self-independence to a greater extent and in a faster manner, not only will the direct healthcare costs associated to chronic treatment of these patients be reduced, but these will also be able to return to their occupations and thus contribute towards the country's productivity and income and, thus, overall economic well-being. The issue of the assistive robots' elevated price remains, however, to ensure that everyone has accessed to the proposed rehabilitative treatment, specially in those countries possess less economical resources and where the incidence of these events is rising at the highest rate.

- **Social impact**. Disabilities are conditions that carry a great social burden as, not only do they affect the patients themselves, but also those close to them: family members, friends and informal carers. By improving the physical therapy's effectiveness and, thus, increasing mobility and self-independence, this navigation system offers great potential for an increased quality of life of both the patients and the rest of the agents involved in these situations. This could prove pivotal in terms of reducing mental health issues that result from disabilities such as low self-esteem or even depression, which can also frequently affect other areas of the patients' lives such as their personal relationships, commonly causing an increased probability of divorce. Therefore, aiding patients in living more comfortably with their conditions presents great potential for improving the quality of life of a vast amount of people surrounding the patient and, hence, for a considerable positive social impact.

- **Environmental impact**. As the material resources employed for the development of this project have been mostly *software* solutions, the environmental impact can be considered to be minimal, aside from the inevitable aspects of carbon dioxide emissions related to electricity consumption to charge the different electronic components. The use of a preexisting robotic device rather than building a new hardware system, which was the originally proposed idea, also contributes towards the low environmental impact of this project. In addition, all the electrical components which will be discarded, due to breakage or malfunction, will be properly disposed, in accordance with the relevant guidelines and applicable regulations at their corresponding recycling points.

A.3 Conclusion

The implementation of the developed navigation system based on computer vision for the purpose of neurorehabilitation will have a positive impact on the care of traumatic brain injury patients, with beneficial consequences in many areas of society and with an expected resulting improvement in the lives of citizens as a whole.

Appendix B

Economic Budget

The economic costs attributable to this Master's Thesis can be classified into two main categories, those related to the type of resources employed, which are **human** resources and **material** resources. The expenses associated to both of these divisions are quantified in Table B.1 and Table B.2, respectively, which will both be exploited to calculate a total budget for the prototype developed over the course of this project's duration.

- **Human resouces**: a total of three individuals have been involved in the development of this project, including: an engineering student, a project manager and a technical advisor (see Table B.1).

	Coste horario (€)	**Horas**	**Total (€)**
Project manager	30	30	900
Technical advisor	30	15	450
Engineering student	20	540	10,800
TOTAL			**12,150**

Table B.1: Economic budget for human resources.

- **Material resources**: this item considers the costs of the materials used for the development of the prototype (see Table B.2).

	Lifespan (years)	**Units**	**Cost (€)**	**Amortisation (€/month)**	**Time used (months)**	**Total (€)**
USB camera	5	1	18	0.30	4	1.20
Jetson Nano	4	1	48	1.00	4	4.00
Micro SD card	1	1	12	1.00	4	4.00
Laptop	6	1	800	11.11	4	44.44
TOTAL						**53.64**

Table B.2: Economic budget for material resources.

As it can be seen, the most notable material cost is that of the computer, with the rest of the components being of a much lower price. Taking into account both human and material resource costs and the pertinent taxes, the final economical budget ascends to 14,766.40€, as displayed in Table B.3. As in any professional project, the human cost far exceeds the material cost, with the latter not even reaching a mere 1% of the former's magnitude.

	Coste
Costes de personal	12,150.00€
Costes de material	53.64€
Subtotal	12,203.64€
IVA	2.562,76€
Total	14,766.40€

Table B.3: Total costs.