

**T.C.**  
**ISTANBUL SABAHATTIN ZAIM UNIVERSITY**  
**GRADUATE EDUCATION INSTITUTE**  
**DEPARTMENT OF COMPUTER ENGINEERING**

**INTEGRATING COMPUTING VISION WITH DEEP  
LEARNING FOR AUTONOMOUS ROBOT  
NAVIGATION**

**Ph.D.DISSERTATION**

**Walead Kaled SLEAMAN**

**Istanbul**  
**April, 2023**

**T.C.  
ISTANBUL SABAHATTIN ZAIM UNIVERSITY  
GRADUATE EDUCATION INSTITUTE  
DEPARTMENT OF COMPUTER ENGINEERING**

**INTEGRATING COMPUTING VISION WITH DEEP LEARNING  
FOR AUTONOMOUS ROBOT NAVIGATION**

**Ph.D. DISSERTATION**

**Walead Kaled SLEAMAN**

**Supervisor**

**Asst. Prof. Dr. Ali HAMITOĞLU**

**Istanbul**

**April, 2023**

## THESIS APPROVAL

This study has been approved in partial fulfillment of the requirements for Ph.D. Degree in Computer Engineering

Chairperson of jury Assist. Prof. Dr. Ali HAMİTOĞLU

Member of jury Assist. Prof. Dr. Şengül BAYRAK HAYTA

Member of jury Assist. Prof. Dr. Mohammed VADİ

Member of jury Assist. Prof. Dr. Amani YUSUF

Member of jury Assist. Prof. Dr. Muhammed DAVUD

Approval by

Assoc. Prof. Dr. Erhan İÇENER  
Director, Graduate Education Institute

## **DECLARATION OF SCIENTIFIC ETHICS AND ORIGINALITY**

This is to certify that this MA thesis/ PhD dissertation titled “integrating computing vision with deep learning for autonomous robot navigation.” is my own work and I have acted according to scientific ethics and academic rules while producing it. I have collected and used all information and data according to scientific ethics and guidelines on thesis writing of Sabahattin Zaim University. I have fully referenced, in both the text and bibliography, all direct and indirect quotations and all sources I have used in this work.

Signature

Walead SLEAMAN

April, 2023

## **ACKNOWLEDGEMENT**

First, I thank God, who gave me the power to complete my thesis, and thank him anyway. Then my thanks to my family, who endured patiently and gave me their full support, especially my dear wife who employed all her effort to get my study done in the best way. So is my mother, who gave me strength with her prayer. My thanks go to all the University, the College, and the Department staff who played a role in completing this task, particularly the supervisors of Assist. Prof. Dr. Alaa Ali Hameed for his patience, valuable guidance, and Asst. Prof. Dr. Muhammed Davud and the other Department of Computer Engineering professors.

This study will have applications in the service of science and humanity, especially in alleviating the suffering of patients and in the health field. It will be the reward of this work for my father. May God have mercy on him.

**Walead SLEAMAN**

## ABSTRACT

# INTEGRATING COMPUTING VISION WITH DEEP LEARNING FOR AUTONOMOUS ROBOT NAVIGATION

Walead Kaled SLEAMAN

PhD. Dissertation, Computer Science and Engineering (30% English)

Supervisor: Assist. Prof. Dr. Ali HAMİTOĞLU

April- 2023, 112 + xv Pages

Independent robot navigation needs to provide enormous information about the environment and maps, use high-resolution and cost sensors, and the allocation of software to characterize each task that the robot performs while navigating, another difficult task. As a result, such robots are highly expensive, can navigate in specific environments, and must have sufficient information about them. To overcome these difficulties and determinants of the cost of building robots, this study combines computer vision with the internal analysis algorithm of mobile robots using a hierarchical structure that combines different layers of the complex neural network. (CNN). Moreover, the decision-making process enables the robot to navigate independently in an indoor environment. It is done by training the overall system holistically on a new dataset built by a depth camera Low-cost that relies on its computer vision system mechanism and a manually controlled mobile robot (HCMR) built for this purpose. The output consists of teaching a proposed model of the critical moving trends that the robot uses to achieve independent analytical capability. It is done by moving this robot in real and diverse environments. The robot is trained using this data using CNN, which presents the outcome of the environmental investigation (control labels) that enables the robot to self-navigate without a map in an unknown environment. Furthermore, extensive indoor trials were carried out and received 77% accuracy. A Comparison of the current results with previous studies. This study concluded that the proposed model could reach better results than previous studies using an expensive sensor. In addition, another comparison was performed between the human-controlled robot and the trained robot using the deep learning process to determine decisions to control the robot's movement.

**Keywords:** Robot exploration, deep learning, CNN and a single-eye camera stereo system.

## ÖZET

# OTONOM ROBOT NAVİGASYONU İÇİN BİLGİSAYAR VİZYONUNU DERİN ÖĞRENME İLE ENTEGRE ETMEK

Walead Kaled SLEAMAN

Doktora, Bilgisayar Bilimleri ve Mühendisliği (%30 İngilizce)

Tez Danışmanı: Dr. Öğr. Üyesi Ali HAMİTOĞLU

Nisan- 2023, 112 + xv Sayfa

Mobil robotların daha önce bilinmeyen ortamı keşfetmesini sağlamak, özellikle kapalı bir ortamda gezinirken zor bir işti. Bağımsız robot navigasyonunun, çevre ve haritalar hakkında çok büyük bilgilerin yanı sıra yüksek çözünürlüklü ve maliyet sensörlerinin kullanımı ve robotun navigasyon sırasında gerçekleştirdiği her görevi karakterize etmek için yazılım tahsisi sağlaması gerekir; bu da başka bir zor görevdir. Sonuç olarak, bu tür robotlar oldukça pahalıdır ve belirli ortamlarda gezinme yeteneğine sahiptir ve onlar hakkında yeterli bilgiye sahip olmalıdır. Bu zorlukların ve belirleyicilerin üstesinden gelmek ve robot inşa etme maliyetini azaltmak için, bu çalışma, kıvrık sinir ağının farklı katmanlarını birleştiren hiyerarşik bir yapı kullanarak mobil robotların dahili analiz algoritması ile bilgisayar vizyonunu birleştirerek çalışır. (CNN) ve robotun kapalı bir ortamda ve bağımsız olarak gezinmesini sağlayan karar verme süreci. Bunun, bir derinlik kamerası tarafından oluşturulan yeni bir veri kümesi üzerinde tüm sistemi bütünsel bir şekilde eğiterek yapıldığı durumlarda Bilgisayarlı görüş sistemi mekanizmasına dayanan düşük maliyetli ve bu amaç için oluşturulmuş manuel olarak kontrol edilen mobil robot (HCMR). Çıktı, robotun bağımsız analitik yetenek elde etmek için kullandığı kritik hareket eğilimlerinin önerilen bir modelini öğretmekten oluşur. Bu, bu robotu gerçek ve çeşitli ortamlarda hareket ettirerek yapılır. Robot daha sonra bu veriler kullanılarak CNN kullanılarak eğitilir ve bu, robotun bilinmeyen bir ortamda bir harita olmadan kendi kendine gezinmesini sağlayan çevresel araştırmanın (kontrol etiketleri) sonucunu sunar. Ayrıca, kapsamlı iç mekan denemeleri yapıldı ve %77 doğruluk elde edildi. Daha önceki çalışmalarla da kapsamlı bir kontrast yapılmıştır. Bu çalışma, önerilen modelin pahalı bir sensör kullanarak önceki çalışmalarda elde edilenlerden daha iyi sonuçlara ulaşabildiği sonucuna varmıştır. Ayrıca robotun hareketini kontrol etmeye yönelik kararları belirlemek için insan kontrollü robot ile derin öğrenme sürecini kullanan eğitimli robot arasında başka bir karşılaştırma yapılmıştır.

**Anahtar Kelimeler:** Robot keşfi, derin öğrenme, CNN ve tek gözlü kamera stereo sistemi.

## TABLE OF CONTENTS

<b>THESIS APPROVAL</b>	<b>i</b>
<b>DECLARATION OF SCIENTIFIC ETHICS AND ORIGINALITY</b>	<b>ii</b>
<b>ACKNOWLEDGEMENT</b>	<b>iii</b>
<b>ABSTRACT</b>	<b>iv</b>
<b>ÖZET</b>	<b>v</b>
<b>TABLE OF CONTENTS</b>	<b>vi</b>
<b>LIST OF TABLES</b>	<b>x</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF SYMBOLS</b>	<b>xiv</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xv</b>
<b>CHAPTER I</b>	
<b>INTRODUCTION</b>	<b>1</b>
1.1. Motivation	2
1.2. Computer Vision	2
1.3. Definition of Autonomous Navigation	3
1.4. Autonomous Navigation	4
1.5. Robot Learning	5
1.6. Problem Statement	6
1.7. The Rationale of the Study	7
1.8. Thesis Structure	8
<b>CHAPTER II</b>	
<b>LITERATURE REVIEW</b>	<b>9</b>
2.1. Mobile robots and addressing their challenges using deep learning	9
2.2. Previous Studies	10
<b>CHAPTER III</b>	
<b>RESEARCH METHODOLOGY</b>	<b>14</b>
3.1. Methodology	14

3.1.1. Computer Vision and Camera Calibration Methods -----	14
3.1.2. Stereo System and Depth Image -----	15
3.1.3. The calibration and Camera Model -----	16
3.1.4. Major types of distortion effects and their cause -----	22
3.1.5. Semi-global block Matching Function (SGBM)-----	24
3.2. Depth Sensing-----	25
3.3. Deep Learning -----	26
3.4. Deep Learning for Decision Making -----	27
3.5 .Training -----	28
3.6. Convolutional Neural Networks-----	29
3.7. Mobile Robot-----	30
3.7.1 Wheeled mobile robots-----	31
3.8. Control and Motion of Mobile Robot-----	34
3.9. Embedded System for DC Motor Control -----	36
3.9.1. DC motor characteristics. -----	36
3.9.2. DC motor speed control technique. -----	37
3.9.3. Energy consumption -----	38
<b>CHAPTER IV</b>	
<b>FINDINGS &amp; DISCUSSIONS -----</b>	<b>40</b>
<b>Design and Implementing the Depth Image, HCMR, CNN -----</b>	<b>40</b>
4.1. The Steps Build Depth Sensors, HCMR, Dataset and Training the Model Using CNN.-----	40
4.2. Visual Sensor Structures-----	43
4.3. Depth image design -----	45
4.4. Building Hand Controller Mobile Robot (HCMR): -----	46
4.4.1. The Hardware Parts -----	46
4.4.1.1. The Main Base of HCMR-----	46
4.4.1.2. Sensor and Vision System: -----	46
4.4.1.3 .Motion Axes in the Robot -----	47

4.4.1.4. The Micro-Controller Unit-----	47
4.4.1.5. Power Unit in Robot-----	49
4.4.1.6. The DC1 and DC2 Engines-----	50
4.4.1.7. Joystick and Control Key -----	51
4.4.2. Software parts-----	51
4.4.2.1 .The software for creating depth images -----	52
3.4.2.2. Software Controller to HCMR -----	53
4.4.2.3. Depth Image Recording Software and labels Control -----	54
4.4.2.4. Joystick Key Identification -----	54
4.4.2.5. The Main Program That Manages All Program -----	54
4.5. Building a New Dataset-----	56
4.6. The Deep learning Techniques -----	57
4.6.1. How DL Works-----	61
4.6.2. The Convolutional Neural Network (CNN)-----	62
4.6.3 The CNN Architectural Design -----	63
4.6.4. Activation Function (AF) -----	66
3.6.5. Pooling Layer -----	69
3.6.6. Fully Connected Layer -----	70
4.6.7. The dropout method in network -----	71
4.6.8. Loss Layer-----	72
4.7. Training Convolution Neural Networks-----	74
<b>CHAPTER V</b>	
<b>FINDINGS &amp; DISCUSSION -----</b>	<b>76</b>
5.1. The Low Cost Sensors to Create Depth Images-----	76
5.2. Prepared by the New Data Collection -----	78
5.3. Modular Training Exercises-----	81
5.4. Comparison of Human Behavior and Acquired Behavior through CNN-----	84
<b>CHAPTER VI</b>	
<b>CONCLUSION AND SUGGESTIONS FOR FUTURE WORK -----</b>	<b>86</b>

6.1. Conclusion-----	86
6.2. Suggestions for Future Work -----	87
<b>REFERENCES-----</b>	<b>89</b>
<b>APPENDIX 1-----</b>	<b>98</b>
<b>APPENDIX 2-----</b>	<b>99</b>
<b>APPENDIX 3-----</b>	<b>100</b>
<b>APPENDIX 4-----</b>	<b>101</b>
<b>APPENDIX 5-----</b>	<b>102</b>
<b>APPENDIX 6-----</b>	<b>103</b>
<b>APPENDIX 7-----</b>	<b>104</b>
<b>APPENDIX 8-----</b>	<b>105</b>
<b>APPENDIX 9-----</b>	<b>106</b>
<b>APPENDIX 10-----</b>	<b>107</b>
<b>APPENDIX 11-----</b>	<b>108</b>
<b>APPENDIX 12-----</b>	<b>109</b>
<b>APPENDIX 13-----</b>	<b>110</b>
<b>APPENDIX 14-----</b>	<b>111</b>
<b>CURRICULUM VITAE-----</b>	<b>112</b>

## LIST OF TABLES

Table 3. 1 The Power Breakdown Of Pioneer Mobile Robot (Y. Mei, 2005:492-497) .....	38
Table 5. 1 Represent How Dataset Is Separate To Parts With Work It (Researcher's Work) .....	79
Table 5. 2 Represents Depth Images Configured From Five Environments by HCMR (Researcher's Work) .....	80



## LIST OF FIGURES

### CHAPTER III

Figure 3. 1 The Step Diagram of Camera Calibration Method (S. Praveen, 2020:113).....	17
Figure 3. 2 The Steps Diagram of Camera Calibration Method (Researcher's Work).....	17
Figure 3. 3 The World Coordinate System and The Camera Coordinate System (Researcher's Work).....	18
Figure 3. 4 The Pinhole Camera Model(S Nedevschi,2002:4), (Researcher's Work).....	21
Figure 3. 5 Image Explaining The Effect Of Barrel Distortion And Pincushion Distortion (K.Sadekar.2020:3.).....	23
Figure 3. 6 The Effect Of Lens Distortion.(Left) And Image After Un-Distortion (Right). (K.Sadekar.2020:4).....	23
Figure 3. 7 Reference Control Scheme for Mobile Robot Systems (R. Siegwart, 2011:10).....	31
Figure 3. 8 System Architecture of (A) The Traditional Map-Based Navigation Control System (B) The Navigation System Based Into DI Without Maps To Control System. (K. Zhu,2021:399).....	32
Figure 3. 9 Idealized Rolling Wheel (G. Mester, 2006:3).....	34
Figure 3. 10 Dynamic Parameters Of A Vehicle In Motion (G. Mester, 2006:3).....	36
Figure 3. 11dc Motor Characteristic Diagram (Researcher's Work).....	38
Figure 3. 12 Pwm Signal of Every Run Cycles (Researcher's Work).....	38

### CHAPTER IV

Figure 4. 1information Obtained From Calibration Apply. (Researcher's Work).....	44
Figure 4. 3 Micro-Controller Circuit Arduino Uno (Researcher's Work)).....	47
Figure 4. 4 The Device Driver L293d With Arduino Uno(R. Farkh,2021:423-235). 48	
Figure 4. 5 Power Supply Unit of Robot Hcmr (Researcher's Work).....	49
Figure 4. 6 The Control Unit Between Pc And Another Device In Mobile Robot (Researcher's Work).....	50

Figure 4. 7 The Keys Used to Control The Robot Hcmr In Joystick of The Ipad 4 (Researcher's Work).....	51
Figure 4. 8 The Control Program In Computer And Arduino (Researcher's Work).	53
Figure 4. 9 Outline Describing Major Programmed And Robot Control Subprograms (Researcher's Work).....	55
Figure 4. 10 Compared To Traditional Ml In B(A) And Dnn In (B) (M. A. Wani2020:3).....	58
Figure 4. 11 Relationship Between Ai, Machine Learning, And Dl (M. A. Wani,2020:6).....	60
Figure 4. 12 Convolution Operation(M. A. Wani2020:19) .....	65
Figure 4. 13 Padding Layer (Researcher's Work) .....	66
Figure 4. 14 Graph Of Sigmoid Activation Function(M. A. Wani2020:23).....	67
Figure 4. 15 Graph Of Tanh Activation Function(M. A. Wani2020:24).....	68
Figure 4. 16 Rectified Linear Unit (Relu) Activation Function (Researcher's Work)	69
Figure 4. 17 The Swish Activation Function(M. A. Wani2020:25) .....	69
Figure 4. 18 Max And Average –Pooling (M. A. Wani2020:26) .....	70
Figure 4. 19 Connection Between Convolution Layer And Fully Connected Layer (W. Njima , 2019:8-26).....	71
Figure 4. 20 Show The Dropping In Network(M. A. Wani2020:28) .....	72
Figure 4. 21 The Proposed Model Use CNN (Researcher's Work) .....	74

## CHAPTER V

Figure 5. 1 Shows Processes Performed To Create Depth Images (Researcher's Work).....	78
Figure 5. 2 The New Dataset Of Storage Of Depth Images And The Control Label (Researcher's Work).....	78
Figure 5. 3 Confusion Matrix On The Predicative Values, The Diameter Of The Matrix Indicates The Accuracy Of Inference	81
Figure 5. 4 Shows The Relationship Between Epoch And Loss Using The New Dataset	83
Figure 5. 5 The Relationship Between Epoch And Loss Using The Dataset On Previous Studies	83

Figure 5. 6 The Compared Between Training In New Dataset And Dataset Use

Microsoft Kinect Sensor.

84

Figure 5. 7 (a) Decisions made by humans and robots compared. A total of 500 points from the human and robot judgment curves have been gather. The average absolute distinction between the two examples is calculated. A median disparity used in the first case is 0.1210 rad/s (a). In the second situation, the value to median disparity 0.1400 rad/s (b)..... 85



## LIST OF SYMBOLS

$(X_w, Y_w, Z_w)$	: World coordinate system
$(X_c, Y_c, Z_c)$	: Camera coordinate system
$N_p$	: Group point use in matching block
$Q_r(t)$	: The rotation of angle in rolling wheel
$Q(t)$	: Angle in very time
$\dot{x}$	: The first Derivative of variable x
$\dot{y}$	: The first Derivative of variable y
$\dot{\theta}$	: The first Derivative of angle $\theta$
$\sigma(x_j)$	: Sigmoid function
$[R_x, R_y, R_z]^T$	: Projection of 3D point of coordinate $P(x, y, z)$
$T_{wc}$	: The camera to world translation vector
$R_{wc}$	: World to camera rotation matrix
$X_{wc}$	: World to camera translation vector
3D	: Three dimension image
2D	: Two dimension image
$W_r$	: The angler velocity to the right
$W_l$	: The angler velocity to the left

## LIST OF ABBREVIATIONS

AGA	: Automated Guided Vehicles
AF	: Activation Function
AI	: Artificial Intelligent
CNN	: Convolution Neural Network
AUV	: Autonomous Underwater Vehicles
PP2	: Personal Robot2
BRETT	: Elimination of Tedious Tasks
IoT	: Internet of Things
LFD	: Learning From Demo
RL	: Reference Learning
LiDAR	: Light Detection And Ranging
PCA	: Principal Component Analysis
ReLU	: Rectified Linear Unit
MCL	: Monte Carlo Localization
ORB-SLAM	: Orb- Simultaneous Localization And Mapping
Q-learning	: Model-Free Reinforcement Learning
Open-CV	: Open Source Computer Vision library
FCL	: Fully Connection Layer
FPGA	: Field Programmable Gate Arrays
CPU	: Center Processor Unit
SGBM	: Semi-Global Block Matching
SVHN	: Street View House Numbers
GPU	: Graphic Processor Unit
DC	: Direct Current.
MDP	: Markov Decision Process
ILSVRC	: Image Net Large Scale Visual Recognition Challenge
VGG	: Is CNN Architecture
PWM	: Pulse Width Modulation
HCMR	: Hand Controller Mobile Robot

# CHAPTER I

## INTRODUCTION

Nowadays, mobile robots are one of the fields of scientific study that is increasing at the highest rate. Because of their abilities, mobile robots can replace humans in many fields, including applications in space exploration, healthcare, industrial, construction, personal services, work in harsh environments, and numerous other industrial and non-industrial applications (F. Rubio, 2019:1-3).

Comparing guided motor vehicles are described as computer-controlled loading carriers moving through the wheels on which they are the base for horizontal transport without the need for an onboard operator. (Lin, 2006:17), in recent years, technology in material processing has progressed rapidly. One important development is development that has been accomplished on automated guided vehicles (AGVs) and converted into mobile robots operating through self-controller (AMRs) (Moser, 2022:4-31).

The robot is independent when it can determine what actions to take to perform a task using a cognitive system. It also needs a perception module or control system to coordinate all the subsystems that control the robot. The fundamentals of mobile robots consist of areas of movement, perception and navigation (Siegwart, 2004:0-262). Understanding the mechanism, engine, dynamics and control theory helps address the challenges of mobile robot movement through the perception of signal analysis and sensor technologies. The perception analyses data sent from sensors and appropriate actions taken to achieve mobile robot targets.

It is done through navigation and knowledge of planning algorithms, information theory and artificial intelligence. All motion, perception and localization systems must work in an integrated manner with a controller to accomplish the mobile robot mission coherently. (Kececi. EF., 2015:182). CNN's use of descriptive power to appreciate and find a mobile robot in an indoor location has recently gained popularity in deep learning networks. The results have been very convincing, especially in localization (S. P. P. da Silva, 2020: 12040-12048). A new approach has also been proposed by designing 3D movements of moving objects, along with the camera. It can adapt to

new environments by learning to improve frames online. It is done through a tentative way of handling motion and restricting the object's size. Through effectively delivering learning from highly dynamic scenes using a monocular environment. (V.Casser, 2021: 8001-8008), an imaging viewpoint was created, where the authors did a fixed camera and object lighting. (H. Kraft,2004: 45-49). The mobile robot can also move in different other environments. A small robot by observing the oceans and underwater environments, suitable for a fixed point or a small scale and for the areas of the incision and groove underwater through which the oceans are observed in its study (Lingshuai, 2016:19-22).

### **1.1.Motivation**

A mobile robot is an independent agent who can navigate smartly anywhere using sensor and operator control techniques. Industry, space, defence, transportation, and other societal sectors have been affected by the rise of autonomous mobile robot applications. Furthermore, navigating between different points in a given environment is one of the most fundamental tasks in almost every automated system.

### **1.2.Computer Vision**

Computer vision is an area of artificial intelligence in which computers extract valuable data from digital photos, videos, and other visual inputs through operations. It is accomplished by examining and covering all of the functions carried out by biological vision systems. Understanding what is seen is Extracting complex data into a procedure that may be employed in other processes by "seeing" or "feeling" and visual stimulation. They create an interdisciplinary field and organize components of human vision systems employing sensors, computers, and machine learning algorithms through these processes. These Characteristics of disciplinary field models use computer sensors and machine learning methods to organize components of human vision systems. Continuous reading of computer vision takes numerous shapes and examples because, Self-driving vehicles, for example, must gather information about their surroundings to establish how they should act, as well as facial recognition. For image searches, computer vision is viewed as a powerful security tool.

Many applications of data vision theory include detecting objects in photographs, searching image databases, and extracting data from images. Most robotic devices in

manufacturing must be able to detect their surroundings to fulfil their tasks. In addition, when it comes to mimicking human vision (Maysa Abdullah, 2019:330). Consider how vivid the 3D structure would be if we sat in a vase on a table next to one another. We can discuss the shape and transparency of each portion of the sophisticated forms of light, the figure (A) in Appendix 7 depicts the natural separation of each flower from the aeon, as well as the shadow that plays on its surface. We can count (and name) all the persons in a closed group photo by looking at it. Furthermore, despite their propensity to create optical illusions, cognitive psychologists have spent decades trying to comprehend how the visual system works and to articulate some of its principles in the form shown in Figure B in Appendix 7. Researchers have created modified mathematical methods in computer vision that improve the look and feel of three-dimensional photographic objects. Calculating an ecological model can be done in many trustworthy and accurate ways. Figure (B) in Appendix 7 shows a finely three-dimensional surface comprising hundreds of partially overlapping pictures. a. We may use stereo maps to create accurate models of a dense three-dimensional surface, as illustrated in Figure (B. 1) in Appendix 7. We can also use Figure (B. 2) in Appendix 7 to follow a person, moving against or dining with him. In addition, as illustrated in Figure (B. 3) in Appendix 7, we can monitor a person travelling unusually. We may even be able to, with some success, Attempt to identify and name all of the people in a photograph very well, as illustrated in Figure B.4 in Appendix 7, employing a combination of detection, picture and facial recognition, clothing, and hair (Maysa Abdullah, 2019: 330).

Despite these accomplishments, the ambition of having a computer still interprets images at the level of a two-year-old child (for example, counting all the animals in the picture). It still needs to be clarified. Modelling the visual world with its rich richness, on the other hand, is far more challenging than utilizing a medium. Physics is frequently used in advanced computer vision models (radiometry, optical line). Both fields represent how objects move, how light is reflected off their surfaces and diffused by the atmosphere, and how this varies as camera lenses are used (or human eyes).

### **1.3.Definition of Autonomous Navigation**

Autonomous mobility refers to a robot's ability to travel, plan its path, and execute its strategy without human involvement. In certain circumstances, remote navigation aids

employed in the planning process, particularly when the robot used in an outdoor setting, however in other cases, the only information available to compute the path input from sensors on board the robot, which is done in indoor environments. An autonomous robot is capable of not just maintaining stability while moving, but also planning its moves. When accessible, autonomous robots can travel in various ways, although they may rely on information from optical, aural, and olfactory sensors. When basic location data is acquired in the form of control signals or visual representations of the surroundings, machine intelligence is necessary for some basic motivation to translate into a path and action plan (O. Mohamed, 2021:1-10).

#### **1.4. Autonomous Navigation**

A completely autonomous robot can gather environmental data, operate for long periods without human involvement, travel in whole or in part through its working area, and avoid potentially dangerous circumstances for people and property. An autonomous robot can also learn and acquire new skills, such as adapting to new task methods or changing circumstances. As a result, mobile robots must have autonomy and intelligence, and algorithms must be developed that allow robots to function autonomously in unstructured, dynamic, partially observable, and unreliable situations. The researchers have been charged with solving important challenges including uncertainty (in both perception and action), Reliability and real-time responsiveness are two important factors to consider. It makes no sense to navigate without the flexibility of mobility in any of these mobile bot apps. Except for most scientific or industrial applications of mobile robots, which require flexibility for intentional movement, random movement that does not require the capacity to move can be beneficial in monitoring or cleaning operations. As a result, autonomous mobility is critical to robotics' development and serves as a model for related autonomous mobile robotics technologies. Furthermore, directing a moving robot entails creating a path that avoids obstacles and leads to a certain target based on sensor readings. In general, robotic navigation entails the following six connected skills:

1. Perception is the process of receiving and processing sensory data.
2. Search: the mechanism that leads the robot's movement in the environment to the next task.

3. Mapping: The process of altering a spatial representation or model of the environment by incorporating sensory data.
4. Localization: These long-term operations estimate the robot's position inside a spatial map while also controlling navigation.
5. Path planning: This strategy works to find a path toward the target site or not.
6. Path execution: detecting kinetic movements and modifying them to changes in the environment, including obstacle avoidance.

The robot requires a system that allows it to move freely in its surroundings, as well as the ability to detect and respond to unforeseen situations. The robots' sensors fulfill the same function as the eyes of the robots. The robot can tell you where it is, how to get there, and where it has gone. Sensors that detect the distance traveled by the wheels on the bottom to measure internal and external environmental changes are frequently flexible and mobile (R. Szeliski, 2010:181-196). To summarize, sensors can loosely be classified into two groups: Internal state sensors, such as accelerometers and gyroscopes, offer internal data on the robot's movements, and external state sensors, such as lasers and infrared sensors. Sonar and optical sensors that provide external information about the environment.

The robot's position in 2D space can be estimated using data from internal status sensors. External sensor data used directly determine an area or position, or to turn it into information on an environmental map. Because the signal sent by the sensor is distorted in most circumstances, it is inaccurate. As a result, processing the data supplied by the sensor, which contains noise, is critical for mobile robotic navigation. Neural networks overcome this difficulty because they have many processing nodes, each with primarily local connections, which can give a stable point.

### **1.5. Robot Learning**

Many machine learning approaches, including reinforcement learning, simulated learning, inverse reinforcement learning, and regression methods, can adapt correctly, allowing complicated robotic systems to be studied. Helicopters, wing flying, anthropomorphic legs and limbs, and humanoid robots are only a few examples. On the other hand, traditional AI-based robotic systems frequently attempt to construct a set of rules and models by hand that permit and allow a robot to sense and act in the

real environment. Learning robots presupposes that we cannot properly forecast all of life's intriguing circumstances. Therefore concentrate our efforts on managing mobile robotics learning through simulation, although robotics education encompasses various topics, including learning, perception, planning, decision-making, and more. Poses that we are unable properly forecast all of life's intriguing circumstances. Two general families of learning policy algorithms mentioned extended learning (RL) and learning by example (LfD). Damage decision issues have been simplified (Cobo, 2014: 216). Control of underwater vehicles (AUVs) (Carreras, 2005:30), navigation of mobile robots (Jaradat, 201:135-1491)/ (Riedmiller, 2009: 55-73).

Several intriguing applications have recently surfaced. Berkeley Robot for the Elimination of Tedious Jobs (BRETT), a Willow Garage Personal Robot 2 (PR2), has been empowered to learn to accomplish various tasks independently through trial and error without pre-programmed details about its environment. As shown in Appendix 8 Figure (A.1), these tasks include putting a wheel on a toy aeroplane, stacking a Lego block, and screwing a cap on a water bottle (Mülling, 2013: 263-279) employed imitation and reinforcement learning approaches to teaching a Barrett WAM arm how to hit a table tennis ball successfully. The researcher taught a robot how to make a pancake flip. Appendix 8 Figure (A1. 2) shows other successful robot learning applications.

## **1.6. Problem Statement**

The rapid and growing development of autonomous mobile robotics applications in our daily lives and their use in different environments has created the problem of finding autonomous mobile robots based on users' requirements and low cost. It easily travels in complex and unknown environments.

The main core of this project is a study and evaluate the latest deep-learning approaches to automated navigation that have recently been proposed in unknown internal environments. After studying the implementation of each approach and the advantages and disadvantages of these algorithms, we were able to formulate a new approach by combining the deep learning mechanism with computer vision to achieve the requirements of directing and controlling an independent mobile robot for navigation in an indoor environment without using maps and information on the environment at low cost. It is done using low-cost image sensors.

This type of robot is used in various tasks such as healthcare, in-home services, businesses, and banks. Our approach is to represent a convoluted neural network (CNN) that can safely drive the mobile robot in the indoor environment such as homes, hospitals and others. The approach could be mainstream to operate in dynamic external and internal environments. However, this approach also needs more target-oriented movement. This thesis aims at low-cost autonomous mobile automation in indoor environments, where studies confirmed this area (C.Xia, 2015:38-44).

### **1.7.The Rationale of the Study**

This study aims to combine computer vision and deep learning to create a mobile robot that can navigate independently at low cost and without using maps and information of the internal environment in which it operates through- :

1. Building low-cost sensors to create depth images using optical sensors (web camera) and stereo systems based on the computer vision system.
- 2 .Building a hand- controller mobile robot (HCMB) that uses depth sensors into (1) its function to build a new dataset to train the proposed model.
- 3 .Building the new training data set to train the proposed model using mobile robot into(2)
4. Building a proposed model enable to training and learning based on CNN through using the new data set created into (3) and works independently on navigation, avoiding obstacles and simulating human behaviour in learning, movement from one place to another without using maps and information about surround environment.

The system presented in this study is designed to simplify the task of creating meaningful navigation controls for a robot to follow. A higher level of control form feeds the system the desired direction, such as the direction of a target object tracked with an RGB-D camera, inputting a series of depth images to produce Robot movement commands that result in safe and efficient travel towards the desired position. We show that in Figure B in Appendix 12.

## **1.8.Thesis Structure**

This thesis is organized as follows: Chapter 1 presents the motives of the thesis, the definition of independent navigation, the problem statement, the objectives of the thesis and the structure of the thesis in Chapter 2 related work. Chapter 3 Mythology Chapter 4 The learning of mobile robots to navigate an unknown environment using CNN, Chapter 5 System evaluation and experimental results. Chapter 6 Conclusion and suggestions for future work.



## **CHAPTER II**

### **LITERATURE REVIEW**

In this section, previous and current studies in the field of mobile robots of all kinds, especially independent ones that operate in internal and external environments, and the challenges they face are reviewed and addressed. We offer studies that used deep learning or traditional methods have used. In addition, we offer studies using the various sensors supplied to these robots to accomplish the independent navigation mission.

#### **2.1. Mobile robots and addressing their challenges using deep learning**

The increasing demand for autonomous mobile robots has prompted researchers to improve these platforms' automation technologies. Their navigation system is one of the most fundamental technologies for automating these platforms.(Guang, 2018:2953-2964). Autonomous mobile robot navigation is an active area of research that has received increasing interest from many researchers recently (Onyekpe,2021:1270-1292). Autonomous Mobile Robots (AMR), for example, are created and reprogrammed to perform various tasks without human interaction. Usually, it is difficult for humans to perform such tasks for various reasons, such as bad environmental conditions or unhealthy and dangerous environments, especially when epidemics spread, or even repetitive tasks that can be boring. Because of these conditions and the necessity for using mobile robots in difficult environments, there has been much research in mobile robotics in recent years. The intention is to have such devices which can perform most tasks automatically without human intervention (Foroughi, 2015:116,1-6). All the advantages can be achieved through this type of robotics, but at the same time, many challenges also come. Localization and mobility are major challenges in the autonomous system of these types of robots. Deep learning is a subsection of machine learning that has proven successful in various applications in domains such as robotics. This success and development in machine learning have led researchers to use deep learning approaches to investigate the localization of autonomous mobile robots in indoor environments. Most of these methods are based on vision-based localization, usually executed as an image detection looking for the

most relevant image in the database with a query image by matching features (Kim, 2015:21636-21659)/(Sattler, 2016:1744-1756). Based on the deep learning approach, many methods have been successfully applied to solve the localization problem. However, this approach has limitations due to the limited size of private datasets. Deep learning-based models have more parameters, and training them using insufficient or small datasets can increase over-composition. Although this problem can be largely solved using transformational learning and weights gained from already-trained models, the model still needs to improve accuracy due to its reduced capabilities in generalizing trained models. The interest in implementing robotic systems has increased for tasks such as internal automation, driverless transport represented by autonomous mobile robots, exploring the wildly unknown environment between the research community and engineers' pre-loaded maps, Etc. Navigating the environment. Available stereo cameras that compose expensive depth images require special engines and software to interact with processing platforms. Moreover, to solve this problem in this thesis by making stereo devices using regular web cameras and creating depth images after working on calibrating these cameras, which makes this technology cost-effective.

## **2.2. Previous Studies**

Proposed a method for image correction through the camera calibration process, which is merged with the associated bipolar lines to obtain the transformation matrix required (N.Ayache,1988:11-16). Enhanced the technique by acquiring two matrices from the original cameras (A.Fusiello, 2000:16-22). Through local contact between cells in neural layers, as well as structures and transformations that have a hierarchical organisation of the image created by (Kunihiko Fukushima, 1980:193-202). This results in translational stability. The gradient error algorithm has been used to create deep neural networks with great results in several fields, including pattern recognition. Classified the activities properly and categorised them by their receptive field of recognition (W. Sun, 2005:356-361),(Lin,2021:7041-7048). Through image calibrated using a stereo vision system, the intrinsic and extrinsic matrices for the cameras are constructed, and the original system's projective matrices are computed by rotating the projection matrix acquired around the visual centres. They are two new projection matrices obtained. The estimation and transformation matrix is

applied to the separated images captured by the two cameras. By using vision systems, we determine the differences between stereo picture pairs to reconstruct depth images (H. Su.,2011:17-24). Recent years have seen the emergence of primary data sets for picture categorisation. Data sets contain handwritten numbers, street-view houses of numbers, and objects. The need for real-time computing for robotic tasks, the proposed working a principal component analysis (PCA) with CNN model based on removing redundancy data in hidden layers ensures rapid implementation of a deep neural network. In addition, many organisation algorithms have been proposed in these models. They observed a co-adaptation of neurons in deep neural networking (Nitish Srivastava, 2014:1929-1958). The Dropout network uses a subnet during training, while testing uses an average model. The dropout procedure, which means disconnecting, broadens the algorithm. Rather than dropping nodes, the discrete network drops connections, illustrating that dropping nodes is a specific case of the proposed network pioneered by using CNN for pixel-wise categorisation tasks. The patch-by-patch scanning technique was computationally wasteful and included duplication. A study presented by(J. Shao,2015: 4657-4666) developed a multi-task deep model to jointly learn using a deep neural network and combine appearance and motion features for crowd understanding (Haoyao.C.,2012:1069-1080). Through the proposal, learning-based exploration algorithms for a strategy to reduce arithmetic time and add to almost ideal exploration strategies in uncertainties. Furthermore, using the local robot map and deep augmented learning (DRL), a method for solving the exploration of autonomous mobile robots without taking uncertainty about localisation into account is proposed (R.Socher,2013:1631-1642). The paper presented the problem of track-tracking control for the mobile robot via disruption rejection in the presence of wheel slips and slip phenomena. Kinetic and dynamic models were created with turbulent non-economic constraints (X. Gao, 2021:222). Several proposed models employ these layers to carry out a categorisation task. AlexNet model (Krizhevsky, 2012:1079-1105), which proposed the corrected linear unit (ReLU), is one of the most popular (Agarap, 2018) to improve precision and performance the benefit of ReLU is that it accelerates the convergence process during network training. Later, the VGG model was developed. (Simonyan, 2014:14091556) introduced convolutions into factors, which improved CNN-based models. VGG uses narrower reception windows in convolutional layers, such as kernels or filters, which

minimises network size and improves performance. The parameter size, however, is big, which cannot be beneficial for embedded systems. Google researchers have released a MobileNet version of their architecture. (Howard, 2017:2-7). MobileNet is specifically designed for mobile systems or systems based on embedded architecture, which have higher hardware limitations. This model reduced training time and improved performance. Furthermore, MobileNet had a total of 28 layers and had an accuracy of 87 per cent on the ImageNet dataset. (Dourado, 2019:100,558-881). He proposed an online IoT system based on machine learning methods to localise mobile bots. They began with a GoPro to collect data and then used the omnidirectional camera to do the translation task based on the image rating. Using Convolutional Neural Networks (CNN), they extracted information from photos and then classified them using a machine learning classifier. Although their model has achieved good accuracy, it does not work totally. (Foroughi, 2019:1-7) proposed the Monte Carlo method and a model for robot localisation in CNN-assisted hand-drawn maps (MCL). Their localisation model is two-stage. In the first stage, they extracted data through the map's different locations and trained this data on a CNN designed to predict the location. In the second stage, the robot's actual location on the map was more accurately assessed using the MCL model (Kendall, 2015:2938-2946), proposing a model called PoseNet ++. For the Android reset problem. Their model is an improved version of the VGG16 grid that uses the input image to retrace the camera's position. Despite using the weight from the pre-trained models, their model had limited accuracy due to the data set's small size (Radwan, 2018:4407-4414). Introduced VLocNet ++ for camera location estimation using a CNN-based multi-tasking learning method. Their model integrates semantic and geometric information surrounded by regression networks. However, their approach may fail in an environment with similar traits due to its low generalizability. (Wu, 2018:373-379). Suggested a point cloud map based on the ORB-SLAM algorithm for robot localisation and extensive 3D application. In an indoor environment. Although their model performs well in a 3D mapping challenge, even in a large-scale scenario, the proposed model's effect on the robot placement procedure is little. The environment is then represented as a semantic network containing topological information, with efficient graph matching used to identify the location. Finally, the object aligned to rectify the drift issue. Despite its excellent endurance, the proposed approach is

unsuitable for scenes with few elements. (Saravanan , 2019:7-13) Using CNN and Q-Learning. Have the robots independently search for and access plants in an indoor environment to monitor plant health. They used CNN and reinforcement learning to create an autonomous mobile IoT robot. (Ran, 2021:189) designed a shallower CNN with higher scene classification resolution and efficiency for single-camera image processing. In addition, an adaptive weighted control algorithm and regular control are used to improve the robot's movement performance. However, its ability to self-learn and make continual decisions is still limited. Through our analysis of previous and recent studies in our field of study, which aim to build an independent robot capable of navigating in an unknown environment using the combination of deep learning and computer vision and to achieve the goals we have referred to in chapter one, we conclude that:

1. The Studies that work to tackle aberration and distortion in the cheap cameras through calibration processes, used to address this problem and then build a depth image that will be inputs to the deep learning network. Adopt each of (N.ayache, 1983) ,(A.Fusied,2000) ,(H.Su,20011) this approach and have reached satisfactory results, part of which can be adopted in our present study.
2. The Studies that work in designing deep learning models that are characterize by the speed and computational lightness that serve our field of study. Adopt each of (Kunihko Fukushima,1980) ,(W.Sum,2005) ,(Haoyao ,2012), (Nitish,2014), (Simonyan,2014), (J.shao,2015) ,(Hword,2017), (Wu,2018), (Agrop,2018), (Foroughi,2019)and (X.Gar,2021) this approach can be used for our current study by designing our model and how it is trained to explore signs of control through deep images.
3. The Studies that build and design autonomous robots and localization and navigation processes that rely on advance maps and deduced from the environment in which the robot operates. Adopt each of (R.Socher,2013), (Kedell,2015) ,(Radwan,2018) ,(Saravanan,2019),(Dourad,2019) and (Ran,2021) this approach. In this study is characterized by a kind of autonomy through our use of new dataset use to trained our model by deep learning to enable the robot to navigate independently, flexibly and at low cost.

## CHAPTER III

### RESEARCH METHODOLOGY

In this section, detailed information about the research model will be included, particularly how low-cost sensors are designed to be depth images as well as the methods used to process aberration and distortion in images that they produce through computer vision using cameras calibration methods, low-cost robot building methods, and methods to control its navigation. It is used to build a new dataset that represents the research sample, as well as the deep learning techniques, achievements, and metrics it uses to detect objects in images and methods and classify them through the design adopted to train our model. Studying these methods achieves the goal of our study in empowering a mobile, independent robot for the navigator in an indoor environment.

#### **3.1. Methodology**

##### **3.1.1. Computer Vision and Camera Calibration Methods**

Determining the camera's internal geometric and optical properties (intrinsic parameters) and/or the 3D position and orientation of the camera's coordinate system relative to a particular global coordinate system (external parameters) in the context of 3D machine vision is known as camera calibration. An accurate estimate of these parameters is crucial for executing successful 3D reconstruction from 2D environmental photos at extended working distances in artificial vision applications. When applying stereoscopic vision algorithms, these applications' precision substantially impacts their performance (S. Nedevschi, 2002:21-28). As a result, in computer vision applications, camera calibration is crucial. Internal calibration and external calibration are the two parts of camera calibration. Some experts whose algorithms are utilized in camera calibration (Zhang Z, 2002), (Scaramuzza D, 2006), (Tsai, 2003: 05-11). Two-stage methods: first, linearly calculate the initial values of the parameters, and then build a non-linear camera model. (This includes lens distortion and noise.) The final parameter values are determined by minimizing the error function between the estimated and observed data when solving the problem of fitting non-linear least squares data. The simplest camera calibration method based on

the punch-hole camera is Direct Linear Transformation (DLT) (Abdulaziz, 1971). It is used to calculate camera parameters in advance. Thus it does not account for radial or unintentional anomalies, but it is still useful in computer vision (B.J.Lei, 1999).

1. The pinhole model implies that only first-order radial distortion occurs in the camera and that only one picture pixel distorted in the radial direction, allowing the use of a radial alignment restriction (RAP) model. To make the calibration procedure more precise (Tsai, 1987:232-244).
2. The camera must control a flat image that appears in multiple directions as part of the calibration method (at least in two). The distortion of a radial lens is modeled. The calibration parameters are determined using a closed (linear) solution based on homogeneous symmetries, and then a nonlinear optimization method based on maximum likelihood criteria (Zhang Z, 1999: 666-673).
3. The Bouguer approach (used in Intel Corporation's (CalTech Calibration Toolbox) (2001)) makes use of an internal camera parameter.

At first, all points around the image series are studied, and the basic parameters are set to 0, with the distortion set to 0. After that, each template image's exterior parameters are calculated. Zhang's approach augmented by random distortion coefficients in the final nonlinear optimization utilizing the greatest likelihood estimate.

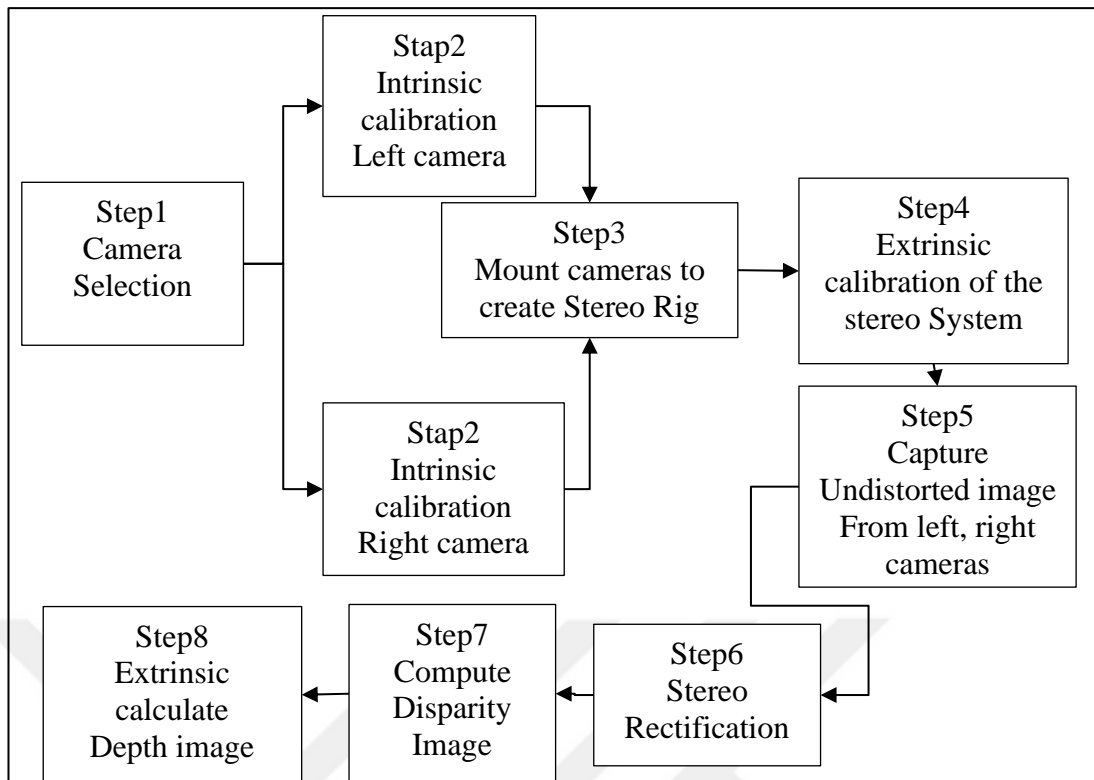
### **3.1.2. Stereo System and Depth Image**

A stereo camera is a type of camera that has two or more lenses, each with its image sensor. In our project, we used two Kisunli cameras with a resolution of 640 x 480 pixels and a USB 2.0 interface, both of which have identical features. Stereo systems can imitate tiny human vision, allowing 3D images to capture (depth images). The baseline length, defined as the distance between the cameras and governs the range of depths that may be consistently observed, is an important parameter of a stereoscopic vision system. The application primarily determines the length of a stereo baseline, as a longer baseline improves both the minimum and maximum range, whilst a shorter baseline can reduce the minimum and maximum range. At the same time, a shorter baseline can reduce the limits (M.Okutomi., 1993). As a result, the baseline length of the stereo device, that is, the distance between two cameras in a stereo system, is set at 63 mm, which is approximately 6.3 cm, based on the typical human eye pupillary

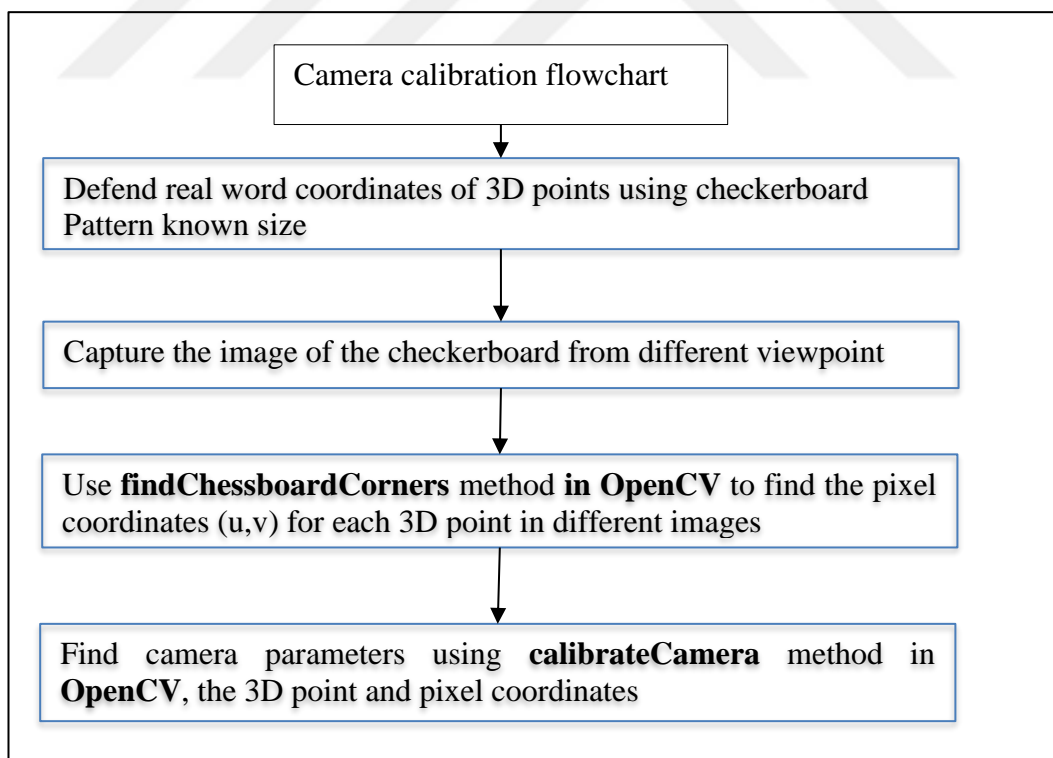
distance of 63.2 mm. Because the internal movement of a human and a robot is similar, the distance from the human eye (Dodgson, 2004:36-46) is the best choice for a base length (see Appendix 8 Figure B). Two webcams with the same function make up the stereo.

### **3.1.3. The calibration and Camera Model**

The lens distortions added to the pinhole camera model are shown in Appendix 9, figure A. The modified version of the original Caltech Calibration Toolbox was used for the initial estimation. The Bouget (CalTech) method (Heikkila, 1997:1106-1112) The intrinsic parameters are initialized by finding symmetries for all control points in the image series during the initialization stage. Reduce the distortion to 0 per cent. After that, the exterior attributes of each pattern image were discovered. Zhang's method is similar to the nonlinear optimization using maximum likelihood estimation (Zhang, 1999:666-673) as the transverse distortion coefficients increase. A flat plate with black and white squares of defined sizes is used as a calibration pattern see Appendix 9 Figure (B), and each camera is calibrated separately. A series of at least 20 picture files should be recorded in varied orientations and distances for intrinsic parameters, with the outline spanning the maximum region of the image as much as possible. An excellent set of input images reduces parameter estimation uncertainty (5 pixels) and allows the method to converge in fewer steps. The external parameters are then calculated. The cameras are mounted on a stereo platform for this task, and the model is put in a known position to the cameras. The two cameras in the stereo system take one picture of the chessboard, and the internal exterior characteristics of a unit camera are determined because the two cameras have identical properties. According to the global reference frame linked with the calibration pattern, these images were saved separately for the two cameras. The step calibration camera shows in Figure 3.1, and Figure 3.2 shows the calibration flowchart using OpenCV in Python.



**Figure 3. 1 The step diagram of camera calibration method (S. Praveen, 2020:113)**



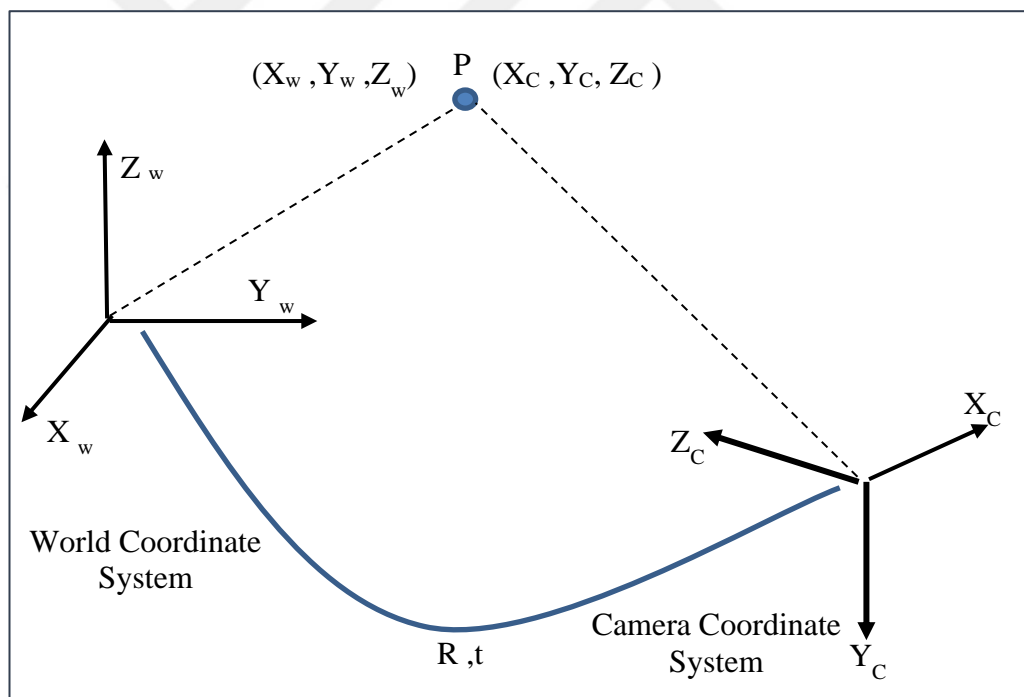
**Figure 3. 2 The steps diagram of camera calibration method (Researcher's work)**

The following camera parameters are considered:

1. Intrinsic parameters: those that describe the camera's inherent geometric and optical features (those that define the camera itself):

- A. The focal length of a camera lens is the distance between the lens and the image plane.
- B. the lens's radial and tangential distortion coefficients;
- C. the primary point - the location of the image centre in pixel coordinates:  $(u_0, v_0)$ ;

2. Extrinsic parameters: the camera frame's 3-D position and orientation in relation to a given world coordinate frame (used to convert world coordinates to a camera centered coordinate frame or vice versa): To undistorted the image, the lens parameters obtained using geometric calibration were apply. To comprehend the calibration process, we must first comprehend the geometry of picture production. As explained in the post, to find the projection of a 3D point onto the image plane.(S. Praveen,2020:111-127)



**Figure 3. 3 The World Coordinate System and the Camera Coordinate System (Researcher's work)**

The extrinsic parameters of a camera are the six parameters (3 for rotation and 3 for translation). We first need to transform the point from the world coordinate system to the camera coordinate system using the extrinsic parameters (Rotation  $\{R\}$  and

Translation {t}).  $T = [T_x, T_y, T_z]$  is a translation vector, as shown in Figure 3.3. The rotation vector  $T$  is  $r = [R_x, R_y, R_z]T$  or  $R$  (rotation matrix equal). The projection of a 3D point  $P$ , having coordinates  $(X, Y, Z)$  in the camera reference frame, on the image plane is computed by using the intrinsic parameters matrix as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = P \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3.1)$$

Where  $P$  is a 3x4 projection matrix made up of two parts: the intrinsic matrix ( $A$ ), which holds the intrinsic parameters, and the extrinsic matrix ( $R | t$ ), which is made up of a 3x3 rotation matrix ( $R$ ) and a 3x1 translation vector ( $t$ ).  $[x \text{ and } y] T$  is the undistorted picture point vector in the camera coordinate frame; the fundamental equations system of the perspective camera model gives the following relationship between  $[x \ y]$  and  $[X \ Y \ Z]$ :

$$x = f_x \cdot \frac{X}{Z}; y = f_y \cdot \frac{Y}{Z} \quad (3.2)$$

$f_x, f_y$  are the focal lengths in pixels, both horizontally and vertically. The letter  $p$  can stand for a variety of things.

$$P = A \times [R | t] \quad (3.3)$$

$A$  – is the internal camera matrix:

$$A = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$[\delta x^r \ \delta y^r]$  : is the radial distortion, modelled using the first and second order radial

$$\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} \delta_x^r + \delta_x^r \\ \delta_y^r + \delta_y^r \end{bmatrix} \quad (3.5)$$

Distortion coefficients  $k_1$  and  $k_2$ :

$$\begin{bmatrix} \delta_x^r \\ \delta_y^r \end{bmatrix} = \begin{bmatrix} x(k_1 \cdot r^2 + k_2 \cdot r^4 + \dots) \\ y(k_1 \cdot r^2 + k_2 \cdot r^4 + \dots) \end{bmatrix} \quad (3.6)$$

Where:  $r^2 = x^2 + y^2$ ,  $[\delta x^t \ \delta y^t]$  : The tangential distortion coefficients  $p_1$  and  $p_2$  model the tangential distortion:

$$\begin{bmatrix} \delta_x^t \\ \delta_y^t \end{bmatrix} = \begin{bmatrix} 2p_1 \cdot xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y^2) + 2p_2 \cdot xy \end{bmatrix} \quad (3.7)$$

The following matrix equation can be used to transfer the coordinates of the point P expressed in the world reference frame  $XX_w = [X_w, Y_w, Z_w]^T$  to its coordinates in the camera reference frame  $XX_c = [X, Y, Z]^T$  (direct mapping).

$$XX_c = R_{wc} \cdot XX_w \quad (3.8)$$

Where: The world to camera translation vector is  $XX_{wc}$ , which represents the position of the world reference frame in the camera's one. The world to camera rotation matrix known as  $R_{wc}$ .  $R_X$  (pitch),  $R_Y$  (yaw), and  $R_Z$  (tilt) are Euler angles that create a series of three elementary rotations of the world frame around the x, y, and z-axes, respectively. The right hand rule provides the angles signals. The equation can be inferred from (2.9) for the reverse transformation (inverse mapping - 3D reconstruction) from the camera reference frame to the world reference frame:

$$XX_w = R_{wc}^{-T} \cdot (XX_c - T_{wc}) \quad (3.9)$$

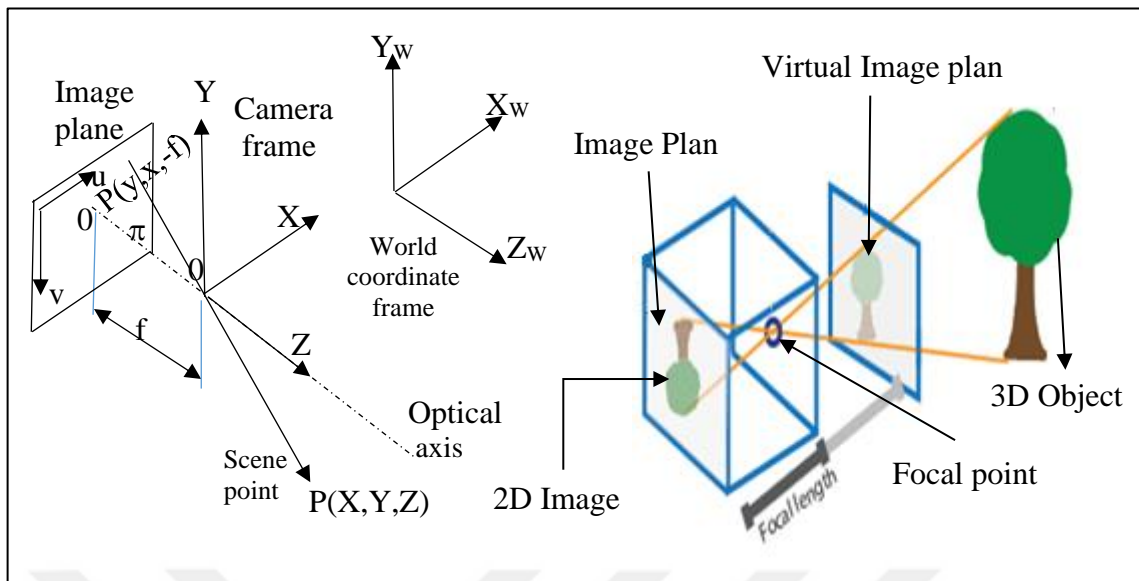
Because the rotation matrix is an orthogonal one the inverse of  $R$  is equal with its transpose:

$$XX_w = R_{wc}^{-T} \cdot (XX_c + T_{wc}) \quad (3.10)$$

$T_{cw}$  : is the camera to world translation vector  $R_{cw}$ - is the camera to world translation matrix

$$R_{cw} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (3.11)$$

world Coordinate System  $O_{world} = [X_w, Y_w, Z_w]$  and Camera Coordinate System  $C_{camera} = [X_c, Y_c, Z_c]$  is Pixel Coordinate System and  $O_{image} = [u, v]$  (S. Nedevschi, 2002:2-8)



**Figure 3. 4 the pinhole camera model(S Nedevschi,2002:4), (Researcher's work)**

Through the following steps, calibrations performed for the calendars configured in the stereo system:

1. Define real-world coordinates using a checkerboard pattern where the world coordinate system means that the world coordinates are fixed by this checkerboard pattern shown in Appendix 9 Figure (A), which is mounted to a wall in the room. Our 3D points are the checkerboard squares' corners. Any corner of the above board can choose as the origin of the global coordinate system. The  $X_w$  and  $Y_w$  axes parallel the wall, and the  $Z_w$  axis is perpendicular to the wall. As a result, all checkerboard points are on the  $X, Y$  plane (i.e.  $Z_w = 0$ ). During the calibration procedure, we determine the camera parameters using a set of known coordinates  $(X_w, Y_w, Z_w)$  and their corresponding pixel location  $(u, v)$  in Figure 3.4. We photograph a checkerboard pattern with known dimensions in various orientations for the 3D points. The world coordinate is tied to the checkerboard, and because all the corner points are on a plane, we can arbitrarily set  $Z_w$  to 0. Because the checkerboard points are evenly spaced, the  $(X_w, Y_w)$  coordinates of each 3D point are readily defined by picking one point as a reference  $(0, 0)$  and defining the remaining concerning that reference point. In an image, checkerboard patterns are unique and easy to spot. Furthermore, because they contain strong gradients in both directions, the corners of squares on the checkerboard are ideal for localization.

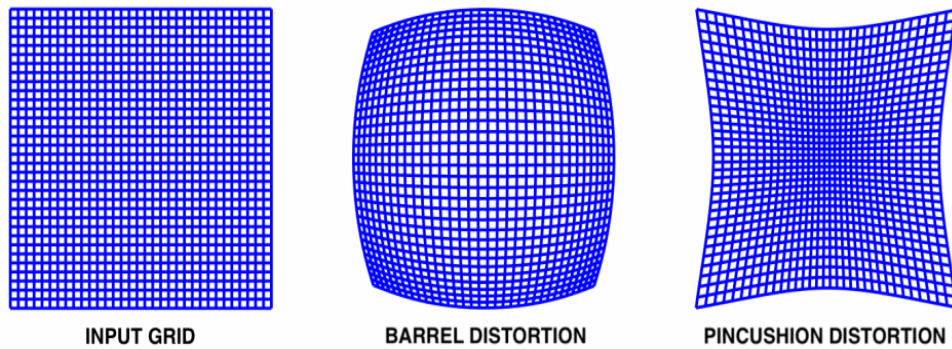
Furthermore, these corners are joined because they are at the intersection of checkerboard lines. All these facts are used to robustly locate the corners of the squares in a checkerboard pattern,

- 2- Capture multiple images of the checkerboard from different viewpoints shown in Figure 2. 3 shows images used for calibrating the camera. Next, we keep the checkerboard static and take multiple images of the checkerboard by moving the left and right cameras in a stereo system. Alternatively, we can keep the camera constant and photograph the checkerboard pattern at different orientations. The two situations are similar mathematically.
- 3- Determine the 2D coordinates of the checkerboard now that we have many images of the checkerboard. We also know the 3D position of checkerboard points in world coordinates. The last thing we need is the 2D pixel coordinates of these checkerboard corners in the images. We used to search for checkerboard corners. Find Chessboard Corners is a built-in function in OpenCV that searches for a checkerboard and returns the coordinates of the corners.
- 4- Camera Calibration The final calibration step is to send the 3D points in world coordinates and their 2D locations in all images to Open CV's calibrate Camera function.( K.D. Kuhnert,2007:311-334)

#### **3.1.4. Major types of distortion effects and their cause**

We obtain better images when we use a lens, yet the lens produces some distortion effects. Distortion effects classified into two types:

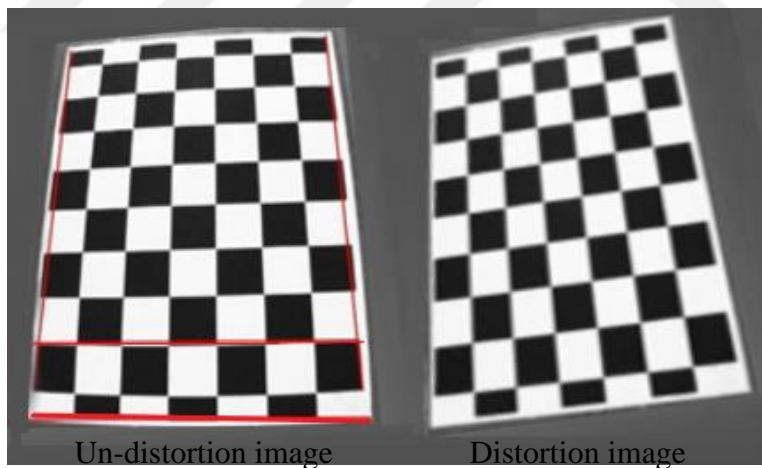
- 1- Radial distortion is the result of uneven light bending. The rays bend more at the lens's edges than at the lens's centre. Due to radial distortion, straight lines in the real world appear curved in the image. The light ray moved radially inward or outward from its ideal location before contacting the image sensor. The effect of radial distortion can be divided into two sorts.
  - A- The effect of barrel distortion, which corresponds to radial displacement in the negative direction.
  - B- Pincushion distortion, resulting in radial displacement that is positive.
2. Tangential distortion occurs when the picture screen or sensor angled in relation to the lens. As a result, the image appears to be skewed and stretched. Figure 3.5 depicts this.



**Figure 3. 5 Image explaining the effect of barrel distortion and pincushion distortion (K.Sadekar,2020:3.)**

This problem is addressed by using the matrices we got through the calibration of cameras e previous post on camera calibration, we obtained the camera matrix and distortion coefficients. These coefficients are used cautiously to prevent distortion of the camera image. It is done through one application to undistort the image using the computed distortion coefficients.

The images in Figure 3.6 show the effect of lens distortion and how it can be removed from camera calibration coefficients.



**Figure 3. 6 The effect of lens distortion.(left) and Image after un-distortion (right). (K.Sadekar,2020:4).**

Based on the cause of the distortion, there are three types of distortion: radial distortion, decentring distortion, and thin prism distortion. Decentring and thin prism distortion both cause radial and tangential distortion. Using OpenCV to remove the distortion . To reduce lens distortion, three fundamental actions must take.

1. Get the intrinsic camera parameters by calibrating the camera. This is exactly what the previous chapter of this series accomplished. The intrinsic parameters also contain the camera distortion characteristics.
2. Fine-tune the camera matrix to keep the proportion of unwanted pixels in the undistorted image under control.
3. Remove distortion from the image using the improved camera matrix.

Various viewpoints around the image's edges for analyzing distortion and improving estimation, more samples around the image border would be accessible. (K.Sadekar,2020:1-10).

### 3.1.5. Semi-global block Matching Function (SGBM)

By using, a few methods, such as Semi-Global Block Matching (SGBM), have proven to be sufficient. Semi-global block matching (SGBM) is an example of a real-time dense matching algorithm. SGBM employs dynamic programming across a discrete cost volume, which can accomplished efficiently with the aid of an FPGA, GPU, or multi-core CPU (HIRSCHMULLE,2008:1582-1599) Global approaches are frequently too computational for real-time application, although some methods, such as Semi-global block Matching (SGBM), have proven to be enough. Semi-global block Matching (SGBM) is an example of a dense real-time matching approach. SGBM employs dynamic programming on a distinct cost size, which can accomplished efficiently with an FPGA, GPU, or multi-core CPU (HIRSCHMULLE, 2008:1582-1599). It was also utilize at OpenCV using Python or C++ and FPGA developed in real time (C. Banz, 2010:93-101).By minimizing the following energy functional.

The SGBM finds smooth disparity maps by use question (3.12, 3.13).And it's the technique used to create depth images in the stereo cameras system, which was use in our study to create a matching and output the contrast between the two images coming from the left camera and the right camera after their calibration.

$$E(d) = \sum_p C(P, d_p) + \sum_{q \in N_p} P1\delta(|d_p - d_q| - 1) \quad (3.12)$$

$$E(d) = \sum_p C(P, d_p) + \sum_{q \in N_p} P2\delta(|d_p - d_q| - k) \quad (3.13)$$

Where  $p \in R^2$  are represent the centre points of the pixels in the image grid than  $d_p$  through disparity values associated with them.  $N_p$  which is a collection of points that comprises eight of  $p$ 's neighbors  $(p, d)$ , is the cost of matching the block to  $p$ , while  $P1$  and  $P2$  are constant stop penalties for unit and larger hops ( $k > 1, k \in Z$ ) (H.Hirschmuller,2008:1588-1599).

The global collection of directional costs occurs in two passes. The first pass collects the costs deployed from the upper and left sides. The second pass incorporates both the lower and upper directions. Finally, the disparities with the lowest overall cost are chosen from the eight trends. To effectively punish oblique surfaces and interruptions,  $P1$  and  $P2$  penalty constants must meet  $P1 \geq P2$ . Following that, the inequity values are enhanced to sub-pixel precision. The obstruction is addressed by assessing consistency, which entails computing the left-to-right and right-to-left disparity maps and then marking disparities as blockages. It is a costly strategy, as the disparity map will be calculated twice. Finally, variation values are refined to sub-pixel precision.

### **3.2. Depth Sensing**

A depth camera is used to obtain depth images in the method described in this letter. Each pixel in the depth image represents the length of the ray that goes from the camera's center to that pixel to reach the obstacle, and the camera is constructed as a single point in 3D space. It creates a detailed depiction of the environment, which is especially valuable for detecting the physical location of barriers concerning the camera. There are various distinct types of depth cameras available in 2019. Stereo cameras, for example, combine two cameras and stereo depth algorithms to simulate the ability of animal eyes to perceive depth. Infrared (IR) integer time cameras are another sort of depth camera that uses pulsed infrared light to measure distance and then measures the time it takes for that pulse to return via each pixel, from which the distance can be computed using the speed of light. Infrared cameras, which monitor the phase shift of returning light and can use to determine distance, are similar. The advantage of depth cameras is that they provide a detailed three-dimensional world image. This information can be utilized to avoid impediments hitting the robot's top from below or above the ground. In addition, infrared-based depth cameras are lighter and more stable than RGB-based, allowing them to operate in different lighting conditions, such as outdoors or in dark rooms (R. Weidemann, 2019:14).

### 3.3. Deep Learning

The structure of the visual system inspired Convolutional Neural Networks (CNNs), the models proposed in (D. H. Hubel, 1962:106-154) The first computational models based on this local connectivity between neurons and hierarchically organized image transformations found in (K.Fukushima, 1982:193-202), which describes how neurons with the same parameters applied to patches of the previous layer at different locations result in a type of translational invariance Yann LeCun and colleagues then used gradient error to create convolutional neural networks that performed well in many pattern recognition tasks.

To recognize it, deep learning has been frequently employed in computer vision. These tasks can be divided into two categories based on the receptive domain of recognition: corrective classification and pixel-wise classification (W. Sun, 2005:356-361), (P.D.L, 2007:22). In image classification, the example of patch-wise classification is utilized. Its purpose is to give each image a label. Primary data sets for image classification have emerged, which have appeared in recent years. Verification methods new methods. Which are including These data sets include handwritten numbers, such as MNIST (Geoffrey E Hinton, 2006:1527-1559), street view house numbers, such as SVHN dataset (J. Shao, 2015:4657-4666), and objects, such as CIFAR-10 (Hongsheng Li,2014:1412) Object detection, localization, and recognition is another example of patch-wise categorization.

In this challenge, one must first detect possible places for an object before recognizing it. Several deep neural networks have been published in the literature to handle the previously listed problems and obstacles. Most of these methods used a convolutional neural network (CNN) for feature extraction. Which model from LeNet5 (Yann LeCun, 1990:396-404) was presented for handwritten digit recognition and outperformed typical shallow models regarding recognition accuracy? Since then, various CNN versions for feature extraction and representation have been presented (Min Lin, 2013:569-597). The network was in the network. Was it a feature extraction model that combined convolution with multi-layer perception?

In our previous work, we presented a principle component analysis (PCA)-The CNN model (Shaohua Li, 2015:57-62), prompted by the necessity for real-time computation for robotic applications. The CNN model is based on reducing redundancy data in

hidden layers, ensuring the speedy deployment of a deep neural network. Furthermore, a variety of organisation algorithms have been presented in these frameworks. They discovered a coadaptation of neurons in a deep neural network there (Nitish Srivastava, 2014:1929-1958). A dropout algorithm was proposed. During training, a subnet is taught for each iteration of the Dropout network, whereas testing uses an average model. Drop-Connect, a more generalized technique, was present for the Dropout action. Instead of eliminating nodes in the network, the discrete network drops the connections, i.e. the neural network's weights, demonstrating that dropping nodes is only a specific instance of the proposed network. Regarding pixel-wise classification challenges, it was the first to incorporate CNN into pixel-wise tasks and use it in classification

.However, their patch-by-patch scanning approach could have been more computationally wasteful, with numerous duplicate computations. It was proposed to use a completely convolutional neural network layer (Clement Farabet, 2013:1915-1929). It greatly lowers processing redundancy and might be adapted to big input sizes. The concept of a deep network is broadened in this work for cognition and decision-making tasks. Although the validity of these proposed methods has been tested in common datasets, how well these methods would work in practice still needs to be determined. Furthermore, recognition tasks can only be viewed as an intermediate result, requiring further reasoning and decision-making when it comes to robotic applications. Meanwhile, a one-stroke training technique frequently utilized by computer vision experts may not be appropriate for robotic applications. It is more suitable to use reinforcement learning algorithms to allow the system to improve performance and increase confidence in each decision-making process.

### **3.4. Deep Learning for Decision Making**

Recently, Q-learning models have been successfully integrated with CNN for human-level control (R.Weideman, 2019:12-15)/ (V.Mnih, 2013:4-6) and have been successfully employed in deep neural networks. Many popular games have validated the proposed strategy. The results show that the suggested approach functions admirably when dealing with uncomplicated scenarios. This system's output will be limited when facing obstacles requiring more consideration. Furthermore, because the user input in this scenario is a gaming screen, probability and uncertainty were omitted

from the model (Jun, 1996:421-434). A learning technique was proposed based on the model planning scenarios in a chosen environment using a 2D laser band detector. DQN was used by (Hel, 2018:2154-2154) to describe a system for driving a car in a virtual environment. Only five actions were established, each corresponding to a distinct steering angle, and training pictures of 4827 pixels were used. The reward function was calculated by considering the distance from the lane center and some additional data (such as the angle error between the vehicle and the centerline). (Hel, 2018:2154-2154)), the simulation results indicate that the model is valid. Although it can be deduced from the models described above that deep neural decision-making networks are viable due to their properties and the Q-Learning approach used throughout the learning process.

However, deep learning methods are not widely applicable to real-world situations.(J. Wang, 2019:1-6), employed the safety rate to evaluate the accuracy of their model, which was derived using collision frequency and gave a clear depiction of stimulation. The results suggest that models that consider the speed and other characteristics beat models that do not consider the average vehicle speed. Decision-making in self-driving cars is a huge challenge. As a result, (M.Ghimire, 2021:2-4) proposes that lane-changing decisions be made using a Deep Q-Network with rule-based limitation.

### **3.5 .Training**

Back-propagation is a method of training neural networks that attempt to minimize some loss function, frequently described as the square difference between the network's actual and desired output. The derivative of this function concerning the network weights can be viewed as the gradient of the high dimensional surface formed by the network weights. Gradient descent, accomplished by propagating calculated gradients backwards through the network, can be used to update the weights. Although a global minimum corresponds to the best combination of weights that would minimize the loss function in theory, gradient descent is only guaranteed to discover a local minimum in practice. There are numerous strategies for estimating the gradient and performing gradient descent such that a good minimum is found (R.Weideman, 2019:11)

### 3.6. Convolutional Neural Networks

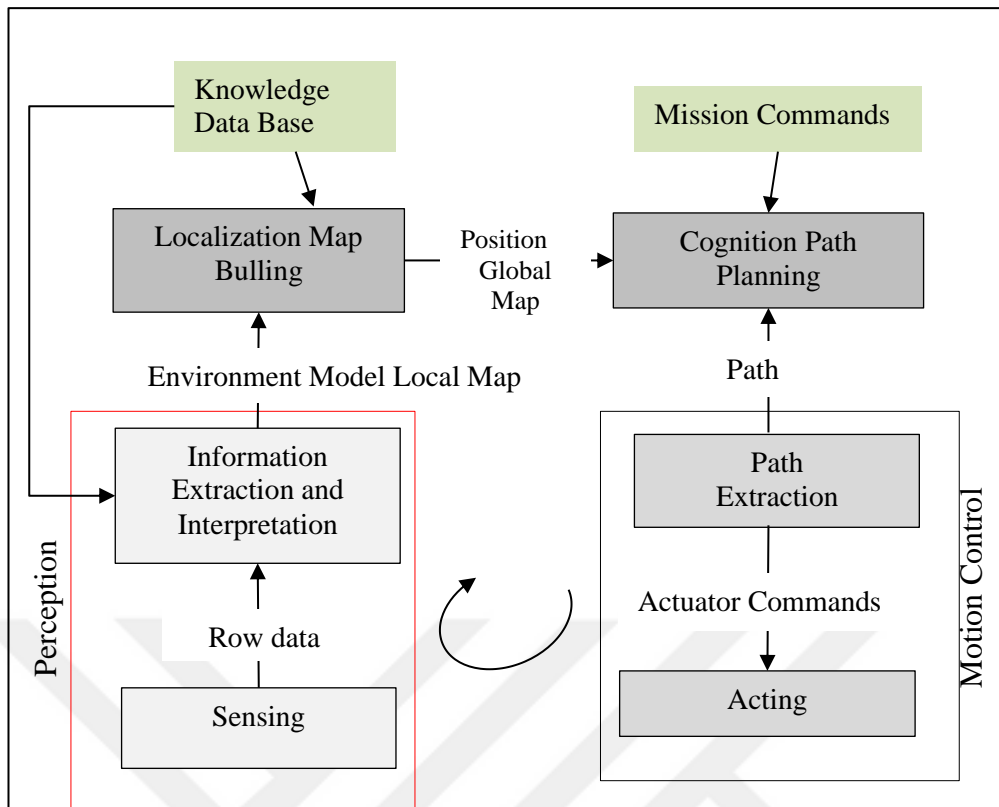
Convolutional Neural Networks (CNNs) are a type of deep neural network that can conduct picture abstraction at a high level. In 1998, Yann LeCunne popularized CNN with LeNet, an early functional CNN handwriting and machine character recognition implementation (Yan LeCun, 1990:396-404). The extracted image features will be manually created in the classic pattern recognition technique, and a separate classifier will be trained to detect these features. The difficulty is that the engineer's capacity to choose the best jobs limits the system. Convolutional neural networks, on the other hand, provide comprehensive learning by learning both feature extraction and classification at the same time.

It enables CNN to choose the best image features on its own during training, putting it at the forefront of pattern recognition standards. A convolutional network has three sorts of layers: convolutional, aggregate, and fully connected. The convolutional layer is a significant advancement. All three dimensions are the breadth, height, and depth of convolutional layers. The filter's dimensions for that layer are width and height, and the number of filters in that layer is depth. The convolutional layers' neurons are only connected to a small portion of the layer before them. Filters are the size of this region, also known as the receptive domain of neurons. The values in the filter kernel are learned during training since the filters are linked to the input images to extract the feature.

The CNN has fewer parameters to train than a fully connected deep neural network since the neurons are only connected to a tiny area of the image, and each region has the same filter weights. Similarly, parallelizing the convolution process in the implementation is simple. CNNs have risen in popularity and success since introducing powerful GPUs that enable effective parallel computation. The output of the network's convolutional layers is normally routed to a single big layer, which is subsequently fed into numerous fully linked layers. These layers are intended to classify features obtained from a CNN network segment.(Yan Likon, 1990:296-404).

### 3.7. Mobile Robot

In the industrial field, great successes have been made in using robots. It is done by installing the robot arm, consisting of robotic arms or manoeuvres at its shoulder in a specific location in the assembly line. This arm can move with speed and high precision to accomplish repetitive tasks and processes such as topical welding and painting, as shown in Appendix 10 Figure (A). Robots have also entered the electronics industry as they install electronic components on the surface with extraordinary accuracy, allowing the creation of mobile phones and laptops. Despite their victories, these commercial robots have a fundamental weakness. Such industrial robots are almost stationary, i.e. have no navigation system where the stationary processor has a limited range of motion. Our thesis focuses on mobility technology, which allows the mobile robot to travel uncontrolled across real-world surroundings to complete its responsibilities. The first challenge is that navigating an unfamiliar ocean, such as Mars, causes more unique mobility mechanisms, as evidenced show Appendix 10 figure(B) (R.Siegwart, 2011: 181-196). Using a variety of sensing techniques. The knowledge of computer algorithms requires information theory, artificial intelligence, and probability theory for localization and navigation. Figure 3.7 shows an abstract control diagram of the mobile robot systems that will generate training data using depth images generated by the stereo systems. In this section, we look at several of the important knowledge groups linked with mobile robotics. As a result, we must provide an overview of all aspects of mobile robotics, such as hardware and software design concerns, related technologies, and computational methodologies, in addition to others interested in this field. Although only partially necessary, knowledge of matrix algebra, calculus, and statistics is advantageous. Probability theory and computer programming will significantly improve our abilities to build a mobile and self-contained robot. Mobile robots are a huge expanse, and this section focuses on mobility and the technologies that enable powerful mobility, ranging from mechanism and perception to localization and navigation. A practical and commercially feasible animatronic robot does more than merely move. It cleans the supermarket floor, keeps a guard at the factory, mows the golf course, gives museum tours, or gives instructions at a supermarket. The exciting era of mobile robotics is about to begin. It incorporates technologies from various domains, such as (R. Siegwart, 2011:10-12).



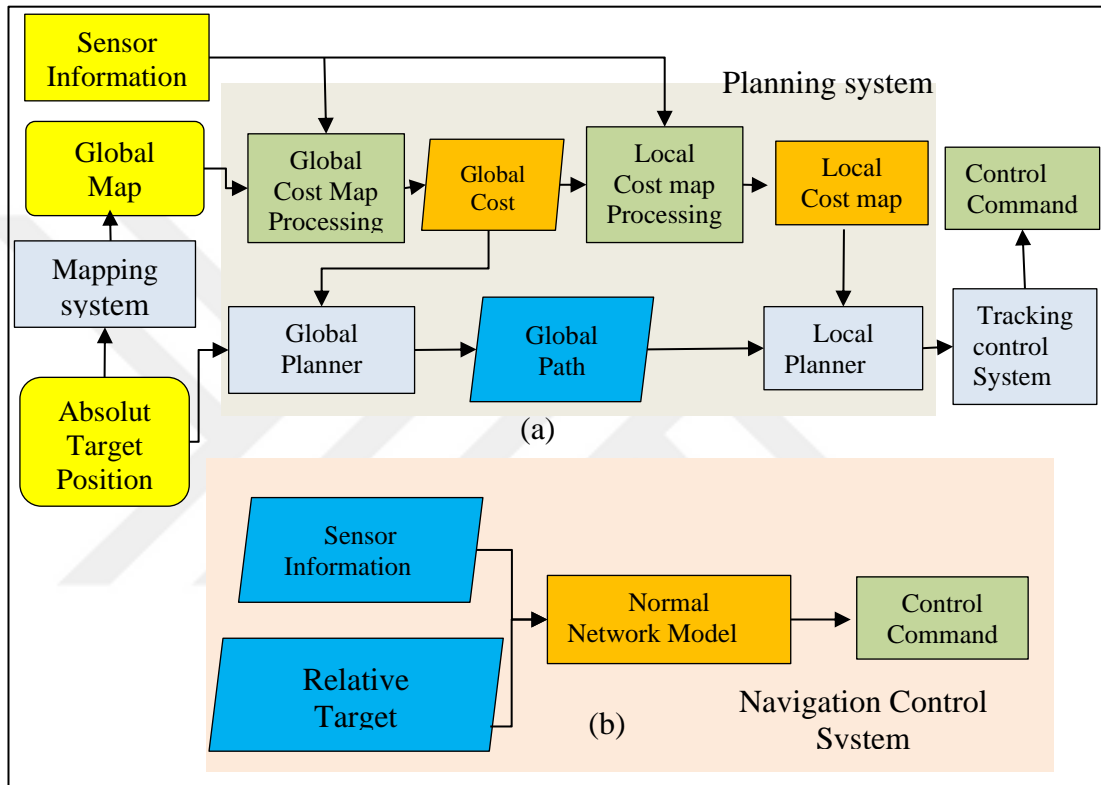
**Figure 3. 7 Reference control scheme for mobile robot systems (R. Siegwart, 2011:10)**

### 3.7.1 Wheeled mobile robots

The wheel is the most frequent locomotion mechanism in mobile robots and manufactured vehicles. Using a relatively basic mechanic, it can reach very high efficiencies. Furthermore, balance is rarely a research challenge with wheeled robot designs because wheeled robots are usually designed so that all wheels are always attached to the ground. Thus, three wheels are sufficient to achieve a stable balance, which is the design chosen in the mobile robot design, the specifications will be explained later, but two-wheeled robots can also be stable. However, if the robot has more than three wheels, a suspension mechanism is required to keep all wheels in contact with the ground when encountering uneven terrain. Instead of focusing on balance, research in wheeled robotics focuses on traction, stability, manoeuvrability, and control problems: robot wheels can provide enough traction and stability for the robot to cover all of the required terrains, and a wheeled robot configuration enables adequate control of robot velocity variation such as human-robot interaction, computer vision, and navigation. (R.Siegwart, 2011: 181-196)

### 3.7.2. Autosense Mobile Robot Controller

Navigation control is one of the fundamental capabilities of autonomous mobile robots; it enables robots successfully execute motion control and obstacle avoidance tasks in the workplace. Figure 3.8 (a) depicts the architecture of the typical map-based mobile autonomous robot control scheme ([http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base), 2021). It is a multi-layered structure containing the mapping, planning, and tracking control systems. The mapping system creates a global map of the uncharted environment.



**Figure 3. 8 System architecture of (a) the traditional map-based navigation control system (b) the navigation system based into DL without maps to control system. (K. Zhu,2021:399)**

A global plotter and a local plotter are included in the planning system to give the ideal course of travel for the moving robot to complete navigation tasks such as approaching a goal or avoiding obstacles. Both the global mapper and the local planner must use the global map containing sensor information and the absolute target position to calculate the ideal path trajectory and investigate autonomy-based robotic motion control commands, respectively. According to (K. Zhu, 2021:396-404), each classic map-based navigation framework system is a difficult research issue. Their integration frequently results in poor computational efficiency due to considerable computational mistakes. Furthermore, the computation of navigation performance

under traditional map-based navigation is highly dependent on the accuracy and quality of the global map supplied for this type of navigation, which is extremely susceptible to sensor noise. The need may limit the navigation system's capacity to operate in unfamiliar or dynamic situations (K. Zhu,2021:396-404). The development of navigation technology based on training data in deep learning networks combined with map-free LiDAR for wheeled mobile robots has attracted growing attention in recent years to reduce the complexity of the architecture of a standard map-based navigation control system. The system architecture of a deep learning-based navigation control system based on model training and learning is depicted in Figure 2.13 (b). It employs a NN model to predict motion control signals for a robot to approach a target while avoiding obstacles, in conjunction with a computer in identifying sensors (depth cameras). When compared between traditional map-based navigation and the technology, deep learning-based map-based navigation offers two major advantages:

1. Global map information requirements issued. This features are makes the navigation system's performance independent of global map quality, but rather based on the diversity of a variety of training data gathered from various surroundings.
2. It predicts motion control commands based on sensor data and relative target location.

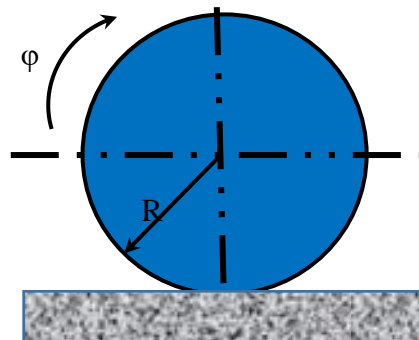
This innovation, which employs a NN model, reduces the complexity of the navigation system model. The possible disadvantage of this simplification is that if the hardware specifications of the mobile platform change, the user may need to retrain the entire neural network model. Several methods have b employed to control navigation without maps. Our model has solved this problem by increasing and diversifying training data in different environments and combining them to boost learning efficiency. The key advantage of simulation learning is that it is extremely useful when an expert system can offer the network model's essential behavior for optimal policy learning by emulating expert behavior.

However, the effectiveness of simulated learning depends on the diversity of behaviors displayed by the expert system to cover a wide range of scenarios. As a result, increasing the behavioral diversity of an expert system is a critical challenge in

using simulated learning (O. Mohamed, 2021:20-29). In this thesis, we combined computer vision and a thorough CNN model to create a navigation control system. The suggested navigation design allows direct use of data supplied by calibrated stereo cameras to create depth images for various settings. Deep learning is used to build a model that outputs the motion control commands the moving robot requires. In the training phase, we use the regression method to train the end-to-end model to learn the mapping between the input data and the output commands and apply the model trained to control navigation. Moreover, we also propose a new data augmentation method by increasing the training samples' diversity to improve simulation learning efficiency.

### 3.8. Control and Motion of Mobile Robot

This study assumes mobile robots (P. Swapna, 2009:541-546). A reliable vehicle with non-formable conventional wheels mounted in horizontal grooves that move on a non-deformable horizontal plane. In motion: The wheel has only one point of contact with the horizontal plane. A mobile robot's wheel and the horizontal plane link are rigid. It assumes optimum conditions with no pure rolling issues and no movement slippage. That is, the velocity of each wheel's point of contact with the horizontal plane is zero. The fixed wheel's centre is a fixed location on the carriage, and  $L/2$  is the distance between the fixed wheel's centre and P, the wheel's rotation angle.  $T$  and the wheel radius denote the axle. Show as figure 3.9



**Figure 3. 9 Idealized rolling wheel (G. Master, 2006:3)**

Hence, the position of the wheel is characterized by 2 constants:  $L/2$  and  $R$  and its motion by a time-varying angle:  $\varphi(t)$ : where the  $\varphi_r(t)$  is the rotation of angle of right wheel,  $\varphi_l(t)$  is the rotation of angle of left wheel. The vehicle velocity denoted  $\mathbf{v}$  can found in equation (3.15):

$$v = \frac{R(\omega_r + \omega_l)}{2} \quad (3.15)$$

Where:  $\omega_r$  is angular velocity of the right wheel, The position and the orientation of the mobile vehicle are determined by a set of differential equations (3.16,3.17) in the following form:

$$\dot{x} = (R \cos\theta (\omega_r + \omega_l)) \quad (3.16)$$

$$\dot{y} = (R \sin\theta (\omega_r + \omega_l)) \quad (3.17)$$

$$\dot{\theta} = R/L (\omega_r - \omega_l) \quad (3.18)$$

Here,

$$\dot{x} = v \cos\theta \quad (3.19)$$

$$\dot{y} = v \sin\theta \quad (3.20)$$

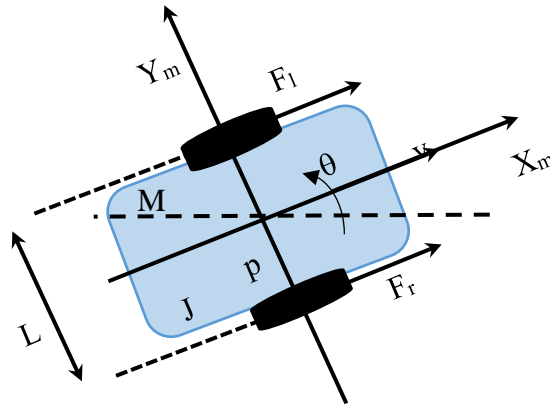
Then the matrix form is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} \quad (3.21)$$

Finally, the kinematic model of the vehicle velocity  $v$  and the angular velocity can be represented by the matrix as follows:

$$\begin{bmatrix} v \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} R/2 & R/2 \\ R/L & -R/2 \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (3.22)$$

Therefore, the angular velocities of the two wheels of the mobile robot figure 3.10 are independently controlled (Shigeki Iida ,1990:859-866)



**Figure 3. 10 Dynamic parameters of a vehicle in motion (G. Mester, 2006:3)**

The vehicle motion equations are:

$$M \frac{dv}{dt} = F_r + F_l \quad (3.23)$$

$$J \frac{d\theta}{dt} = L(F_r + F_l) \quad (3.24)$$

Where the parameters of the vehicle motion are represented too

$F_r$  is the driving force at the right wheel,

$F_l$  is the driving force at the left wheel,

$M$  is the mass of the Vehicle, and

$J$  is the inertia of the Vehicle.

In this section, a method has been proposed to control the front wheels of the robot and the rotation to the left and right sides, according to the scheme in Appendix 6, which contains the dynamics of the reverse robot. The inputs to the front compensator are both sides of the desired wheel angle.( G. Mester, 2006:2-6)

### **3.9. Embedded System for DC Motor Control**

#### **3.9.1. DC motor characteristics.**

The DC motor is an electric motor that converts direct electric current into mechanical power (Moser, 2022:4-31). DC engine speeds can be seamlessly controlled by changing the power supply voltage. In addition, the direction of motion can be reversed by changing the direction of the power supply. Thus, DC engines are used in many

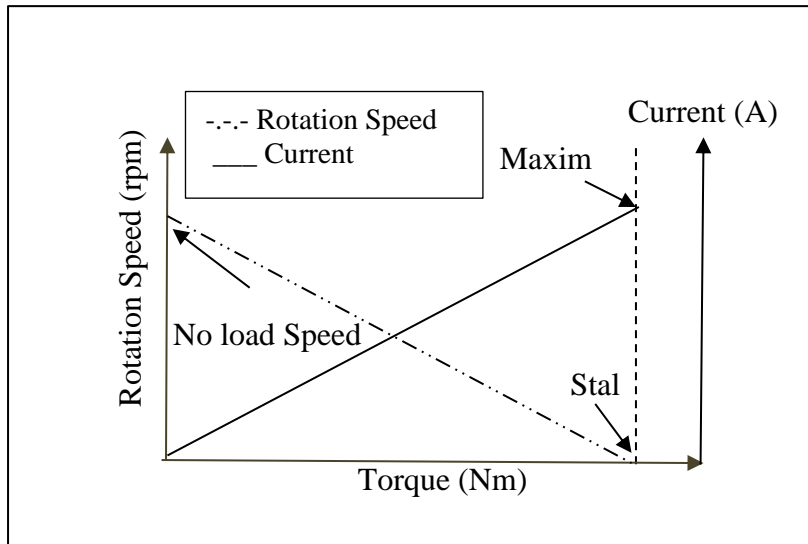
mechatronic applications due to their ease of use and torque properties. Figure 3.11 and Appendix 5 illustrate the distinctive DC engine graph. The DC engine has specific properties, which initiate torque is a large response and linear rotation speed through input voltage and linear torque output by the input current. In addition, the DC engine has a problem because the stalled current is too big when the engine stops.

### **3.9.2. DC motor speed control technique.**

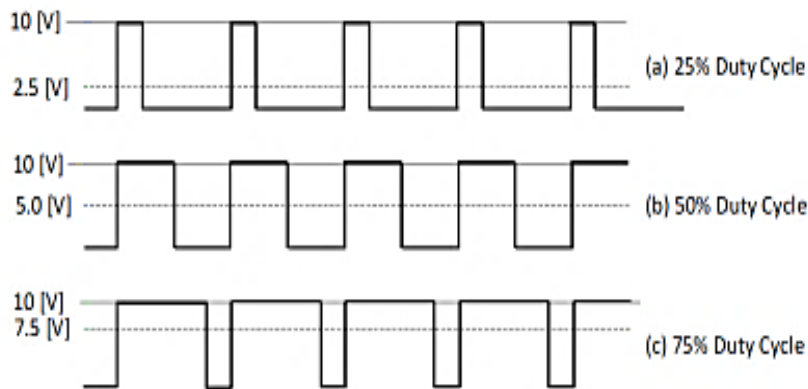
We also mentioned the properties of DCI engines, where the input voltage controls the speed of the DC engine. Thus, ways are needed to change the DC engine input voltage to control the DC engine. However, the built-in computer cannot connect directly to the DC engine because the embedded computer output pins "are limited, and the resulting voltage is no different.

Therefore, the compact computer uses the motorist as an electric power device driving the engine. A motor driver is an electronic circuit with power containers or transistors (Toshiba Inc, 2001). The motorist receives an on/off signal from a built-in computer and drives the engine. The engine driver is widely used in electronics and robotics as a power transformer. The DC engine driver can switch on/off a DC engine, but a DC engine driver has no function in the speed control method. Turn on/off these keys using Pulse Width Modulation (PWM) and control engine speed by changing the PWM ratio. PWM is a method of modifying the signal by changing the pulse operation ratio at the frequency of a constant carrier signal (period) and pulse amplitude.

The computer controls the ends using a digital output pin with PWM mode. Figure 3.11 shows three different signals from PWM. Figure 3.12 (a) shows a PWM of 25% run. In this case, the output signal is triggered for 25% of 1 cycle and the other 75% exit, so the output voltage becomes 2.5 [V]. Figures 3.12 (b) and (c) show outputs (PWM) at 50% and 75%, and this output voltage become 5.0 [V] and 7.5 [V].



**Figure 3.11 DC motor characteristic diagram (Researcher's work)**



**Figure 3.12 PWM signal of every run cycles (Researcher's work)**

Thus, a computerized computer can control a DC engine using a PWM signal. (Matsuzaki, 2020:997-1007). Therefore, the built-in computer is rapidly shifting. In our study, we solved the problem of the speed of DC engines by connecting the engine with the gearbox to increase kinetic power and reduce the voltage consumption rate in these engines. Additionally, the engine tips are delivered through capacitances to absorb the bouncing voltage and reduce the noise ratio from the turn-on/off processes.

### 3.9.3. Energy consumption

This section discusses the various sources of energy consumption in a typical mobile robot. Based on research and experience, an approximate categorization of energy uses robotic actuators, planned motion, engines and layout, and resources (M.Shahab, 2009:4-27). Table 3.1 displays the power distribution in the mobile robot and the percentage of each component. The predicted force motion is from 0.2 m/s to 0.8 m/s

when the robot is loaded with 9 kg. The sensing power was evaluated between 20 and 80 Hz. The component's minimum and maximum percentages are calculated. The minimal percentage is where some components use the least power, but others use the most here is how to calculate the maximum motor power percentage: The motor output is 10.6 Watts. The least power for the other components is  $0.58 + 4.6 + 8 = 13.18$  Watts.  $10.6 + 13.18 = 44.6$  per cent. Our calculations show that robot motion consumes up to 44.6 per cent of total energy (S. Liu, 2011:43-48).

**Table 3. 1The power breakdown of pioneer mobile robot (Y. Mei, 2005:492-497)**

<b>Component</b>	<b>Power in Wet</b>	<b>Percentage%</b>
Motion	2.8 to 10.6	12.1to 44.6
Sensing (sonar)	0.58 to 0.82	1.9 to5.1
Microcontroller	4.6	14.8 to 28.8
Embedded Computer	8 to 15	33.3% to 65.3%

Small mobile robots commonly employ DC motors as green motors. Overall, the robots move around on wheels. DC motors drive the wheels. Therefore, DC motors power the movement. Typical DC motors supply current or voltage to the corresponding circuit (driver circuit). Therefore, battery usage is connected to the required electrical signal consumption. The DC motor is a physical system; power loss occurs due to load friction, inertia, or the electronic circuit controlling the motor. Construct a better engine with a more efficient design to compensate for frictional losses. Reduce the input power supply to the motor to reduce dissipated power (loss). The engine's rotational speed affects power usage. Power management requires an engine's variables, such as speed or acceleration, to relate to the input power. Due to its kinematic nature, this robot's behaviour must be modelled independently. Power consumption increases linearly with speed for some engines.

## CHAPTER IV

### FINDINGS & DISCUSSIONS

#### **Design and Implementing the Depth Image, HCMR, CNN**

In this section, we will review the methods we used based on scientific scales studied in the previous chapter, through which low-cost depth sensors and the HCMR robot were designed and operated and the new dataset built using the HCMR robot. Moreover, the steps are assembling hardware and software to work in an integrated way to complete the construction of this new data. We provide a detailed explanation of all parts and ways to control them. In addition, we offer steps training processes to the proposed module using CNN designed in the previous chapter to achieve the autonomous navigation of mobile robots using depth images and control labels assembled using a new dataset.

#### **4.1. The Steps Build Depth Sensors, HCMR, Dataset and Training the Model Using CNN.**

1. Navigation capability is our most important feature during the mobile robot design process. The goal is for the robot to travel from one location to another in a known or unknown environment while considering the sensor values to achieve the intended goals. It implies the robot must rely on its other capabilities, such as perception (the robot must utilize its sensors to gather useful data), localization (the robot must know its location and configuration), cognition (the robot must determine how to fulfil its objectives), and motion control (the robot must calculate its input forces on the actuators to achieve the desired trajectory). Most of the time, the mobile robot cannot follow a direct path from its starting point.
2. A mobile robot cannot always take a direct path from the initial moving point to the specified point, so we need to draw the marquee through which the robot must move, which entails using motion planning tools. The result prevents the robot from colliding with obstacles encountered during its movement. The ability of a mobile robot to navigate through a known or new environment to reach a goal without colliding with any obstacles is referred to as robot navigation. Calculating a trajectory for the robot's movement from its current location to the target is required. The procedure necessitates using a map, a desired location, and

the robot's current position as determined by sensors or another positioning system. (W. K. Sleaman, 2021:2-26). A cleverly designed motion planner should also be able to determine the type of collisions between a robot and obstacles in the area where the robot moves, which can, with the audacity of control, change direction or stop before a collision occurs.

3. Collisions can be avoided with the use of obstacle-avoidance algorithms. Therefore this study presents a robot that can perceive and avoid obstacles in indoor environments using data for low-cost optical sensors operating using the computer vision system. In addition, the robot is guided by the mentoring marks learned by the model trained using deep learning. After our analysis and study conducted through Chapter III, the methods used to build the stereo system from low-cost photo sensors are done through homogeneous web camera calibration methods through which the images it produces are purified. It is then used to build depth images representing inputs to the deep learning network CNN.
4. As well as to study methods of building and controlling a mobile robot at the lowest cost and low energy and methods to build a deep learning network used to extract properties from images and perform classifications of desired targets for the network. Therefore this section details the work and achievements we aim to achieve through this study, which involves integrating computer vision with deep learning technology employed to build a mobile, independent robot that can navigate in an indoor environment that is unknown to it.
5. This step is accomplished by training our model in a new dataset consisting of depth images and control labels through which the model is trained to learn the robot orientation mechanism on independent navigation depending on the situations in which it is trained, which means simulating human behaviour in motion and avoiding obstacles. Where this dataset consists of two parts:-
6. The first part represents the depth images of the scenes that came across the real robot as it moved through building a new dataset.
7. The second part is the control label, which represents the behaviour accompanying the scene, i.e., the robot's movements to guide its movement, which includes moving forward or turning to the left and right sides and to different degrees that it will learn through its new dataset. In order to accomplish this task, a real robot has been created that can carry a laptop and add it to its

control and power-fueled equipment. We use depth images as semantic inputs in deep learning. These images are built through low-cost, lightweight, small-sized sensors that reduce the robot's physical cost and the computation needed to guide its movement in the internal environment. These objectives require action to:

- 1- Building low-cost sensors done by creating a stereo system that includes cameras, which in its work simulates the human vision system. But the weakness of this system, which is a strong point at the same time, is that we used low-cost cameras (web cameras). So we have to do preparation's which include on his calibration processes this type of sensor , reconfigure an image that is taken , purified by removing the aberration , distortion in which by extracting his matrix the internal and external properties of the cameras obtained through software that depending on algorithms in the computer vision system. Eventually we were able to create a stereo system and its images are pure through which deep images are create at another stage.
- 2- The construction of an animated robot consisting of hardware I and software parts where it works together through a control unit, which it represented to Arduino Uno type. This module link the robot to the laptop, control the rotating the wheel engine player unit installed on the base of this robot that controls the movements of the robot by directing the data sent to it by the laptop connected to the control unit through the USB port. In addition, the robot is equipped with software that operates the physical parts to robot with the laptop computer as it works together as an integrated unit that accomplishes the control of robot movements that require guidance in diverse environment. The procreated stereo system was add to the main robot base. This linked to the laptop unit's controller to become an integrated system that conducts motion processes and stores the new dataset of depth images with labels control through compatibility and synchronization with the depth images produced by the new sensors when the robot move in the environment and the scenes facing it.
- 3- This real robot is design to create a new dataset to train the model through a supervised deep learning algorithm that offers full detail it in this class. Building new training dataset, the main objective of this study is to build a mobile, independent robot that can navigate in an environment without using maps or advance data on this environment or programming the behavior of the robot

depending on the situations and states encountered during navigation. which requires building a dataset that mimics the real environment of robot movements and this data is obtained by moving a real robot that is we designed in navigation through diverse real environment .Where it collects deep images with control label 1 synchronized with its movement or behavior depending on the scenes facing the robot, we perform to guided and control this robot by manual

4- This part includes the final stage by which we use the new dataset created after it separate into four sets of two training groups and two test sets with a ratio of 30 to 70%. These totals include depth images and control marks (consisting of five markers). We use this data in deep learning to train our layered model. Where the signs of control are explore for each scene facing the moving robot and eventually our model learning on navigation in unknown environments and without maps.

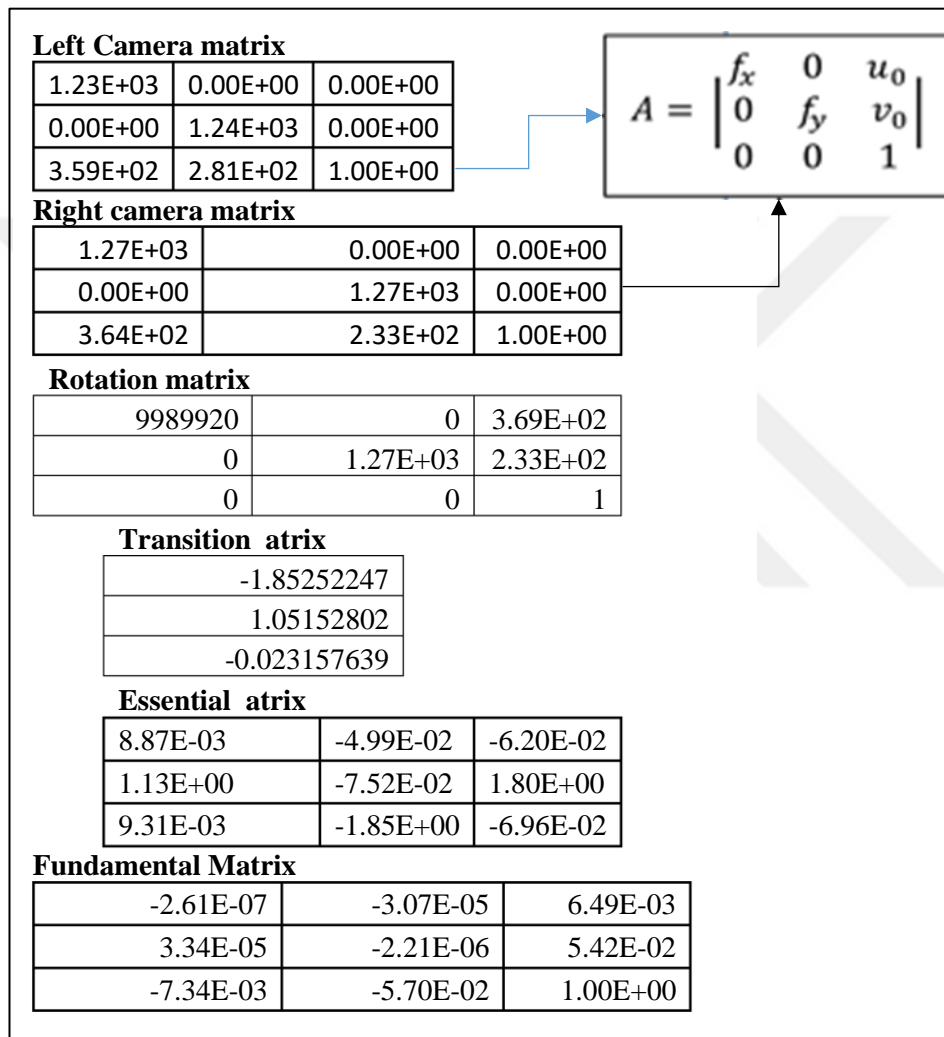
The following details the work and construction:

#### **4.2. Visual Sensor Structures**

The visual sensor structure is a stereo system that includes hardware representing web cameras, which are low-cost and energy sensors. A stereo is a system that simulates human visual structures where each camera represents a human eye and separates this camera at a distance of 6 cm, the same distance as the human eye. Those cameras were installed by measuring the distance and direction of the camera and adjusting it with correct and accurate images shown in Appendix 11, figure (A). Then the stereo system is designed, and the calibration process is performed on this camera by using images of its chessboard that are taken and stored through the stereo system at different angles and directions. Camera calibrations are required in many computer vision applications, resulting in a projective digital shift between the object's location and the image plane. Measuring equipment with cameras allows users to recreate 3D images via stereoscopic endoscopy.

The reconstruction quality depends on the digital image obtained by the cameras used for the experimental calibration. Techniques to improve the calibration process have gotten much attention. To estimate intrinsic parameters, employed computer methods, for example, the suppression of non-linear optimization and removal of noise and distortion, to mention a few recent examples. Numerous studies have evaluated the

accuracy of these new approaches. Also, the impact of calibration target precision on parameter estimate quality has been studied (P. D. Lin, 2007:3012-3022). The Open CV Vision Library was utilized to construct the algorithms. A free, open-source program that includes basic and advanced computer vision functionality. This project utilized a Python library built in this language. Enable the stereo camera to take a left and right images simultaneously and use computer vision algorithms to extract two internal and external matrices for those cameras.



**Figure 4. Information obtained from calibration apply. (Researcher's work)**

This stage is accomplished by installing a base on which a chessboard picture is placed. This base contains joints that can rotate to turn in all corners and directions in the form of a hemisphere and move this base at manageable distances from the base to which the stereo cameras are attached. This base created the 23-image from the chessboard at different angles and distances of the left and right cameras (stereo system). The

Open CV includes established routines to find these matrices using the RANSAC algorithm. Calibration requires a homogeneous calibration item with easily identifiable features. The stereo camera calibration algorithm was utilized in this study. See Figure 4.1 for an example of a calibration technique that estimates intrinsic camera parameters and stereo parameters.) Stereo correction is achieved by aligning two images on the same plane (see illustration below). Produce the internal and external matrices cameras and use this matrix to remove the aberration and distortion of the stereo system (left, right cameras), The aberration and distortion are found in the web camera we used in my study using the internal and external matrices for those cameras and the Open CV library in Python. After the calibration, we move on to other programs using computer vision to construct the depth images of the scenes facing these cameras.

### **4.3. Depth image design**

Using Correction to improve dependability and processing efficiency from an in-depth perspective. A misaligned or mechanically defective camera requires this stage in the protocol. For reliable results for interior navigation, the unique stereo setup employed in this research had similar misalignment. The sum of absolute differences (SAD), the sum of squared differences (SSD), and normalized cross-correlation (NCC) were used to quantify the match. Small windows SAD is used to locate the matching blocks between the left and right images. This technique can find good stereo matching between two photos. The algorithm finds the low correspondence better with high texture content, but only sometimes. The algorithm contains three stages:

1. Processing photos to adjust brightness
2. Enhancing texture content
3. Searching for correspondence using a sliding SAD window of Post-filtering of discovered matches to eliminate bad correspondence determines it.

The algorithm's speed depends on the window size (SAD) and the post-filter threshold. Larger SAD windows yield worse results but take less time. The quality of the findings and the algorithm's speed are adjusted by the window size chosen during the algorithm's execution, indicating that this parameter is application-specific. The window size of the algorithms utilized in this project is 5x5. The correspondence

search algorithm also has minimum and maximum tolerances. These two numbers define the stereo algorithm's search volume in 3D. The stereo camera correspondence algorithm produces a grayscale image whose intensity is proportional to the contrast associated with the pixel's location. Shown in Appendix 11, Figure (B) is a scene disparity map.

#### **4.4. Building Hand Controller Mobile Robot (HCMR):**

This important part of our thesis, which includes two parts:

- A .The hardware part containing the devices use to run and control mobile robot.
- B .The software part consisting the software used to manage and control hardware equipment.

##### **4.4.1. The Hardware Parts**

This hardware and devices are consisting of the following:

###### **4.4.1.1. The Main Base of HCMR**

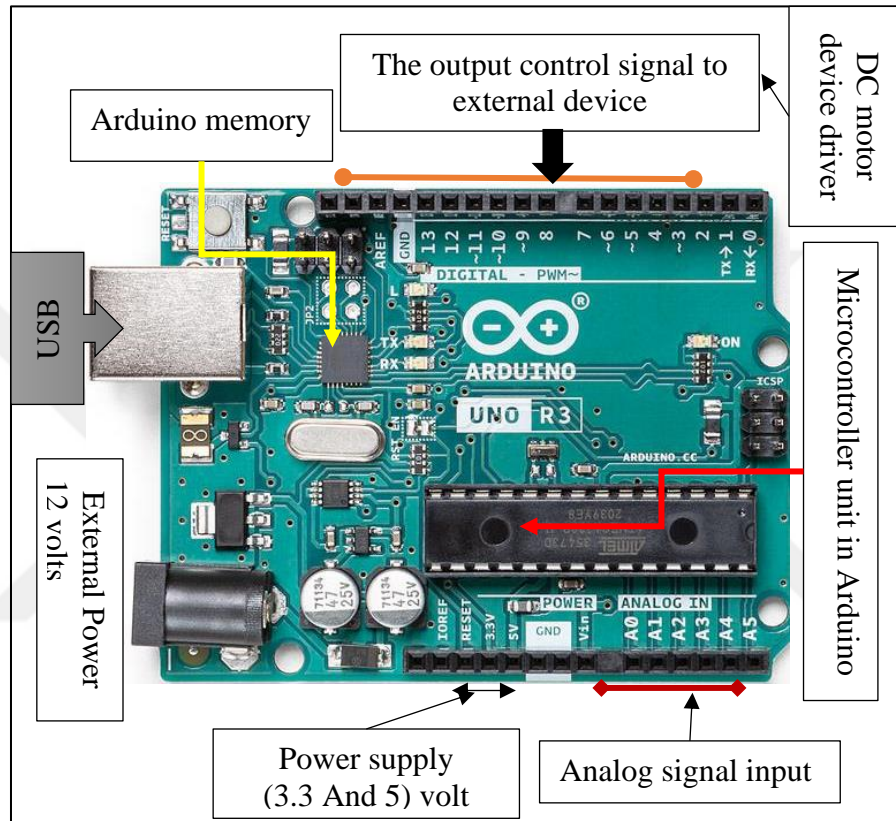
The main base of the robot is the part that is based on all parts of the robot, and in our study, the main base is defined as a rectangle that enables us to place the laptop with the other devices and their size. 40 x 30 cm This base is designed to reduce the energy costs needed to move the robot, in addition to enabling the robot to add a laptop on its base weighing about 7 kg and other parts of the HCMR robot. Figure A in Appendix 13 illustrates the design form and the true shape designed for the flexible movement in moving at different angles.

###### **4.4.1.2. Sensor and Vision System:**

The Vision System structure of the robot, which simulates the human eye and is installed at the front of the and in the front edge of the robot through a plastic base on which the camera is installed, base is represent with the stereo system, is attached and connected to the laptop through a USB 2.0 cable. This cable work to transmit data between the two cameras and the computer where depth images of the viewer can take. This process can controlled by software that works with robot movement of control software.

#### 4.4.1.3 .Motion Axes in the Robot

The robot (HCMR) is fitted with three geostationary, vertical two wheels on the spin axis and located 5 cm from the front edge of the base HCMR robot. These wheels rotated using a 14-volt DC engine. In addition to the gearbox transmission, it increases the torque of the engines to move the robot forward and back, which generates rotation to the right and left sides, with flexibility forward and back.



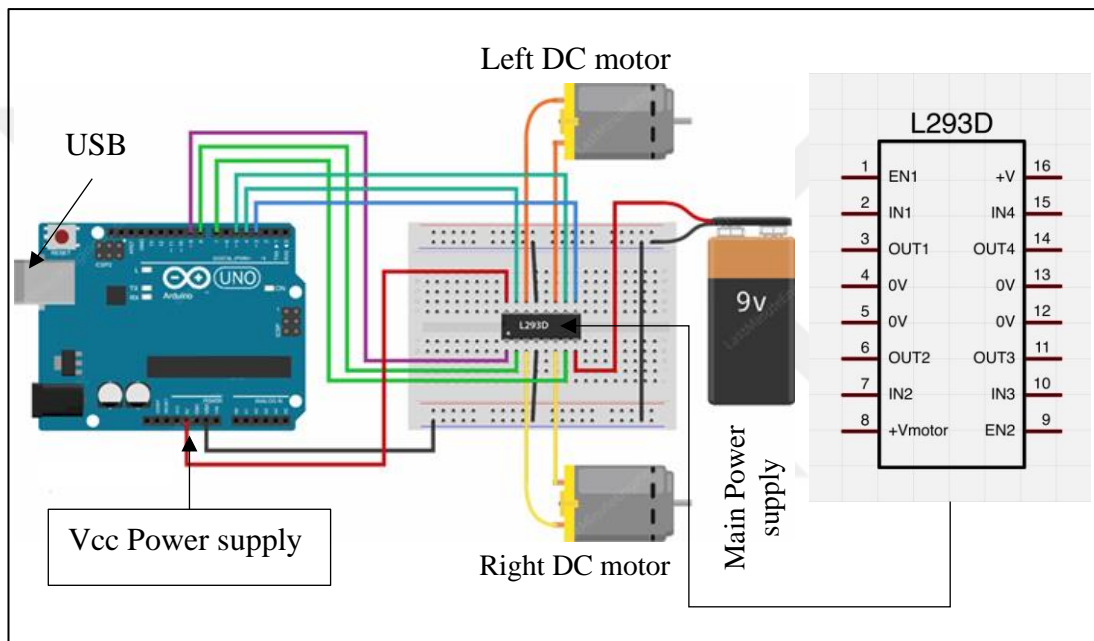
**Figure 4. 2 micro-controller circuit Arduino Uno (Researcher's work)**

Where voltage consumed low. The third wheel is free to rotate and mounted on the main base and in the middle of the rear edge of the base, 5 cm away from it, and does not have any engine on its rotation. As in figure 4.2 and see appendix 5 it shows the types of wheels used front and rear.

#### 4.4.1.4. The Micro-Controller Unit

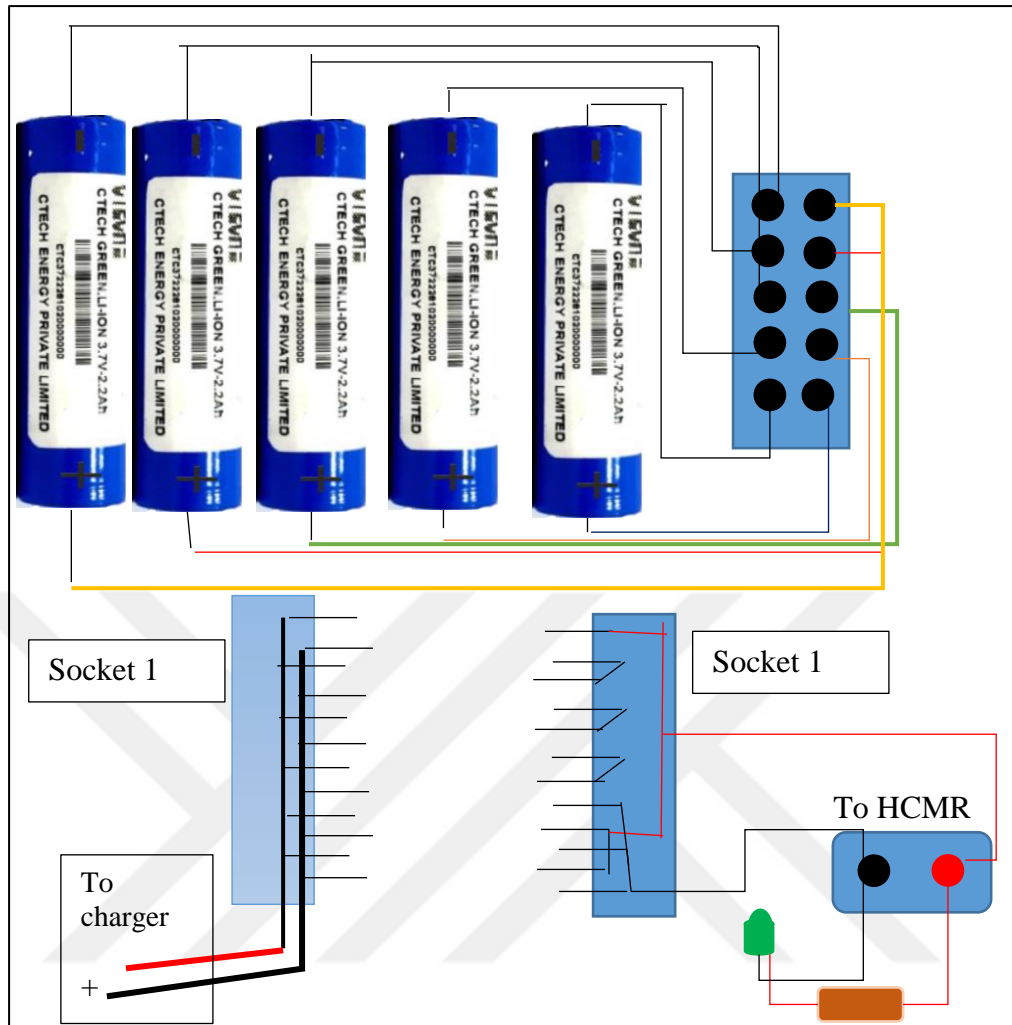
Consisting of the well-known microcontroller circuit Arduino Uno with the Engine operating circuit represents the mobile robot control unit as shown in figure 3.5. Control and power signals are transmitted through low-energy wires. So is the transmission of energy and signal between the different parts of the robot. Control data

is transferred between the macro-control circuit and the laptop using the USB cable, which prepares the 5-volt power needed to power the Arduino. The robot's movement is controlled using a control label sent by the computer to the control unit. It is done through the software developed for this purpose, through which the rotation of DC engines is controlled, called the Engine shown in Figure 4.3. The Operating Circuit, whose function is to reduce the load on the control circuit because the rotation of these engines requires 14V voltage and high current to be provided in the Arduino circuit, But the Aduino signal becomes a mere signal that opens a gate that doesn't consume a high current that can shatter the unit of control.



**Figure 4. 3 the device driver L293D with Arduino Uno (R. Farkh,2021:423-235)**

We use L293D it is a very useful chip. It can actually control two motors independently. Through this electronic circuit, both DC engines move in both directions. That means we need four lines per two-line engine. The first line runs the engine against the opposite direction.



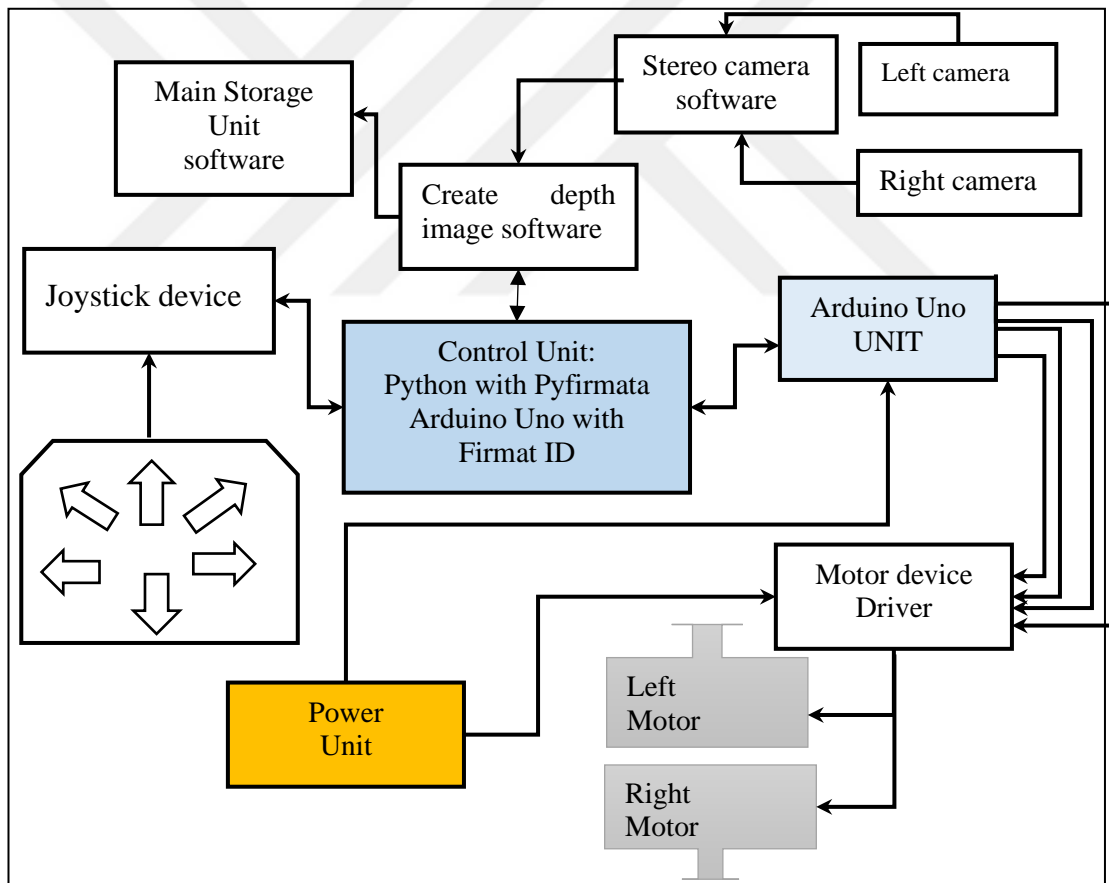
**Figure 4. 4 power supply unit of robot HCMR (Researcher's work)**

#### **4.4.1.5. Power Unit in Robot**

This unit consists of a group of 6 lithium batteries. Each 3.7-volt battery voltage was linked four in parallel and two in a row simultaneously to increase the current value and to produce a total power unit voltage of up to 14 volts and a current of up to (5) amperes. An electronic circuit has been built through which batteries are easily recharged without battery destruction as well as to protect against explosion during charging, shown in Appendix 3. This circuit is installed on the base of the robot. Another electronic circuit that sends a signal with energy to praise the robot's operation was also built it. A switch to cut power to the robot was added, as in Figure 4.4.

#### 4.4.1.6. The DC1 and DC2 Engines

When the robot was designed in our study, DC engines were used are powered by a maximum voltage of 14 volts. We used small motors in our study, but the number of cycles was high. These engines were connected to the wheels through a gearbox transmission designed to reduce the speed of rotation and aim to increase engine torque and reduce power consumption while moving the HCMR robot carrying the laptop and other devices. These engines are used to move robots depending on the period of rotation of the DC1 and DC2 engines and in both directions. The front motion of the robot needs to rotate the two engines simultaneously for a period of 8 seconds, which can be changed, but after practical action, we have concluded that the best period of time is to move forward so that the movement of the robot becomes smooth, flexible, accurate and reduces energy consumption.



**Figure 4. 5 the control unit between PC and another device in mobile robot (Researcher's work)**

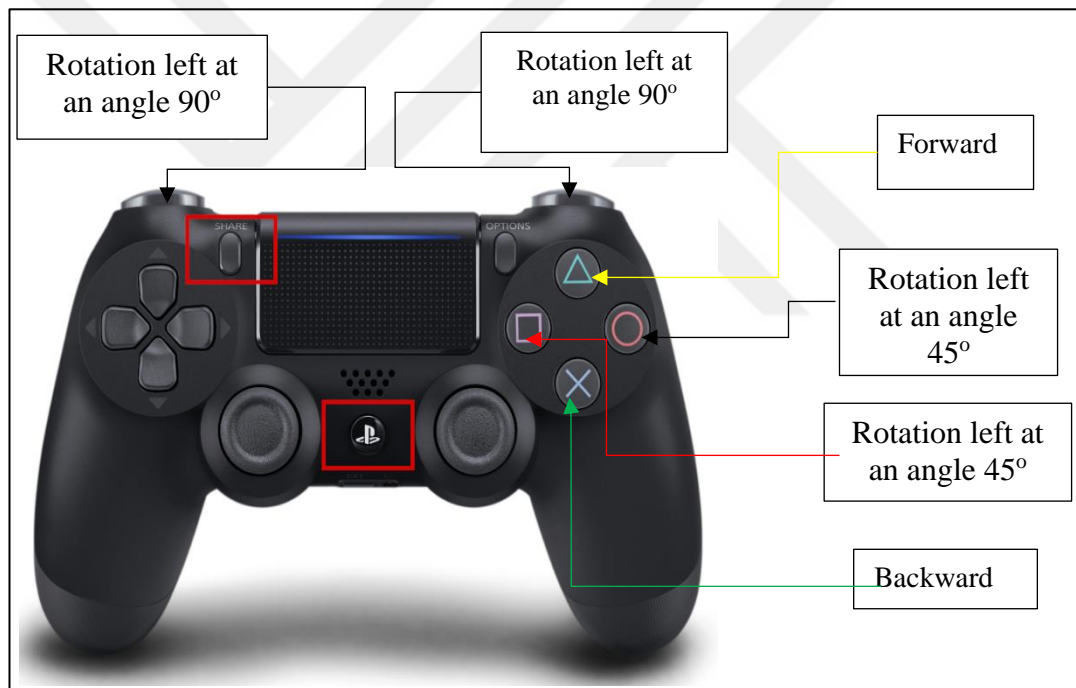
The robot movement can made backwards, in which we do not use control marks, it done by turning the two engines back together in the same period. This movement is

use in emergencies and that the software does not record depth images or control signs in this case because it is not a training data input as in the show in figure 4.5

#### 4.4.1.7. Joystick and Control Key

The Joystick is a device that enable to controls the movement the targets in a particular game after a link it to a computer, iPad, or Xbox, among others .We used

Joystick iPad 4 to as in figure 4.6 because of its high specifications and to be the laptop control stick that can controlled without Wire by use Bluetooth connectivity, which facilitates the control robot mission remotely, through software and libraries added to the Python language. We were able to customize the keys in the joystick so that each executive key works a specific procedure by the robot via the Arduino control unit and the motors associated with the robot wheels.



**Figure 4. 6 the keys used to control the robot HCMR in joystick of the iPad 4  
(Researcher's work)**

#### 4.4.2. Software parts

These sections include:-

1. Software for creating depth images consisting of stereo system creation processes, calibrating cameras, refining the image from aberration and distortion, creating depth images.

- 2 .HCMR robot movement control software.
- 3 .Depth image recording software and signs of control where the scenes facing the robot are HCMR as it wanders.
- 4 .Joystick Keys Identification and Connecting Software in HCMR Robot Control Software.
- 5 .The main programmer that manages and operates the four programmers.

#### **4.4.2.1 .The software for creating depth images**

This software works together to accomplish depth images by using the stereo system, which is the base on computer vision algorithms and can detailed as follows :

1. Software extracts the basic internal and external matrices of both cameras, which work by using images taken by chessboard after being install on the mobile base designed for calibrating the cameras, which have the ability to move at spin angles in the form of a hemisphere and after obtaining these matrices where these matrices are stored in a text file.
2. Software that uses matrices obtained through the software in stage A. Through which we are able to purify the image taken by the two cameras of the viewer, by removing the distortion and aberration of these images, which depend on the software provided by the computer vision after rewriting it to deal with new cases.
3. The depth image building software for the scene facing the stereo system for cameras. After obtaining the depth images, we filter the depth images by using digital image processing software to remove the patches that appear in the depth images and improve the purity of the resulting image.

To reduce the computation processes used to explore objects in the image using deep learning that will be the basis for training the proposed model to produce the learning process. The a result, obtain control labels for each scene facing the robot in a real environment, i.e. in the general sense directing the autonomous mobile robot to navigate in environments without the use of advance maps, which is the primary goal of our study.

### 3.4.2.2. Software Controller to HCRM

The software includes Arduino operating software and laptop connectivity by downloading the Arduino IDE app and libraries needed to activate Arduino's connection with the computer through the Com4 communication port. The libraries in operation are also loaded to connect Arduino to Python using the fermata library, which must be made available to enable the Python language to send and receive control signals on the output and interventions of the micro-controller (control unit) of Arduino Uno. This program, built in Python, sends five signals, and each includes three control signals for DC1,2 engines. Two of these signals are used to operate, and one is to stop motors, depending on the movement required by the HCRM robot. The number of rotating robot engines is controlled by the period specified in the control program. It has also been introduced through the work of physical parts and as shown in the layout in Figure 4.7.

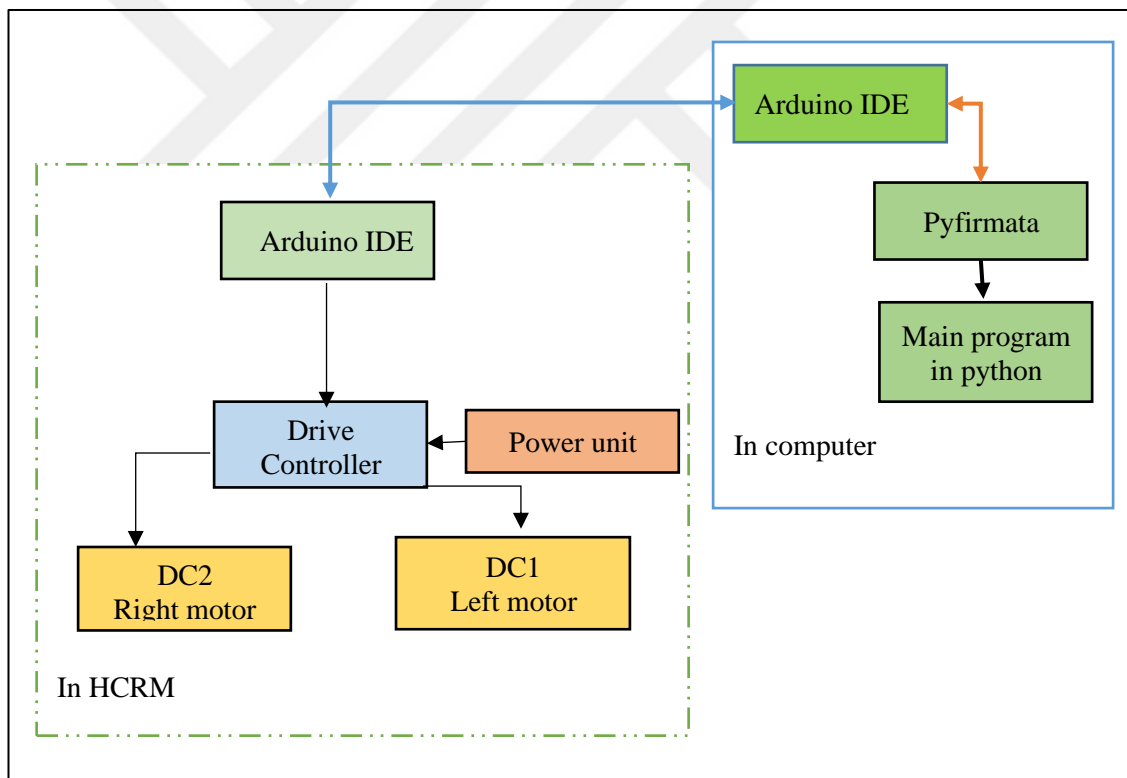


Figure 4. 7 the control program in computer and Arduino (Researcher's work)

#### **4.4.2.3. Depth Image Recording Software and labels Control**

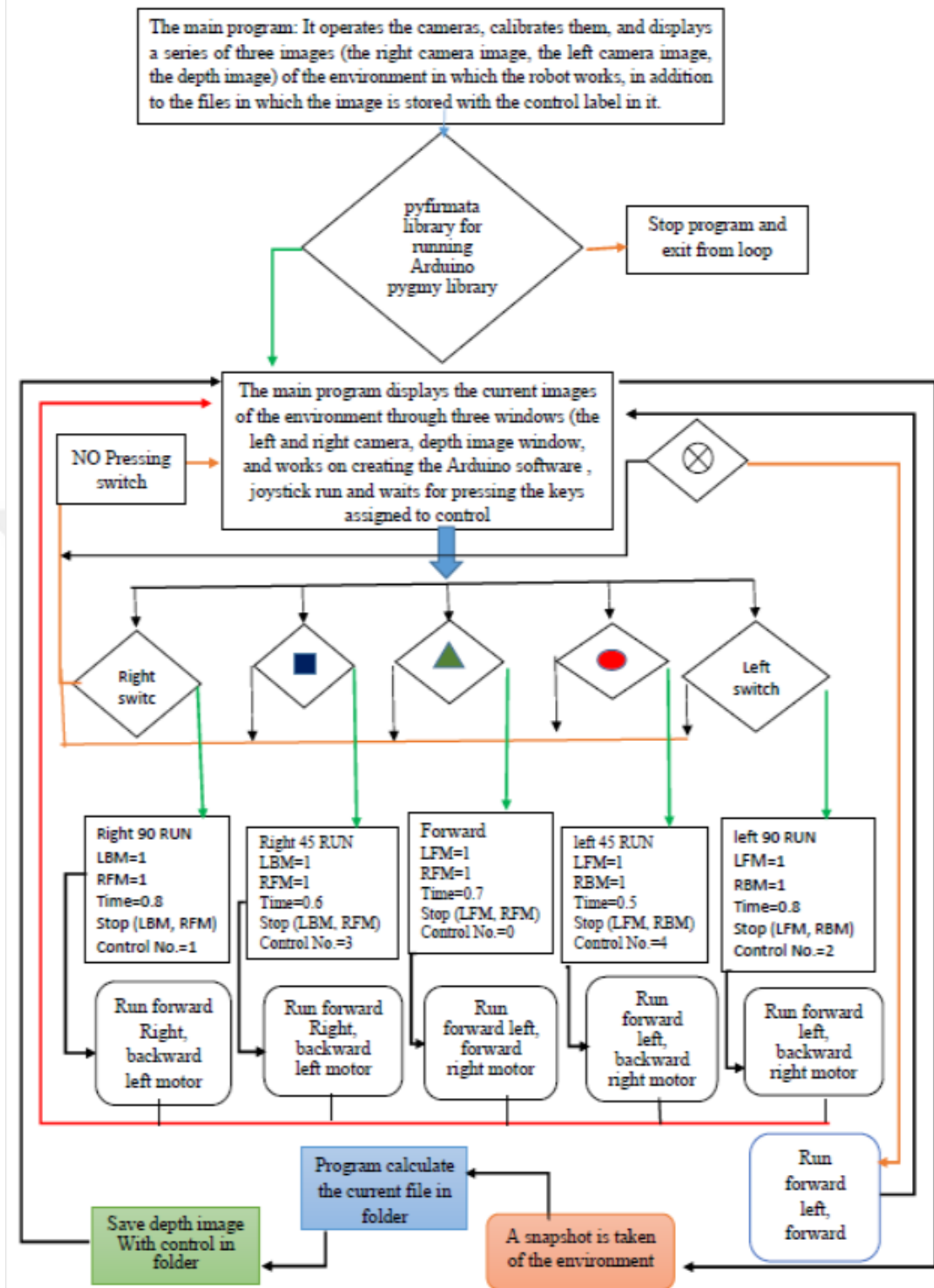
These programs do depth image taking, connect them to the control labels, and store them in the computer to be assembled. The depth Images are matched by scenes facing the robot HCMR as it wanders: After the stereo system consisting of web cameras is connected and operated on a plastic base according to the stereo system work requirements we referred to in Chapter 2 of this thesis. Through this program, we run the two cameras in a live video series format, where this program is based on showing three types of video chains of the reality facing the cameras. (Left and right installed on the base of the HCMR robot) through two windows of the left and right landscape and a window of the depth image in the main program screen. This program's function is to send a copy of the depth images to the training data set building program to be linked to the control labels, which means the robot's behaviour now the image is being taken at this moment. This software is linked to the main software in the laptop and transmits data from the stereo system to the computer via USB 2.0 wires. Arduino has nothing to do with it but is a function of the main software.

#### **4.4.2.4. Joystick Key Identification**

The software is part of the main program whose function is to convert the Joystick keys associated with the laptop into control commands depending on the switch being pressed by the system user on the Joystick keyboard. After the corresponding blocks of the subprogram allocate the work of these keys, each key installed in Python is called the Pygame library. This program controls the main program and waits for keys for the main program provided by the core of the main program. The main software senses which key is pressed by the user. It operates on the terms of which the work of this key is assigned according to the code that corresponds to it within the customization table. This program does not implement Joystick's keys if it is not allocated to the Joystick apportionment table.

#### **4.4.2.5. The Main Program That Manages All Program**

The software brings together the four platforms that we have already talked about in the form of subprograms that are called upon in these programs according to the function of each program that these programs have programs that contain a range of subprograms and whose characteristics called upon and inherited by their branches.



**Figure 4. 8 Outline describing Major Programmed and Robot Control Subprograms (Researcher's work)**

Therefore, it is called in this program, and a software plan has been implemented to manage the software and activate and stop libraries that we do not need. To call as

much as possible from subprogram, and eventually, we will have data that's many depth images files a type of this file is (png) with serial numbers starting at one after a program survey each storage unit in the designated computer spaces for this purpose. It works to calculate the number of files that have been stored in it. The meter is increased by a new number in each storage process so that it is stored at any time without fear of repeating the name of the files in the storage place. This subject leads to great confusion in the data, and in this way, there is clarity in the data storage. In addition, each new environment's storage locations are changed and separated from another environment's storage data. It means that there is a disciplined mechanism for storing data produced by the movement of the HCMR robot in a predetermined environment. Figure 4.8 show that.

#### **4.5. Building a New Dataset**

The new training dataset is used to train the proposed model using deep learning is a very important part of the tasks we wish to accomplish through this mechanism. In addition, through which we train our proposed model, we learn the target model we want to reach through a deep learning application that uses this data as input to its multiple layers. It enables us to equip robots learned to enable independent navigation. This achievement will enrich or spare us from equipping the moving robot in a given environment with maps, creating corridors, and programming it for every task it accomplishes. This process is very expensive, especially when the robot is used to perform navigation in an indoor environment where GPS stops. Building dataset requires serious work for these data to simulate the real world and achieve the goals for which these data have been built. On this subject, it is recalled that the quality of the data we need in training a mobile robot in different environments, collecting this data from each environment through the use of the HCMR robot that was designed for this purpose and whose characteristics we explained in advance. This process is done by storing data collected from each different environment containing two important molecules.

Part 1- depth images of the scenes confronting the robot HCMR as it roamed this environment.

Part 2- is the labels of control, which mean the robot's behaviour during the roaming process.

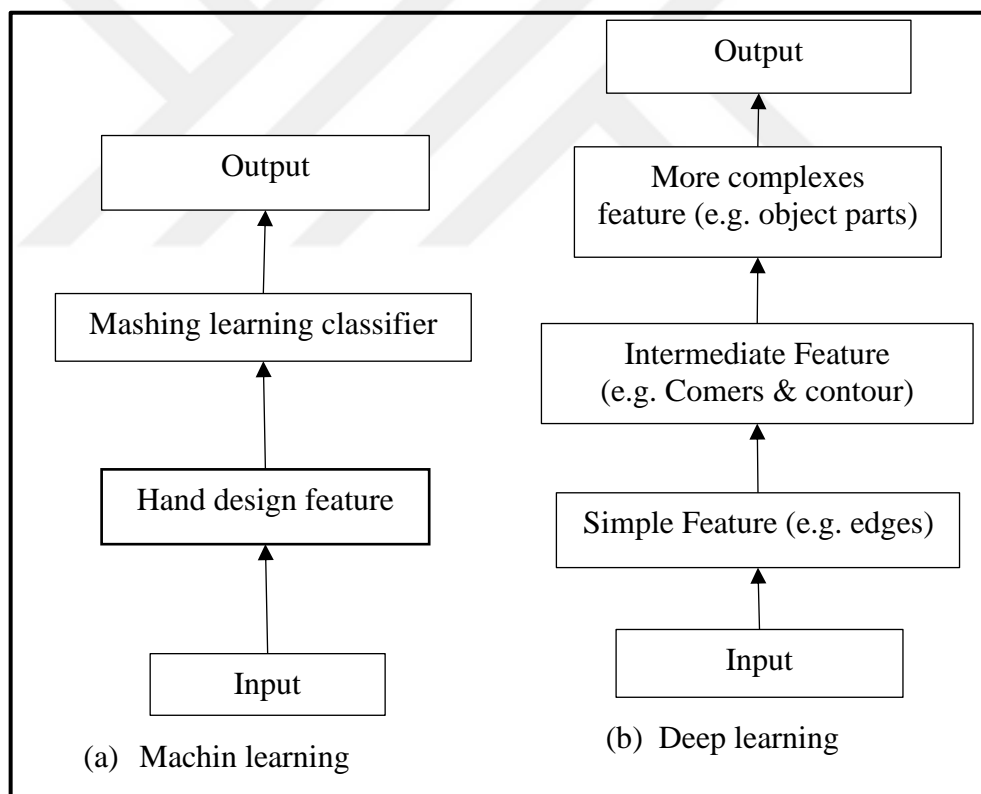
After this process, a program that compiles the data created in each environment into two files is stored in the laptop placed on the mobile robot's base. After this process, this data is converted into another program that divides it randomly to ensure that we are impartial in selecting the training and testing sample in deep learning technology. It is done for what has been scientifically agreed upon, 70 to 30 per cent for training and testing. The training sample has depth images and control labels for 70% of the total data in the same way the test samples are created but for 30% of the total data. Compatible with the work of a program that trains our proposed model, as follows: (train\_data, test\_data, train\_label, test\_label), it was done through our use of the sklearn.model\_selection import library, train\_test\_split use python program . We have obtained beforehand. The training dataset was broken down to be compatible to serve as input to the deep learning process built using the popular Pytorch library in deep learning.

After these processes, another process is performed by which we reduce the size and depth of images to focus on the object in this image by reducing insignificant calculations on image cells represented by image pixels. Through these steps, the outstanding work of our study is evident, which involves building a new training data set. In order to verify the accuracy and strength of these data, we conducted a scientific comparison of the results obtained from the training of our proposed model on these data, comparing these results with the results of previous research and studies in the same field where the results were identical. It proves that our approach to building the special robot training data set is to build the robot to navigate independently based on deep learning and computer vision.

#### **4.6. The Deep learning Techniques**

The DL is a modern field of ML. This kind of learning has aroused the interest of researchers recently. We can define deep learning through the structures it includes, which consist of multiple layers between the input and output layers. These layers are invisible and called hidden layers that can be used to learn different features at different levels of abstraction. Higher-level features are identified in terms of lower-level features of biological data in their simplest form. DL algorithms offer the unknown structure of input distribution, during which new representations are frequently detected at multiple levels. It has been demonstrated by researchers using deep

learning in building a pattern recognition system. A traditional ML system that performs pre-processes is performed on data entered into machine learning by using manual engineering to extract features from primary data (e.g., pixel values in an image), then inserted into the learning algorithm, forming an effective internal representation or feature vector that enables the system to learn from classification, recognize patterns in inputs or classify them. However, a deep learning algorithm enables practically feeding raw data without pre-processing; it extracts features first or creates the characteristic vector. Furthermore, DL algorithms can quickly discover the relevant, exact set of characteristics. DL entails learning to extract these attributes automatically throughout the training phase. Rather than manually developing a set of rules and algorithms to explore features from raw data. Therefore, the problem he addresses with his hierarchy of concepts is deep learning, with each idea built on top of the others.



**Figure 4. 9 Compared to traditional ML in b(a) and DNN in (b)**

**(M. A. Wani,2020:3)**

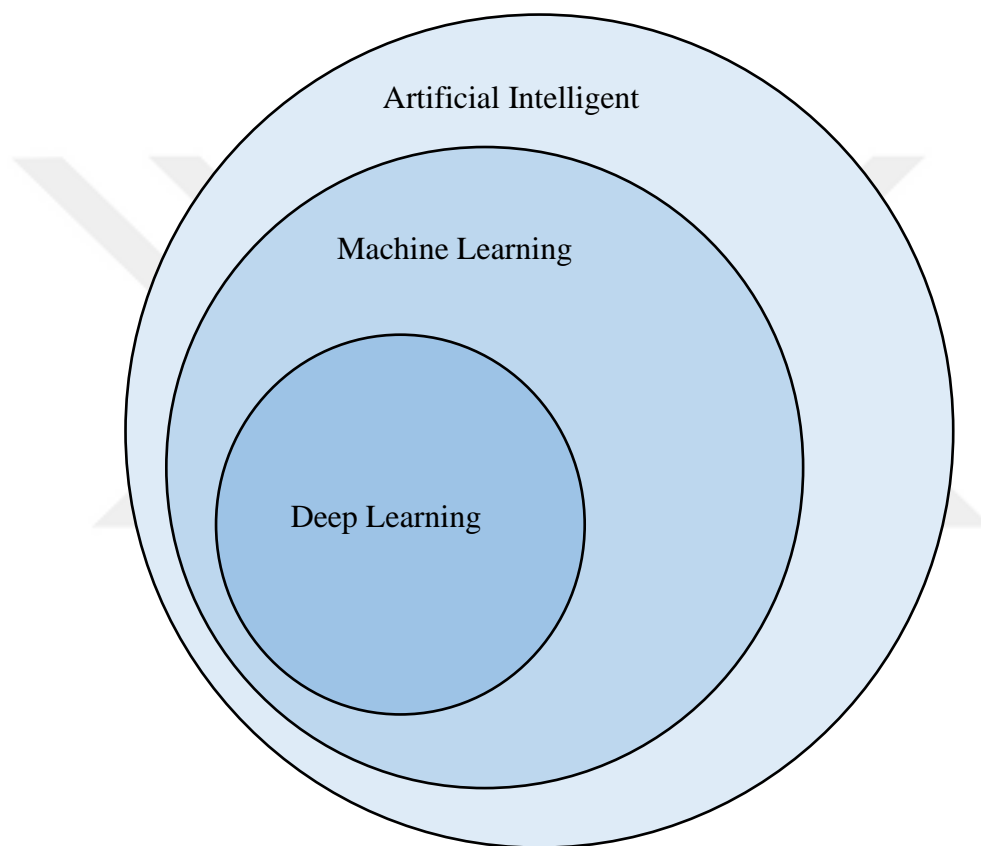
The base layers Are the model's lower-level layers that define several basic, initial descriptions of the issue. In contrast, the upper layers create the results of the lower

layers to generate more complicated notions. The image density values can send as input to the deep training method given the image without preprocessing. Then, many hierarchical hidden layers perform feature extraction operations on the input image (properties). These concealed layers are stacked one on the other in a hierarchical pattern. Initially, the network's bare minimum layers only investigate edge-like regions. It works by specifying angles (where edges overlap) and edges that use these edge regions (the outlines of objects). Top-level layers mix corners and outline in the following layer to produce more complex "object parts" in the subsequent stratum. The DL benefit is that these layers of the feature are not created using hand. Rather, they gradually taught with the help of data and a general-purpose learning algorithm. Finally, the output layer is defined as the image output layer. All other nodes in the network directly impact the output obtained in the output layer. It is known as recursive learning because every network uses the outputs of the earlier layers as "building blocks" to build increasingly complex conceptions in the higher levels, as seen in Figures 4.9 and 4.10.

Clearly, we compare the standard ML method depending on handmade characteristics to the DL based on spatial structure. In DL, meaningful representations are generated by focusing on constructing complex maps from a sequence of simple mappings. Deep learning is a learning process that is accomplished on multiple layers of representations that are highly relevant to the input data to the learning network. The multiplicity of layers used to simulate information and extract features from input data is determined by the depth of the model in most cases, where the current DL uses as many layers as hundreds of consecutive layers included in the training data for learning. However, when comparing these processes with traditional machine learning algorithms, we observe that these processes are performed on learning only one or two layers of data representation. These methods are known as shallow learning.

As a result, deep learning and machine learning are important sub-areas of artificial intelligence (AI). Figure 4.10 shows the relationship between artificial intelligence, machine learning and deep learning. The DL has the potential to learn consecutive layers of representations via sub-models that are organised in the form of layers layered on top of each other. Because a deep learning network typically has more layers and parameters, It existed almost 40 years ago but was used sparingly. The deep

learning algorithm needs high calculations due to the inadequate computer infrastructure (hardware and software) to execute calculations representing processing speed. Moreover, it was a drop in accessible data at the time. As traditional forms of entertainment have become less popular and neural networks, only recently have deep networks emerged that have produced astounding gains in the field of speech recognition as well as computer vision. In our study, we describe the factors that accelerated the development of deep learning, its applications in computer vision and robotic orientation, as follows:



**Figure 4. 10 Relationship between AI, machine learning, and DL**

**(M. A. Wani,2020:6)**

Machine learning practitioners have focused heavily on extracting informative properties from data. Modern machine learning algorithms have previously required decades of human labour to synthesise significant features to identify inputs presently. DL beyond standard algorithms progresses to accurate attainment, where features are learned from data using general-purpose learning technology rather than technologies designed by engineers dedicated to the field. Deep networks have shown significant gains and extensions in computer vision, particularly in robotics orientation. As well

as greatly enhancing machine translation. It has become a successful way for artificial intelligence to detect spoken language (voice) words so that a person can recognise them. It achieved high accuracy in ML modelling procedures and displayed a surprising generalisation power. It gathered experts from diverse academic disciplines. It is now utilised as a guide for making vital medicine, finance, and industry decisions. DL has demonstrated encouraging results in processing and recreating cognitive activities in humans that can be represented by hearing and visuals. Computer scientists have employed massive volumes of data – picture, sound, and video data – to teach computers to perform learning processes. Things that appear natural and obvious to humans, like detecting objects in images, identifying words or sentences, and translating a document into another language, can all be performed by exploiting this technology's immense processing power. It allowed machines to extract text from an audio clip through speech recognition; there was a sufficient dataset to train using machines to accomplish multiple tasks. (M. A. Wani ,2020: 3-10).

#### **4.6.1. How DL Works**

The core function of deep networks is to identify the input to the target by a series of transformations of a collection of layers that comprise the network's hierarchical structure. These layer changes are learned through exposure to training instances. A layer's weights, essentially an array of numbers, govern the modifications it performs to its inputs. In another sense, a layer's parameters are decided by their weights. Learning can thus be defined as a process that aims to find the through weights in all layers included in the network being trained so that input examples can accurately be assigned to the associated goals. Because the deep learning network includes thousands of factors, identifying the best set of parameters is difficult, particularly when one parameter's value affects another's value. One must first understand how far the network's output is generated to train a neural CNN. This scale, calculated using an LF, often means a fitness function. The goal is calculated by subtracting the network's predicted output from the true target value for a specific scenario established in the training dataset. This indicator measures how often the network learned a specific example. Therefore, the training aims to find the weight values that reduce the selected error function. Then the resulting differences are used as a reference via which we try to adjust the network's weights to reduce the degree of loss. The optimized back-

propagation algorithm performs this tweaking, which is crucial and central to DL. At the start of the network, the back-propagation method assigns random values to the weight effects by executing a sequence of random transformations. Initially, the network's output may be far from what we desire, resulting in a significant loss. With each example fed into the grid, the weights changed to reduce the degree of loss. This procedure will repeat until weight data that reduce the objective functions are obtained, showing that the predicted value and the potential target output are well-suited. When the features extracted from the trained network are almost identical to the desired values as much as possible, the network is said to have learned.

(M. A. Wani,2020:6-12)

#### **4.6.2. The Convolutional Neural Network (CNN)**

CNN is commonly referred to as a multi-layered DL system. The biological visual cortex inspires CNNs. This visual cortex is made up of small groups of cells that are sensitive to different parts of the visual field. These neurons in the brain can respond to a wide range of stimuli. Some neurons, for example, only respond when exposed to lines of a specific direction, whilst others respond when exposed to horizontal or diagonal edges. Convolution neural networks arose from the concept that particular neurons perform a specific task.

It has demonstrated excellent performance in various computer applications, including image classification, object detection in digital images, and voice recognition. CNN is at the heart of the computer visions we are connecting in our current project. It has many applications, including self-driving automobiles and autonomous mobile robot navigation. It also pertains to the design of wheels to serve the visually handicapped. CNNs work by extracting local information from the network's input (often an image) in the higher layers and combining them with more sophisticated characteristics in the lower- levels. However, because of its multi-layer architecture.

Training such networks is computationally intensive. When the dataset is huge, this can take many days. As a result, deep neural networks are typically trained on GPUs. CNNs are so effective at visual tasks that they outperform nearly all traditional approaches. We focus on reducing calculations in our study by giving the DL network

with depth images in which the three-dimensional objects in the photos are clear and semi-separate from the background image, facilitating the identification process, and this image is small. (Yann LeCun, 2013:1915-1029)

### **4.6.3 The CNN Architectural Design**

By carefully analysing a structure in a typical neural network, we will observe the overall interconnection of neurons between layers, where the hidden layers are the layers between the input and output layers in the neural network.

The hidden layer consists of several neurons, each linked to all previous layers' neurons. In this scenario, the issue with an FCNN is that the network model is densely connected, which could perform better with huge photos. As a result, for large photos, using a CNN is the best option.

A CNN is a deep neural network structure that handles data with a recognized network-like structure, such as time series data, two or three dimensions as data like images and audio signals, and four dimension data like videos. The convolution has the following primary characteristics:

- Filed a local receptive
- Weight distribution
- subsampling (maximum or average Pooling).

#### **1. The Local Receptive Field**

Each neuron or hidden unit in a classical neural network (ANN), as it is known, that linked with every neuron into the preceding layer or every insertion unit in the network. In contrast, CNN has a local receptive field structure. As a result, each concealed unit can only connect to a small area of input called the local receptive field. To achieve this, the filter/weight matrix is made smaller than the input matrix. Neurons can extract key visual properties such as edges, angles, endpoints, and so on by using the local receptive field ( Schmidhuber, 2015:85-117).

#### **2. Weight distribution.**

The process of using the same filter or weights in all receiving fields in the input layer of the training network is referred to as weight sharing. Since filters used in CNN are smaller than the data entered in the training network, they are applied to each input site in an image. For example, the same filter is used in all local reception areas for input data.

A convolution neural network comprises many layers that work together to achieve various tasks. We note the basic composition of the typical CNN layers that include this layer: "convolutional", "activation Function", "pooling", and "completely connected layer".

We are working to build a comprehensive convolution neural network architecture by stacking these layers. The convolution and activation layers are often placed together and followed by the pooling layer. The completely connected layer is the grid's final layer, through which the required classification of selected targets occurs. The output of the last fully connected layer results in the grades of the input image category. In addition to the aforementioned main layers, CNN may incorporate extra layers such as a batch-levelling layer to increase training time and a leak layer to alleviate the over-fitting issue (Simonyan, 2014:14091556):

### 3. Subsampling (Pooling Layer)

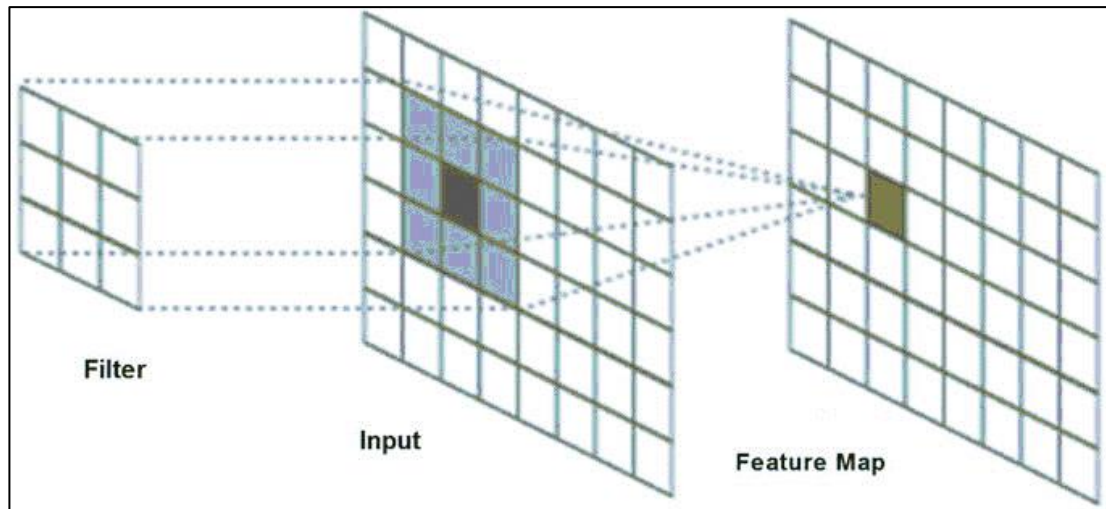
The sub-reduction guest minimizes the spatial volume of inputs, hence lowering network parameter usage. Just a few sub-sampling techniques are available, with maximal Pooling being the most frequent.

- Layer of Convolution:

The convolution layer is the foundation of a CNN that uses convolution rather than ordinary array multiplication. It has parameters containing a collection of filters that enable learning, Often called "kernels".

The primary task of the convolution layer is to extract and recognize features through characteristics that appear in the local areas of the input data that the convolution layer of a similar input image has been applied across the dataset. Where its shape is transferred and converted to the feature map, as shown in Figure 4.11, it uses convolution to get the feature map on each candidate in the layer by repeatedly applying the filter to the entire image subzones.

After adding the term bias, the filter is applied to the input image, and then the activation function (AF) is applied. The candidate's input area is referred to as the local reception area. The reception field has the same dimensions as the candidate (corresponding).



**Figure 4. 11 convolution operation (M. A. Wani,2020:19)**

The descriptor is produced by adding the correction factor and applying a non-linear function AF to the convolution process product. The purpose of the non-linear function is to measure non-linear data that results from applying filters to inputs into the CNN model. Our study summarises many non-linear functions and their uses according to the problems that require the network to solve.

- The kernel/filters:

The weights in each convolutional layer define the convolutional filters, where each convolutional layer may have many filters or kernels. Each filter extracts numerous visual features, such as the edge and corner, by passing them on. With a forward pass phase, each filter evaluated the width and height of the filter's input generation features map.

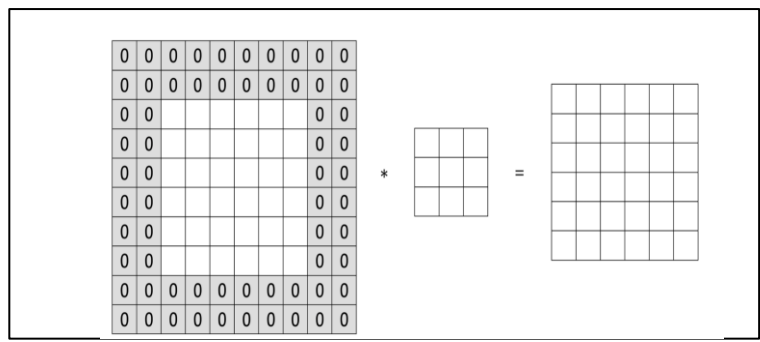
- "Hyperparameters"

A CNN's design incorporates numerous hyperparameters that can be used to alter the model's behaviour. Some super parameters can be used to regulate the output size, while others can be used to change the model's runtime and memory cost. The four most essential hyperparameters in the ConvNet convolution layer are as follows:

A. Filter diameter: Filter can have any size larger than 2 x 2 and should be smaller than the local area size of the input. However, the standard diameter ranges between (11 11 - 3 x 3). The filter's size is fully separate from the input size.B. Filter count:

all numbers of filters permitted. In the first convolutional layer of AlexNet, 96 x 11x11 filters were used. While the VGGNet utilized 96 x 7x7 filters in the first convolution layer, another variant used 64 x 11x11 filters.

B . Stride: The number of pixels travelled simultaneously sets the user filter's local receiver field. One step equals one pixel up and down. The step value must be manageable and manageable. The two tiny steps will overlap the received fields more, but a large step will overlap less, resulting in a spatially varying output size with smaller dimensions. D. Zero padding: Indicates how many pixels in the input image will be full with zeros. To control the spatial size of the output volume, zero padding, which used. If A is the size of the input image, Z represents the value of filter or kernel size kernel, D represents the value of padding that will apply (zero padding), and the S value is the step. Please assume that the input image size is  $IN=120 \times 120$ , the kernel size is  $Z=5 \times 5$ , which means five filters, and we apply  $S=1$  that represents to step, and zero padding was applied. Where if we have this data about input network such as  $IN=120$ ,  $F=5$ ,  $D=0$ , and  $S=1$ . When applying the equation  $([IN -Z+D]/S) + 1$ , we can see the number of feature maps that equal  $([120 - 5+ 0] / 1) + 1 = 116$ . This means that the output size will be  $[161 \times 116 \times 5.]$  show in figure 4.12



**Figure 4. 12 Padding Layer (Researcher's work)**

**4.6.4. Activation Function (AF)**

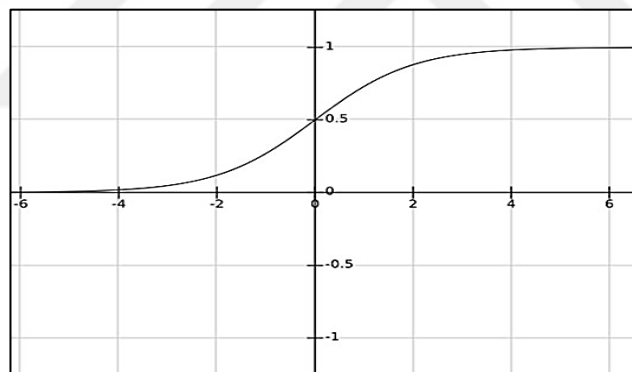
1. After applying the convolution layer to the input data of the network, the output of this process is converted into a new layer called the activation layer. This layer has the activation function used to output the feature map created by the neural network by applying the convolution layer. It takes a layer as input and outputs an activation map. The activation function is used to convert a neuron's activation level into a

layer signal output. Where the neuron outputs in response to a specific stimulus. A squish effect is frequently included in an activation function, Where an activity level of these neurons is extracted into a specific interval between which a given range, for example, between (0 to 1) or between (-1 to 1). Where the activation function must be visible and consistent throughout its execution, it is because CNNs are trained using gradient base methods where these methods operate in the context of their work using differential equations, so the AF must be differential at any time. On the other hand, if we do not use gradient-based methods, the differentiation process does not make sense. Therefore, many different activation functions that used in ANN and some of them are commonly used as follows:

2. **Logistic/Sigmoid:** The sigmoid a function represented mathematically equation in (4.1) :

$$\sigma(X_j) = \frac{1}{1 + e^{-x}} \quad (4.1)$$

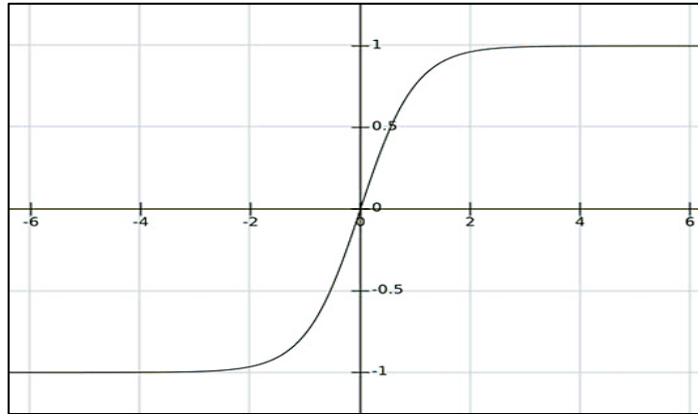
This function can be seen as a S shape, as shown in Figure 4.13. It squashes the input  $X_j$ , where  $j=0,1,2,3,4,\dots$  and outputs into the range  $[0,1]$ .



**Figure 4. 13 Graph of sigmoid activation function(M. A. Wani,2020:23)**

3. **Tanh AF:**

The nonlinear activation method is similar to the "sigmoid function", even though its output is limited inside its value of  $[-1, 1]$ . Tanh has an advantage over sigmoid in that Figure 4.14 depicts the tanh graph, where minus values are negative and zero inputs are close to zero.



**Figure 4. 14 Graph of tanh activation function (M. A. Wani,2020:24)**

4.The "Softmax" Function (Exponential Function):

We use this function frequently in the neural network, it has an important and essential role in addressing the classification of what is required of the network to accomplish during the training process, as this observed in the mathematical equation (4-2):

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}} \quad (4.2)$$

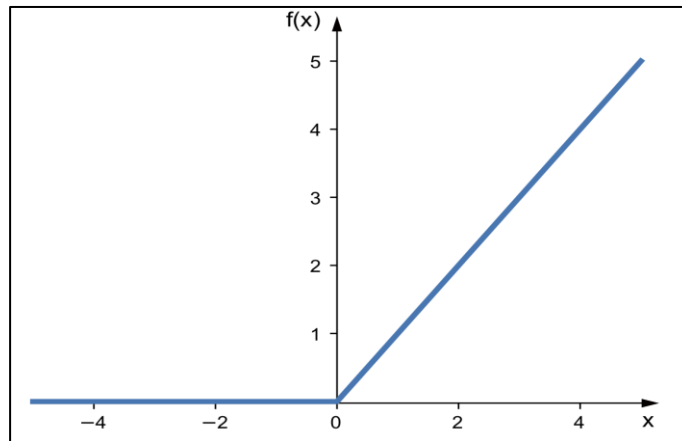
This is a wider logistical activation that can be used to classify multiple categories.

3. The Rectified Linear Unit (ReLU) AF:

This activation function is biologically and mathematically sound. This function has grown in popularity recently and is now the most commonly used AF in DNN. This function has the advantage of training neural networks more quickly than others. The AF is similar to Sigmoid and Tanh. The ReLU computes the activation by setting the input threshold to zero. In other words, if the input is less than zero, a rectified linear unit has output; otherwise, it has raw output. It is given mathematically in (4.3).

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ X & \text{if } x > 0 \end{cases} \quad \text{or } f(x) = \max(0,x) \quad (4.3)$$

It is an activation function called ReLU as shown in Figure 4.15, and its value is zero when  $x = 0$  and linear with value 1 when  $X > 0$ .

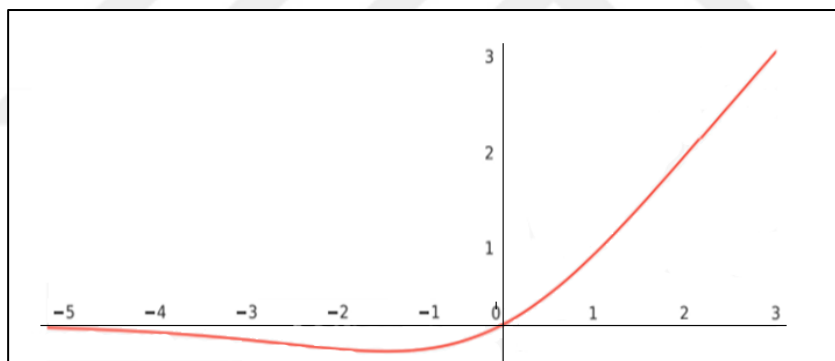


(4.4)

**Figure 4. 15 Rectified Linear Unit (ReLU) activation function  
(Researcher's work)**

4. The AF Self-Gated (SWISH):

In this subjective gate that performs the activation function in neural networks, which depends on the amount of  $\sigma(v)$  with input  $v$  to extract the network output and this non-linear function is called "SWIS," as shown in the equation (4.4).



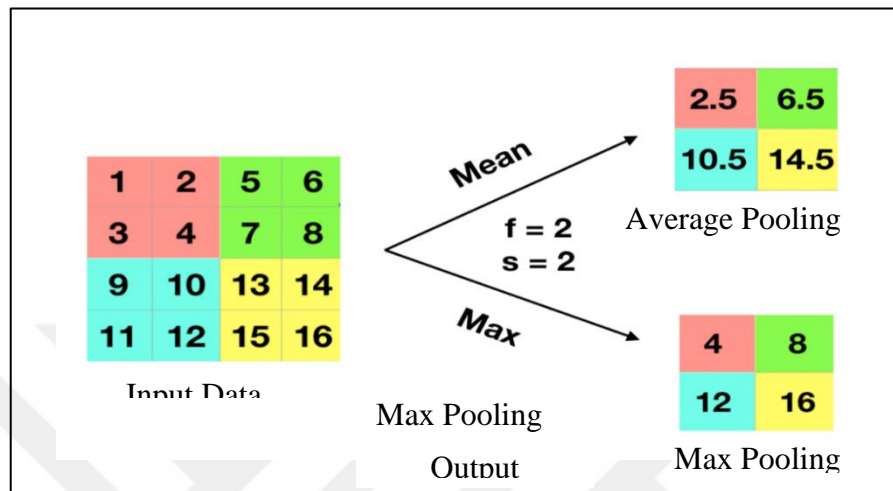
**Figure 4. 16 The Swish activation function (M. A. Wani2020:25)**

Where  $\sigma(v)$  is the sigmoid of activation function with value  $v$  is given in the question (4.4) the SWISH is AF that called non-monotonic function (Djork-Arné Clevert,2015:1511) and is shown in figure 4.16.

**3.6.5. Pooling Layer**

After the completion of the convolution and activation of the input data, to dispose of the values far from the data centre of an elected point means reducing the spatial size of the input. The methods used to extract the higher values of an optional aggregation layer. It entails reducing the number of network parameters, reducing future

calculations, and enhancing the intensity of the object's edges within the image. The assembly layer takes and maintains all the output of the characteristic map of the ability of the convolute layer, i.e. the assembly layer summarizes the area of the neurons in the curvature layer. Assembly techniques vary, but the most common are maximum and medium assembly.



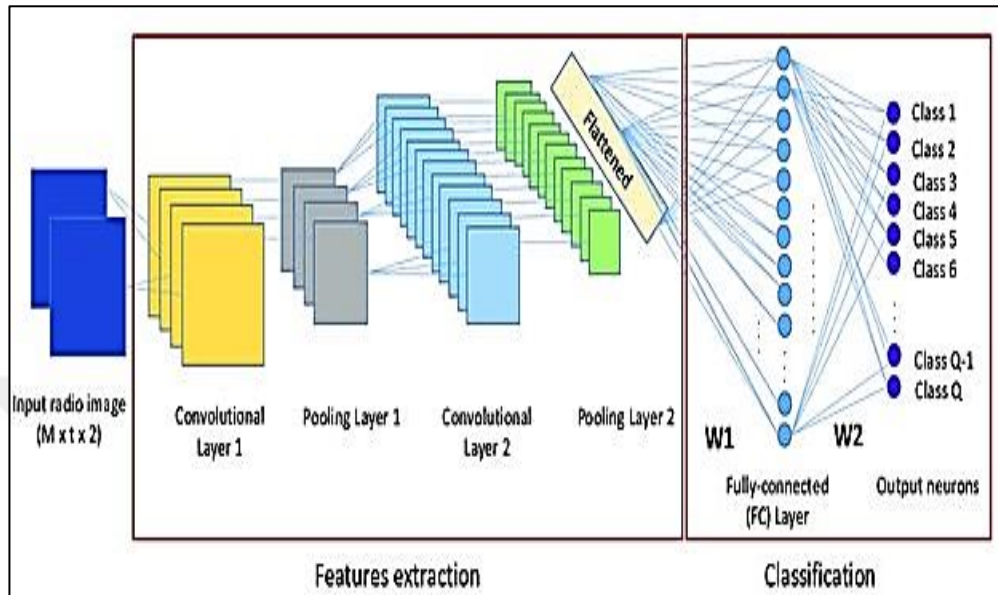
**Figure 4. 17 Max and average –pooling (M. A. Wani2020:26)**

For example, if the input area is 2x2, the maximum summation unit will return the maximum four values of this matrix, as illustrated in Figure 4.17. The alternative formula is average-pooling, which extracts the average for the preceding example's 2x2 matrix. (Ihab S. Mohamed,2017). The major purpose of the max-pooling layer is to remove less important data while maintaining the identified features in a smaller representation since the exact placement of the feature has a strong value. This critical strategy is effective in simple and fundamental situations but has limitations and could be more effective in others.

### 3.6.6. Fully Connected Layer

The investigations revealed that convolutional neural networks have two stages: (1) the stage of feature extraction and (2) the stage of classification. The group of convolution layers and the pooling layer are used as the fundamental stage in CNN to extract features from input during the classification stage, The second stage: its structure consists of multiple or single layers and is fully connected, followed by the "Softmax function layer". The following stage is to base a decision on the investigated features. At this point, we are attempting to solve the categorization problem using the features that have been identified. Calculate the likelihood that certain traits exist in

the spatial space where each category is represented. We will add multiple levels or one completely connected layer to the final network. Each neuron from the preceding layer is connected to the layer of convolution, pooling, and fully connected, which is connected to each neuron in the following layer via this layer.



**Figure 4. 18 The link between the conv layer and the fully linked layer (W. Njjima , 2019:8-26)**

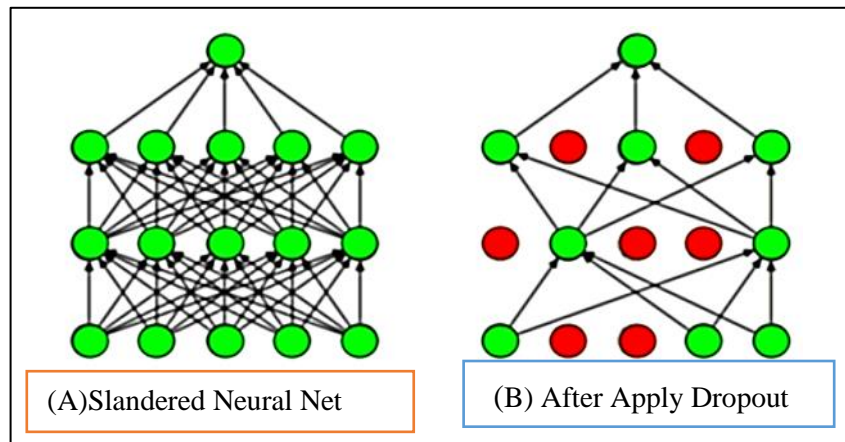
Each value helps to predict how well the value matches a specific class. Figure 4.18 depicts the connection between convolution and fully linked layers. Convolutional and completely linked layers are related because they can be used to learn increasingly complex collections of information.

The final FCL output is passed into the classifier. It yields the output values. The two primary classifiers in CNN are Softmax "Support Vector Machines" and Softmax. The first categorizes the class, whereas the second rank it. It generates the degree of probability for each class plus the probability of one aggregate for each category, and the category with the highest score is considered a target category.

#### **4.6.7. The dropout method in network**

A deep neural network architecture consists of a set of multiple hidden layers that can learn the more complex features. It is followed by a fully connected layer utilized for taking the appropriate decision. The completely connected layer is connected to all features, and these layers are susceptible to over-fitting. Over-fitting is a problem that

occurs when a model is trained and performs well on training data but needs to improve on new data.



**Figure 4. 19 show the dropping in network (M. A. Wani2020:28)**

To solve the over allocation problem, we must use the leak layer in the approved model, which randomly removes some neurons and their connections from the network during the training process shown in Figure 4.19. Where it has been done, The network is reduced to 23 knots, as are the incoming and outgoing edges of the descending node. At the time, the small network was only rehearsing data. After making the reduction, the deleted nodes are re-added to the grid with the weights they were with before the reduction. Leakage significantly reduces excessive composition and improves the possibility of generalizing the learned model. (A.de,2014:6192-6198)

#### **4.6.8. Loss Layer**

It is important to choose a loss function suitable for a particular task. Loss layer determines learning process by comparing network output to real label (or target) and minimize cost. Overall, the loss is calculate by the forward pass and the network parameters gradient for loss is calculate by backpropagation. For multi-category classification problems, softmax rating with loss is common used. First, it takes multi-class scores as an entrance, and uses the softmax function to normalize Input and get a distribution-like output. Thereafter, the loss is calculate by calculating the distribution of probabilities from the target category and the estimated distribution. The softmax function is define as (T. Tykkälä ,2013).

$$y(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (4.5)$$

Where  $0 \leq y(x)_i \leq 1$   $\sum_{j=1}^n y(x)_j = 1$ ,  $i = 1 \dots n$  &  $n$  is number of classes And the cross –entropy between the Target distribution  $P$  and the estimation distribution  $q$  is give by(4.6): (T. Tykkälä ,2013).

$$H(p, q) = \sum_i p_i \log q_i \quad (4.6)$$

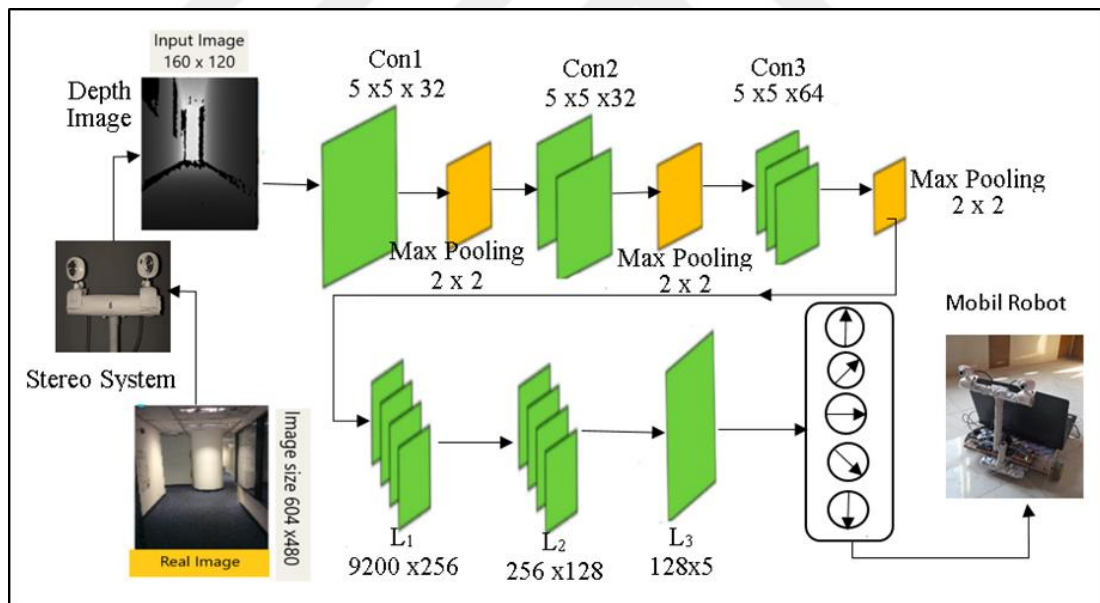
The cost function is the key to tuning neural network weights in our training. To create a better-automated learning model, the neural network operates on training group data during the network's diffusion redirection process, and outputs that, in the case of classification, occur in the full communication layer are obtained to indicate probability or confidence in potential labels. Comparing these alternatives to the target designations and calculating the loss function as a punishment for each variation between the target label and the neural network output. During backpropagation in the network, the approximate derivatives of the loss function are calculated for each trainable weight of the neural net. Weights are modified using these partial derivatives in this process. Under normal conditions, backpropagation is a frequent format that adjusts the trainable weights of a neural network to produce a model with loss, implying that the process is improving the network output, which periodically increases the training accuracy factor and increases the efficiency of network education. Equation (4.7) gives the usual binary entropy loss function: Y. Ho,2019:4806-4813)

$$L_{bce} = -\frac{1}{M} \sum_{i=1}^M [y_i \times \log(h_{\theta}(x_i)) + (1 - y_i) \times \log(1 - h_{\theta}(x_i))] \quad (4.7)$$

Where :  $M$  -is the number of training model,  $Y_i$  – is the target label in training dataset example  $I$ ,  $X_i$  – is the input for training example  $I$ ,  $h_{\theta}$  - is the model with NN weight  $\theta$

#### 4.7. Training Convolution Neural Networks

Supervised deep neural networks are designed to reduce loss function. It means that training is a deep, supervised neural network that completes the search for a set of weights or (parameter values) for the neural net with the reduced value in the loss function Gradient decrease is an error-reduction strategy that calculates the gradients required to update a neural network's weights or parameter values at each step. The most common and successful deep learning algorithm is the gradient based on backpropagation, which is done from the end layer to the beginning layer. Using this learning technique, all neural network weights are initiated randomly using a probability distribution. The inputs fed the network, which produced the output. The collected and needed outputs are used to compute the error, then use a "cost function". This letter used deep learning technology to build a learning network using CNN. As shown in Figure 4.20, the following details the construction of this network, and it works using stacked hierarchical layers to build depending on the supervised learning network.



**Figure 4. 20 The proposed model use CNN (Researcher's work)**

We aim to learn our model of navigation in a previously unknown environment for a mobile robot through learnt it to conclude the control label. This robot's movement can be directed, which includes a 5-control label which involves forward movement, rotating to the right with an angle of 45°, rotation to the left with an angle of 45°,

rotation to the right with an angle of  $90^\circ$ , rotation to the left with an angle of  $90^\circ$ . Where this CNN that we are working through three convolution processors followed by the activation function of the type RELU and then MAX POOLING. After each convolution function, the max pooling and convolution are followed by three linear layers, and the last layer of these three layers contains a full communication layer. The five control markers are classified according to the situation faced by the moving robot as it navigates in the experimental environment, which has been selected with high precision to achieve neutrality in training, as well as its diversity in that it is rich in the various obstacles facing the experimental robot designed for this purpose. Then we evaluate the error during model training through deep techniques. This process is done using the SOFMAX algorithm, which works to figure out the differences between the real target and the target reached via the network. After this process, the gradient algorithm is used, which updates the error in the network's conclusion by reducing the error to as little as possible to achieve high resolution using RMP-like output optimization technology with the back propagations algorithm. The following explains each detail with the mathematical equations we use to extract results from each model training stage and RMP technology. Eventually, we get model training to avoid obstacles facing the moving robot in the real environment. It has been self-controlled by the experience it gains during its training and testing process.

## CHAPTER V

### FINDINGS & DISCUSSION

System Evaluation and Experimental Result In this section, we will review the analyses of the output results that have been reached to determine the extent to which the objectives of our study have been achieved by implementing the model that was designed in our study using deep learning techniques, Python language and libraries specializing in deep learning and comparing it with previous research findings in the same field of research on independent mobile robot navigation.

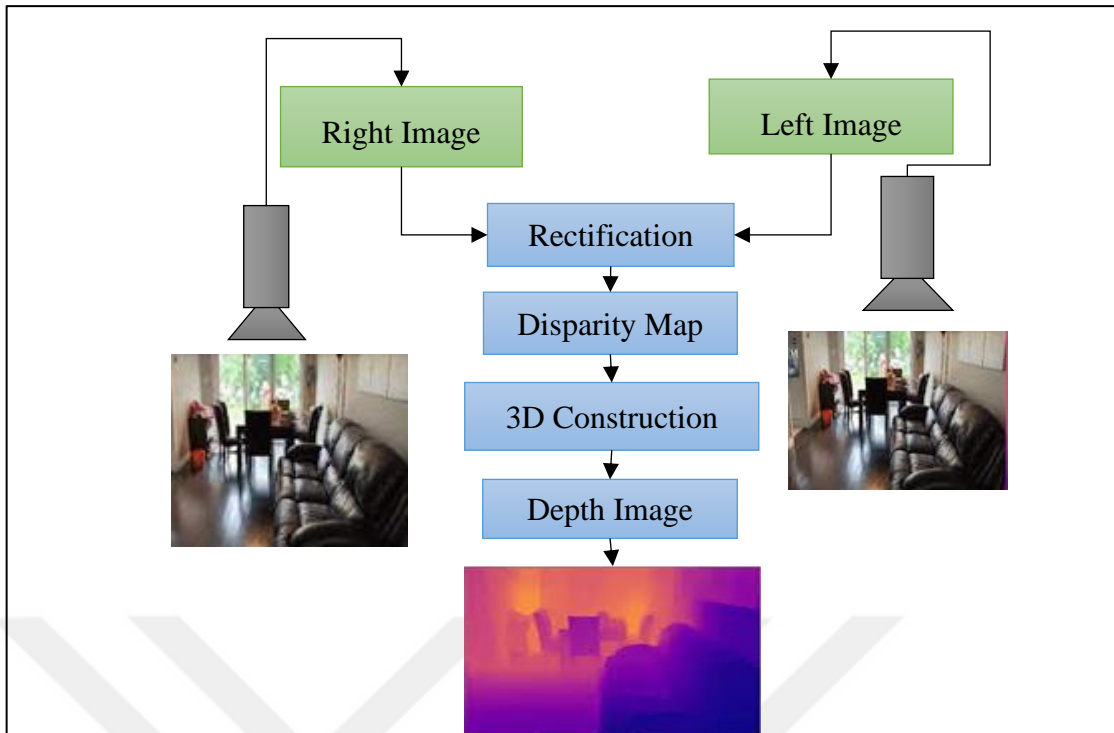
#### 5.1. The Low Cost Sensors to Create Depth Images

To accomplish his duties and achieve his goals, the mechanical structure of the moving robot needs to be controlled. The robot's motion control system is based on three pillars: perception, processing, and action. The cognitive system provides information about the environment, the robot itself, and the relationship between the robot and the environment. It is processed before relevant commands are sent to operators who operate the mechanical structure. Once you know the direction, destination or purpose of the environment and robot, the cognitive structure of the robot must plan the path the robot must go to achieve its goals. As a result, the cognitive level of the robot is a key element used in the decision-making and execution processes that the robot uses to achieve high-level goals. Based on the data collected by the sensors used in our research, the robot's objectives, which are the experience gained by training it in a variety of environments that include different types of obstacles and scenes that enrich the robot's memory, enable it to build a smart map of the paths. That will depend on the environment the robot faces while navigating in a real environment. The cognitive and control system must therefore decide how to act and what to do to achieve its goals.

Furthermore, the control system is responsible for coordinating all input data and planning the robot's motion so it can move accordingly. Computer vision is used by building scenes of depth images that enable deep learning to track objects and images without effort, rather than using map algorithms to create maps of the environment. It ultimately means using motion layout and AI algorithms to determine how the robot interacts. It simulates humans' and organisms' perceptions of the environment through

learning. Therefore we refer to the first goal of the current study, which involves "building low-cost sensors whose function is to create depth images through photo sensors (Web cameras) and stereo systems based on the computer vision system. The results have been good at obtaining depth images using low-cost cameras after building software and using the algorithms we referred to in chapter III of this study, through which more than 6000 images of depth, clarity and high resolution were created through the set of steps,, as follows:

1. The construction of the stereo system simulates the human visual system and the preparation of the hardware necessary for this process. This process has been completed, and the stereo system is being built according to the approved standards.
2. The calibration of reduced-resolution cameras containing aberration and distortion of captured images was done through a set of software and algorithms based on the computer vision system, which is an important decisive stage and was accomplished through the acquisition of the external and internal matrix of the two cameras. This data are stored in their transactions to use in refined images taken by low-cost cameras
3. Building the depth image is part of the data used to train the model through low-cost sensors. This process has been accomplished using a range of approved standard software and algorithms. Figure 5.1 illustrate the steps and outputs that demonstrate the achievement of the first objective of our study.



**Figure 5. 1 shows processes performed to create depth images  
(Researcher's work)**

## **5.2. Prepared by the New Data Collection**

After completing the process of storing the new dataset that we obtained through the robot that navigation in different environments, our use of diverse environments aimed to increase the number of observations that the robot is exposed to during navigation, thus increasing its experience in avoiding various obstacles and increasing the diversity of scenes that positively affect the sample of new training data, as noted by (W. Seaman, 2020:2-26). Show Appendix 12 Figure (A) the different environments in which the data are recorded. A training sample is created where each environment is stored in its folder and then separated from other environments using a laptop based on the mobile robot base, which can capture depth images. These warehouses contain depth images of the sightings facing the robot as it moves, and control labels are identified, namely, the behaviour of the robot when it moves, which is included in the file name in which it stores its deep images, where the file name for each image consists of three parts. The first part is the sequence of images at their storage location. The second part is the mark of control. The last part is the image quality, where each folder contains estimated 1500-depth images in one environment, as in Figure 5.2. The

second part, separated by (the sklearn. model\_selection function using Python), represents the signs of control because of which the image was taken, and then the third part represents the type of images stored. After this task, we have four sites where images of depth and signs of control are stored. Through a Python program, each site is grouped into a single file with a (pth) extension, so we have four files of different environments, approximately 6,000 images of depth with control markers combined with each file name.

Name	Type	Size
img-1-0	PNG File	92 KB
img-2-0	PNG File	109 KB
img-3-0	PNG File	88 KB
img-4-0	PNG File	61 KB
img-5-0	PNG File	112 KB
img-6-2	PNG File	110 KB
img-7-4	PNG File	110 KB
img-8-0	PNG File	112 KB
img-9-0	PNG File	73 KB
img-10-0	PNG File	114 KB
img-11-4	PNG File	123 KB
img-12-2	PNG File	123 KB
img-13-0	PNG File	96 KB
img-14-4	PNG File	109 KB
img-15-0	PNG File	88 KB
img-16-4	PNG File	107 KB
img-17-2	PNG File	90 KB
img-18-0	PNG File	110 KB
img-19-4	PNG File	94 KB
img-20-0	PNG File	112 KB
img-21-0	PNG File	105 KB
img-22-0	PNG File	109 KB
	PNG File	109 KB

**Figure 5.2 the new dataset of storage of depth images and the control label (Researcher's work)**

After this step, these files are grouped into one type of file. We divide the data into two parts with a ratio of 30 to 70% of the images randomly to ensure neutrality in selecting the training sample that we will process into CNN. After this process, the first and second parts of the data that we got before are separated into two parts, the first part is the depth image, and the second part is the label of control, so we have four (pth) variety files for the training system that uses CNN, which works through his office Pytorch in Table 5.1

**Table 5. 1 Represent How Dataset is Separate to Parts with Work it**

(Researcher's work)

File name	Work file
train_data8.pth	Depth images use in train CNN
test_data8.pth	Depth image use in test CNN
train_label8	The control labels use in train CNN
test_label8	The control labels use in test CNN

Table 5.2 shows the folder names that use to save dataset, size, type and number of files within this folder. Preceding its process of storing images through the mobile robot where filtering of the depth image taken from the stereo system done using three consecutive filters.

**Table 5. 2 Represents Depth Images Configured From Five Environments by HCMR (Researcher's work)**

Folder name	No. file	size	File type
Dataset 1	1,509 Files	148 MB	PNG 480x600 pixel
Dataset 2	1,787 Files	129 MB	PNG 480x600 pixel
Dataset 3	1,411 Files	114 MB	PNG 480x600 pixel
Dataset 4	2,557 Files	233 MB	PNG 480x600 pixel
Dataset 5	1,299 Files	101 MB	PNG 480x600 pixel

Normalization filter: Image normalization is modifying the intensity values of pixels in a given image to make the image more appealing to the senses. Image normalization is used to increase the contrast of the image, which helps in better extraction of features from the image or segmentation of the image, as well as to remove the noise content from the image. We use the normalize function in OpenCV to perform image normalization.

Histogram Equalization: an image processing technique that uses the Histogram to adjust the contrast of an image. It spreads out the most common pixel intensity values or stretches the image's intensity range to improve its contrast. Histogram equalization allows portions of the picture with lower contrast to attain a greater contrast. The Histogram makes the images that appear washed out due to a lack of contrast, and equalization might be applied. The bright and dark portions of such images merge, resulting in a flatter image with no highlights or shadows.

Filtering convolution 2D: Image filtering is the process of changing the pixel value of a particular image in order to blur, sharpen, emboss, or make edges more visible. It changes the original image's look. Many image filters are now prevalent in social networking platforms. Most Integral users are aware of the picture filters available. Image filtering is application-specific; since we sometimes need to blur an image and other times, we want it to be crisp. The effects on our photograph are simple to use and see.

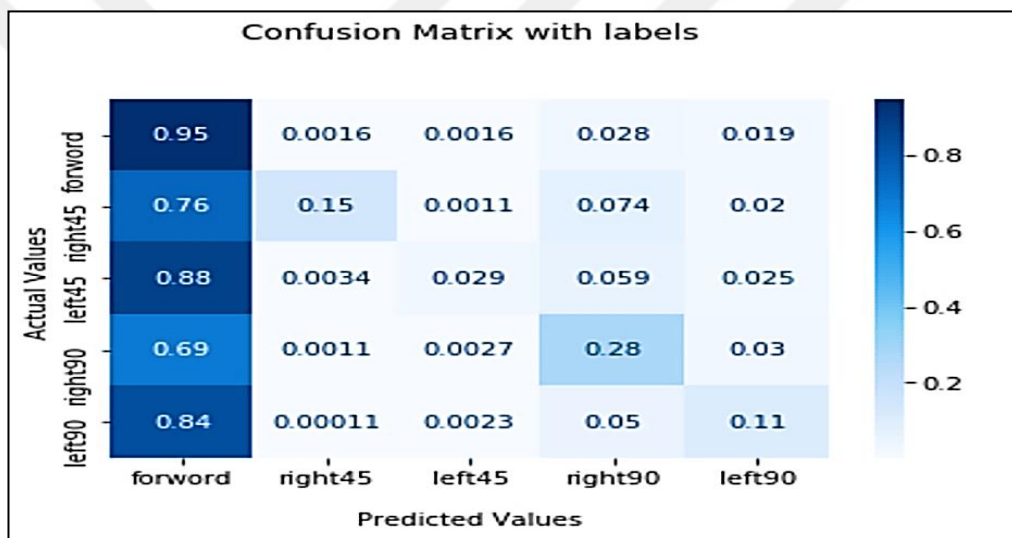
Learning the math and image processing behind it is intriguing. Convolution kernels are used in image filters to perform various image filters. Convolution filters, sometimes known as kernels used to blur, sharpen, emboss, and edge-sharpen pictures. After this process, the new data collection is ready to train our model using CNN. Our results, reflected in this chapter, show the strength and efficiency of the data set that was a construct compared to previous studies' data sets. It demonstrates the success of the first objective of this study.

### **5.3. Modular Training Exercises**

After conducting, the CNN network construction process according to previously explained specifications and the necessary software prepares the procedure for this design by using his office Pytorch to build a deep learning network that uses GPU that works to increase the speed of model teaching and accomplish training processes. Where the model's training is done through three layers of (cov1, cov2, cov3 ) interspersed with the known activation function RULO, then the first layer is followed by the MP1 layer and the two layers are followed by the AP1, AP2. These layers follow three linear layers, the third layer of which is a complete communication layer that classifies control signals and, through the use of the SOFTMAX function, performs the normalizing process of network output, which is combined with the network error exploration function in one formula through our use of the CrossEng function. After extracting the network error, we reduce the error to increase the accuracy of the network classification, which means the optimization processes are made through the gradient algorithm with the rear diffusion algorithm using the RMSprop. This function will reduce the error in the network by performing differentiations on the error function, which is a continuous function, then adjusting the weights in the network and preparing it to spread back to the layer. From Figure 5.3, we can see that the overall

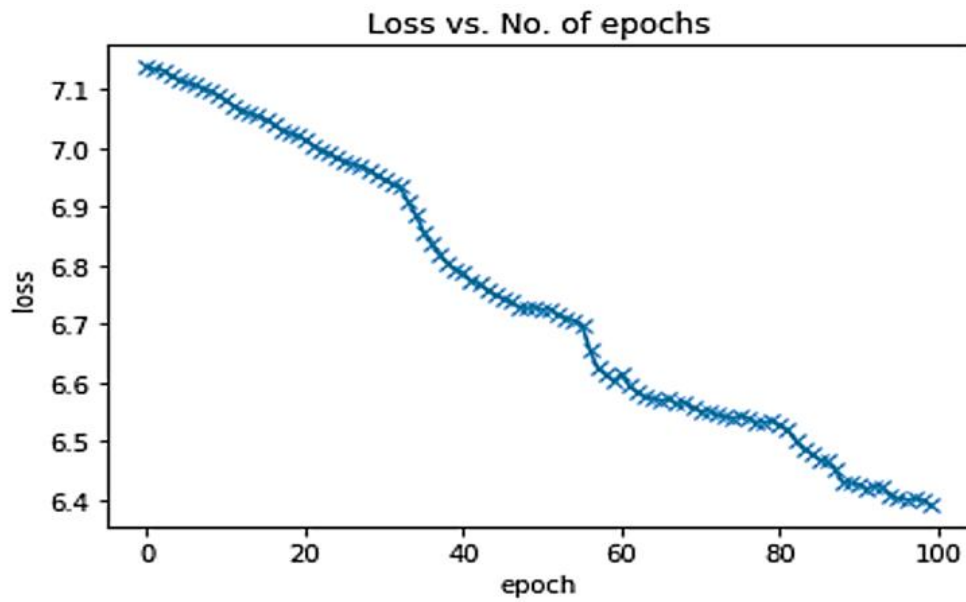
accuracy of the test kit is 80%. Row resolution is 77.05%, the average accuracy of each class.

furthermore, concerning misclassification, which means the system chooses the control mark of any wrong direction or angle of motion where there is very little chance of our system, we are working on generating or choosing the opposite decision (control mark), for example, to misnomer the word "left" as "right." A good chunk of the error rankings might miss the label "half-left round" to "spinner left" or "go straight" and so on. Through this analysis with the example presented, we can demonstrate the effectiveness of training for our model through the excellent reliability that the model has in terms of the distribution of errors.

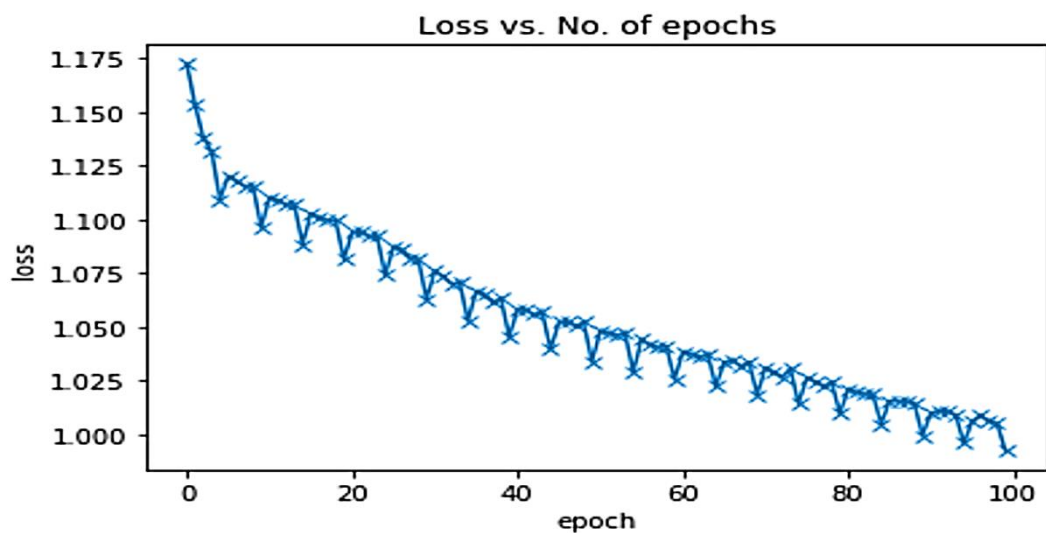


**Figure 5. 3 Confusion matrix on the predicative values, the diameter of the matrix indicates the accuracy of inference**

We can show this in Figures (5.4 and 5.5). This soft decline is due to the data set, the size of more than 6,000 depth images, and the biodiversity used to train our model in this study. In addition, we got 77% accuracy by training the model in 100 batches, which is indicative of the success of our second goal of this study.

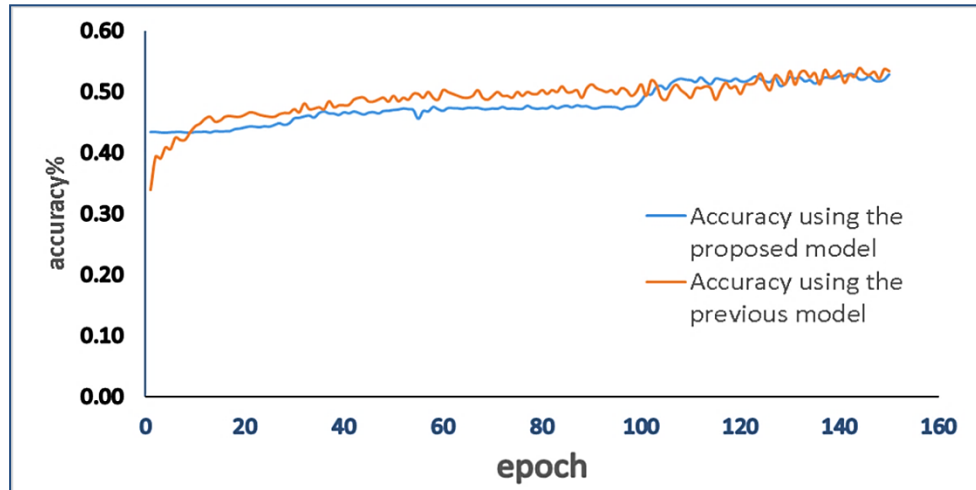


**Figure 5. 4 shows the relationship between epoch and loss using the new dataset**



**Figure 5. 5 The relationship between epoch and loss using the dataset on previous studies**

In addition, our results from previous studies using popular depth sensors (Microsoft kinetic camera) were compared. With our results using depth sensors built from low-cost cameras (web cameras), we got a match in results. That means we got results at a low cost. Table 4.2 shows the difference between the cameras they used and other cameras in terms of size, weight, and energy consumption. It means that through our study, we reduced the cost, size, weight and energy. This result shows in Figure 5.6.



**Figure 5. 6 The compared between training in new dataset and dataset use Microsoft Kinect sensor.**

#### **5.4. Comparison of Human Behavior and Acquired Behavior through CNN**

To demonstrate the success of our approach aimed at enabling the mobile robot to simulate human behaviour in motion and to explore and avoid obstacles. We demonstrated this behaviour by making a comparison of the human behaviour of thinking or human behaviour while directing the robot through the process of building the dataset that connects the scenes facing the robot with manual control using the joystick associated with the main console represented by the control markers that are stored with the depth image. We have proven that human behaviour matches robot behaviour by comparing the manual control sign with the control marker produced by the Deep Learning Network and Limited Sample (CNN output), where the decisions we make are compared by humans (The person responsible for controlling the movement of the robot during the real environment training data collection process), between the decisions the robot makes during model training processes. Five hundred control data points were produced from human and robot control collected during the model training process. Then we calculate the absolute average difference between the two cases. In the first case, the difference in absolute mean (MAD) is 0,1210 rad/s, as shown in Figure 5,7a, and in the second case, the value of the absolute difference average is 0,1400 rad/s shown in Figure 5,7b. Although the robot's path using the dataset created with mono cameras is more consistent with the human walk, the statistic shows our new approach is efficiently working. The new suggested sensor appears more sensitive than the Microsoft Kinect sensor used in the previous study, which accounts for the timing difference. It is owing to the high processors used by

the Kinect cameras to construct the RGB-D image and other issues, including battery drain and excessive heat generated by the sensor's structure. It was unfound in our modest cameras.

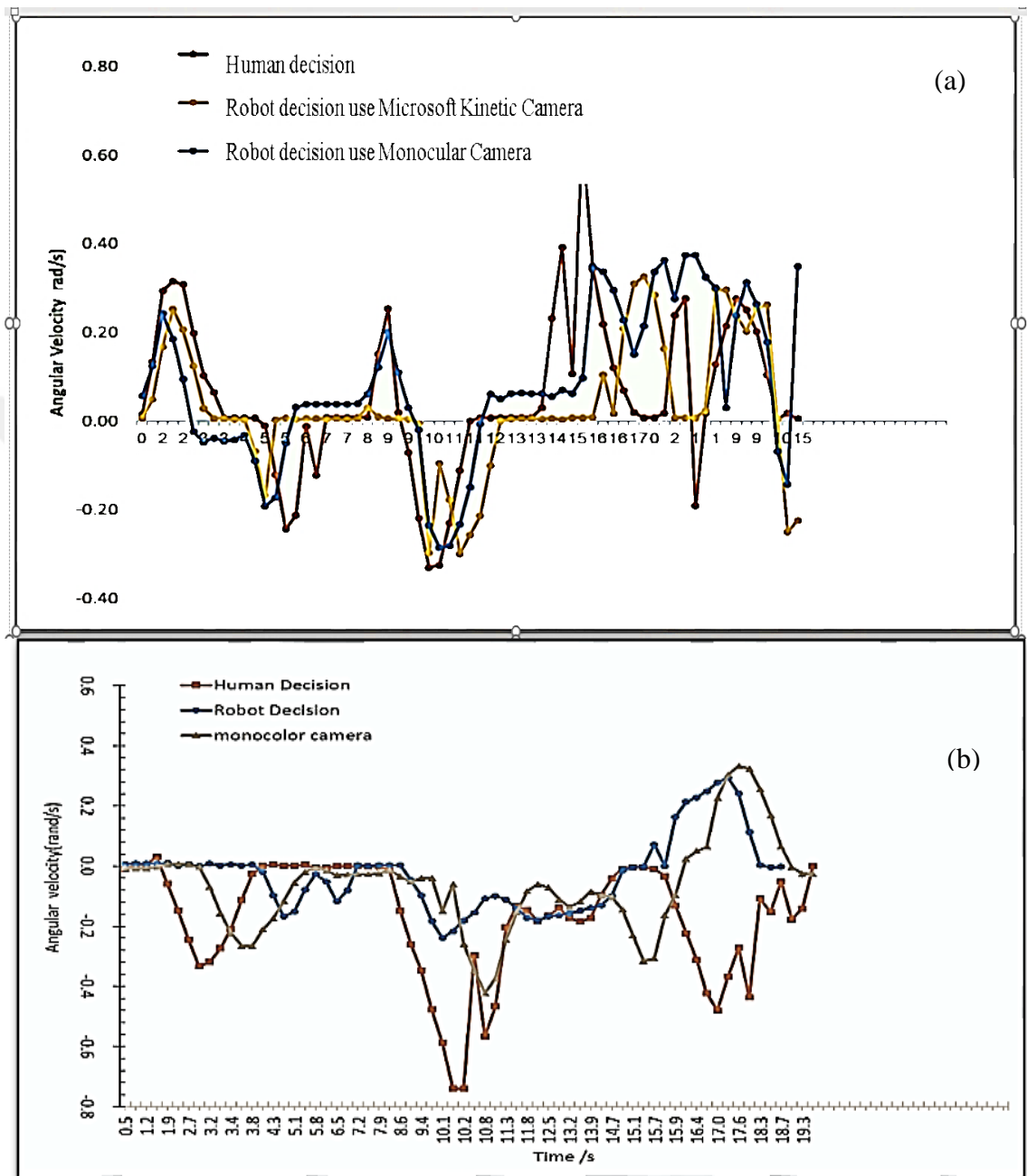


Figure 5. 7 (a) Decisions made by humans and robots compared. A total of 500 points from the human and robot judgment curves have been gather. The average absolute distinction between the two examples is calculated. A median disparity used in the first case is 0.1210 rad/s (a). In the second situation, the value to median disparity 0.1400 rad/s (b).

## CHAPTER VI

### CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

In this section, we will review the findings of our study and the proposed future work that could be added to develop our model for use in broad applications in the service of man and society in different fields. Through previous chapters of our study, we were able to build new low-cost sensors that would produce depth images and build a robot HCMR capable of moving in a real environment after supplying it with new sensors. Through this robot, we built a new dataset to train our model through deep learning on the conclusion of control labels through depth images. We got good results after testing and discussing these results, and in this chapter, we discuss the conclusions and future work of this study.

#### 6.1. Conclusion

Through previous chapters of our study, we could build and apply new low-cost sensors that would produce depth images and build a robot HCMR capable of moving in a real environment after supplying it with new sensors. Through this robot, we built a new dataset to train our model through deep learning on the conclusion of control labels through depth images. We got good results after testing and discussing these results. The dissertation's conclusions drawn from this work are summarised in this chapter as follows:

An approach to human-like internal exploration using deep learning networks was introduced and tested in common internal situations. Experiments show that this system can navigate and avoid obstacles. As a result, automated and human decisions are very similar. However, due to specific constraints, offline training technology is unsuitable for robotic applications, especially when the robot's operating environment is an outdoor environment that needs GPS. The separate classification may need to be more accurate to tighten the continuous space of the condition.

Nevertheless, in a situation where the robot is operating in an indoor environment where we need to activate its Internet system, there are no preconceived maps of the robot. This proposed model will reduce the cost of designing mobile robots using low-cost, low-resolution, low-power cameras. This modest work provides a mobile robot capable of navigating independently in an indoor environment without maps or

information about it where GPS information is unavailable and is done without the internet. It allows mobile robots to navigate independently in dangerous areas where humans are not free and flexible, which proved the important requirement for such robots in the health impacts the world was exposed to in 2019. When the Comprehensive Corona Epidemic (Coved 19) occurred, this project aimed to teach mobile robots navigation by combining computer vision with a deep learning mechanism and developing low-cost, small-scale cameras that used to become sensors installed on the body of mobile robots. Additionally, it is working to reduce the power consumed by the engines, which drive these robots in diverse situations using the microcontroller Arduino Uno.

Ultrasonic Sensor HC-SR04 allows for avoiding obstacles using low accounts after its control sign is linked to the control labels.

Then we trained the robot in the new dataset to become the sixth control label and added other control labels related to new sensors that connect to the main base of the HCMR. In the future, we can see improved training and decision-making procedures combining deep learning, enhanced learning and computer vision. In addition, full information about scenes facing the mobile robot during its training led to the developing a model capable of direct learning and improving its behaviour in independent mobility. We are achieving the main goal of fully imitating human behaviour and even responsive insects in avoiding obstacles.

## **6.2. Suggestions for Future Work**

There are still sides, which need further work, and additional development could be made. There are varieties of directions that could be suggested for future work, which are:

1. To employ and deploy mobile robot services to humans in all aspects of life. It requires a major effort to reduce the cost of building these devices as well as to improve their accuracy in carrying out the tasks required of them
2. We need to increase the number of different sensors that can be equipped and installed in the robot body, and that means increasing the signs of control in the dataset and, as a result enabling the robot to increase its knowledge of the environment surrounding the result increasing its field of work.

3. Increasing the training dataset's size and diversity affects the robot's efficiency in avoiding obstacles to the robot's travels in an unknown environment, i.e. increasing the robot experience.
4. Increasing the control markers in the data set enables the mobile robot to move with high accuracy, especially in narrow spaces without sufficient rotation freedom. In addition, the third wheel rotation can be controlled by adding a linear motor that rotates it depending on the amount of angle of rotation required.
5. We suggest engaging the mobile robot in the medical field by adding necessary sensors that enable the robot to make medical decisions and are trained in them to increase its expertise in the medical field.
6. We suggest using mobile robots to teach children with limb disabilities to move and navigate. It is done by changing the wheels to controllable stems and adding the movements these moving limbs provide to the dataset for us to train the robot.

## REFERENCES

- A.De, K. S. Bayer, and D. E. Koditschek. Active sensing for dynamic, non-holonomic, robust visual serving. Conference: 2014 IEEE International Conference on Robotics and Automation (ICRA) .pp. 6192-6198.
- A.Fusiello,et al.(2000).A compact algorithm for rectification of stereo pairs,(2000). Machine Vision and Applications, vol. 12, 2000, pp.16-22.
- A. Hughes , B. Drury. (2013). Electric Motors and Drives Fundamentals Types and Applications. 4th Edition, 2013.
- A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. (2017).Imitation learning A survey of learning methods. ACM Comput. Surv., vol. 50, no. 2, pp.1–35, Jun. 2017, doi: 10.1145/3054912.
- Abbeel, P., Coates, A. & Ng, A.Y. (2010). Autonomous helicopter aerobatics through apprenticeship learning. The International Journal of Robotics Research.
- Agarap, A.F.(2018).Deep learning using rectified linear units (relu). arXiv, arXiv:1803.08375.
- B.J.Lei, E. A. Hendriks, M.J.T. Reinders .(1999).Reviuwing Camera Calibration and Image Registration Techniques. Technical Report for Calibration and Registration, 1999
- C. Banz, S. Hesselbarth, H. Flatt, H. Blume, and P. Pirsch.(2010).Real-time stereo vision system using semi-global matching disparity estimation Architecture and FPGA-implementation. pp. 93-101.
- C.Xia. (2015).Intelligent Mobile Robot Learning in Autonomous . NavigationEcole Centrale de Lille, 2015.
- Carreras, M., Yuh, J., Batlle, J. & Ridao, P. (2005). A behaviorbased scheme using reinforcement learning for autonomous underwater vehicles.Oceanic Engineering, IEEE Journal of , pp.30, 416–427.
- Ceccarelli M and Kececi EF. (Eds).(2015). Designs and prototypes of mobile robots. ASME Press Robotics Engineering Book Series. Momentum Press Engineering 2015. ISBN: 9780791860472.
- Clement Farabet, Camille Couprie, Laurent Najman, and Yann LeCun.(2013) .Learning hierarchical features for scene labeling. Pattern Analysis and Machine Intelligence. IEEE Transactions on, 35(8):pp.1915–1929, 2013
- Cobo, L.C., Subramanian, K., Isbell, C.L., Lanterman, A.D. & Thomaz, A.L. (2014). Abstraction from demonstration for efficient reinforcement

- learning in high-dimensional domains. *Artificial Intelligence*, pp.216, 103–128.
- D. H. Hubel and T. N. Wiesel.(1962).Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex, *The Journal of Physiology*, vol. 160, pp. 106–154, 1962.View at: [Publisher Site](#) | [Google Scholar](#)
- Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. (2015) .Fast and accurate deepnetwork learning by exponential linear units (elus). In: *arXiv preprint arXiv: pp.1511.07289*.
- Dodgson, N. A. (2004) .Variation and extrema of human interpupillary distance. In A. J.Woods, J. O. Merritt, S. A. Benton and M. T. Bolas (eds.), *Proceedings of SPIE: Stereoscopic Displays and Virtual Reality Systems XI*, Vol. 5291, pp.36–46.
- Dourado, C.M.D., Jr.da Silva, S.P., da Nobrega, R.V., Barros, A.C., Sangaiah, A.K., Reboucas Filho, pp., de Albuquerque, V.H.C,( 2019).A new approach for mobile robot localization based on an online IoT system. *Future Gener. Comput. Syst.*, pp.100, 859–881.
- F.Rubio, F. Valero, and C. Llopis-Albert.( 2019) .A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, pp. 172-296, 2019.
- Foroughi, F.; Wang, J.; Chen, Z.(2019). Indoor Robot Localization in Hand-Drawn Maps by using Convolutional Neural Networks and Monte Carlo Method. In *Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering,Shenzhen, China, 19–21 July 2019*; pp. 1–7.
- Foroughi, F., Zong, P. (2015).Controlling servo motor angle by exploiting Kinect SDK. *Int. J. Comput. Appl.* 2015,pp. 116, 1–6.
- G. Mester,(2006) *Introduction to Control of Mobile Robots* .book 2006 <https://www.researchgate.net/publication/333386558>
- Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7),pp.1527–1554.
- Guang, X.; Gao, Y.; Leung, H.; Liu, P.; Li, G.( 2018) .An autonomous vehicle navigation system based on inertial and visual Sensors. *Sensors* , 18, pp. 2952–2964.
- H. Kanayama, T. Ueda, H. Ito, and K. Yamamoto.(2020) .Two-mode mapless visual navigation of indoor autonomous mobile robot using deep convolutional neural network. in *Proc. IEEE/SICE Int. Symp.Syst.*

- Integr. (SII), Honolulu, HI, USA, Jan. 2020, pp. 536–541, doi:10.1109/SII46433.2020.9025851.
- H. Kraft, J. Frey, T. Moeller, M. Albrecht, M. Grothof, B. Schink, H. Hess. (2004). 3D-camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD. (photonic mixer device)-technologies, in OPTO, 2004, pp. 45-49.
- H.Hirschmuller, and D. Scharstein.(2008).Evaluation of stereo matching costs on images with radiometric differences. IEEE transactions on pattern analysis and machine intelligence. vol. 31, no. 9, pp. 1582-1599, 2008
- H. Su , B. He .(2011).Stereo rectification of calibrated image pairs based on geometric transformation International Journal of Modern Education and Computer Science (IJMECS), vol. 3, pp. 17-24.
- Haoyao.C., Dong Sun.(2012) .Moving groups of microparticles into array with a robot–tweezers manipulation system. IEEE Transactions on Robotics, 28(5):pp.1069–1080, 2012.
- Heikkila, J., Silven, O. (1997). A four-step camera calibration procedure with implicit image correction. Proc. IEEE Computer Society Conf, 1997 pp.1106 –1112.
- Hel, J. Wolff, K., Laine, L.(2018). Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning.arXiv2018, arXiv:1803.10056
- Hold, S. Nunn, C. Kummert, A. Muller-Schneiders, and S. Fac.( 2009). Efficient and robust extrinsic camera calibration procedure for lane departure warning.IEEE Intelligent Vehicles Symposium, pp.. 382-387, doi: 10.1109/IVS.2009.5164308
- Hongsheng Li, Rui Zhao, and Xiaogang Wang.(2014) .Highly efficient forward and backward propagation of convolutional neural networks for pixelwise classification. CoRR, pp.1412,4526, 2014
- Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. Mobilenets .(2017). Efficient convolutional neural networks for mobile vision applications. arXiv 2017, arXiv:1704.04861,pp.2-7.
- Intel Corporation (2001), “Open Source Computer Vision Library.” Reference Manual, <http://www.intel.com/research/mrl/research/opencv/>
- J. Shao, K. Kang, C. C. Loy, and X. Wang.(2015). Deeply learned attributes for crowded scene understanding. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4657–4666, June 2015.

- J. Wang, Q. Zhang, D. Zhao, and Y. Chen.(2019) .Lane change decision making through deep reinforcement learning with rule-based constraints. 2019 IEEE.International Joint Conference on Neural Networks (IJCNN).pp.1-6.
- Jaradat, M.A.K., Al-Rousan, M. & Quadan, L. (2011). Reinforcement based mobile robot navigation in dynamic environment. *Robotics and Computer-Integrated Manufacturing*, 27, pp.135–149.
- K.Sadekar.(2020). Understanding Lens Distortion.  
<https://learnopencv.com/understanding-lens-distortion>
- K.Fukushima.Neocognitron.(1982). a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.View at: Publisher Site | Google Scholar
- K. Zhu , T. Zhang. (2021). Deep reinforcement learning based mobile robot navigation: A review. *Tsinghua Sci. Technol.*, vol. 26, no. 5, pp. 674–691,Oct. 2021.
- Kececi.EF , Ceccarelli M. (2015). Mobile robots for dynamic environments. ASME Press Robotics Engineering Book Series. Momentum Press Engineering 2015, pp. 182. ISBN-13: 978-0791860526.
- Kendall, A.; Grimes, M.; Cipolla, R. Posenet . (2015). A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
- Kim, H.; Lee, D.; Oh, T.; Choi, H.T.; Myung, H.(2015) .A probabilistic feature map-based localization system using a monocular camera . *Sensors* 2015, 15, pp. 21636–21659.
- Krizhevsky, A., Sutskever, I., Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf.Process. Syst.* 2012, 25, pp.1097–1105. [CrossRef]
- Kunihiko Fukushima ,Neocognitron. .(1980). A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):pp.193–202.
- Lin,L., Shinn, S.W., Gen, M. *et al.*(2006) Network model and effective evolutionary approach for AGV dispatching in manufacturing system. *J Intell Manuf* pp:17, 465–477 (2006). <https://doi.org/10.1007/s10845-005-0019-4>

- Lin,S.; Wang, J.; Xu, M.; Zhao, H.; Chen, Z. (2021)Topology Aware Object-Level Semantic Mapping Towards More Robust Loop Closure. *IEEE Robot. Autom. Lett.* 2021, 6,pp. 7041–7048. [CrossRef]
- Lingshuai M, Yang L, and Haitao G. (2016).A new type of small underwater robot for small scale ocean observation. In *The 6th Annual IEEE International conference on cyber technology in automation, control and intelligent systems*, Sofitel Chengdu Taihe, Chengdu, China, PP:19–22 June 2016. IEEE
- M.Ghimire, M. R. Choudhury, and G. S. S. H. Lagudu.(2021) .Lane Change Decision-Making through Deep Reinforcement Learning. *arXiv preprint arXiv:2112.14705*, 2021.pp.1-7.
- M.Okutomi ,T. K anade. (1993) .A multiple-baseline stereo. *IEEE Transactions on Pattern Analysys and Machine Intelligence*, Vol. 15, No. 4, pp.353-363.
- M.Shahab, and A. Masoud.(2009).Energy-efficient motion control of mobile robots. *King Fahd University of Petroleum & Minerals, Course Task EE*, vol. 656.pp.4-27, 2009
- Min Lin, Qiang Chen, and Shuicheng Yan.(2013). Network in network. *CoRR* , abs/1312.4400, 2013.
- Moser, J. Hoffman ,R. Hildebrand, and E. Komendera .(2022).An Autonomous Task Assignment Paradigm for Autonomous Robotic In-Space Assembly. *Frontiers in Robotics and AI*, vol. 9, 2022-February-25, 2022.
- Mülling, K., Kober, J., Kroemer, O. & Peters, J. (2013). Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research*, 32,pp. 263–279.
- N.Ayache ,C. Hansen. (1988).Rectification of images for binocular and trinocular stereovision. in *9th International Conference on Pattern Recognition* , vol.1, 1988, pp. 11-16.
- Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. (2014) . Dropout A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1),pp.1929–1958, 2014.
- O. Mohamed, Z. Mohsen, M. Wageeh, and M. Hegazy. (2021).Autonomous navigation in dynamic environments: Deep learning-based approach. *arXiv preprint arXiv:2102.08758*, 2021.
- Onyekpe, U., Palade, V., Kanarachos, S.(2021). Learning to localize automated vehicles in challenging environments using Inertial Navigation Systems. (INS). *Appl. Sci.* 2021, 11, pp.1270–1292.

- P. D. Lin , C. K. Sung.(2007) .Comparing two new camera calibration methods with traditional pinhole calibrations. *Opt. Express* 15(6), pp.3012–3022.
- P. Swapna, N. Krouglicof, and R. Gosine.(2009) .The question of accuracy with geometric camera calibration. in *Proceedings of the Seventh IEEE Canadian Conference on Electrical and Computer Engineering (IEEE, 2009)*, pp. 541–546.
- R. Farkh, K. Al Jaloud, S. Alhuwaimel, M. T. Quasim, and M. Ksouri(2021) .A Deep Learning Approach for the Mobile-Robot Motion Control System. *Intelligent Automation And Soft Computing*, vol. 29, no. 2, pp. 423-435, 2021.
- R.Siegwart, I. R. Nourbakhsh and D. Scaramuzza.( 2011). *Introduction to autonomous mobile robots: MIT press*, 2011,pp:181-196
- R.Szeliski. (2010).*Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.
- R.Weideman . (2019).*Robot Navigation in Cluttered Environments with Deep Reinforcement Learning*.
- Radwan, N., Valada, A. Burgard, W. Vlocnet++. (2018) . Deep multitask learning for semantic visual localization and odometry. *IEEE Robot. Autom. Lett.* 2018, 3,pp. 4407–4414. [CrossRef]
- Ran, T., Yuan, L.; Zhang, J. (2021) . Scene perception based visual navigation of mobile robot in indoor environment. *ISA Trans.* 2021, pp.109, 389–400. [CrossRef] [PubMed]
- R. Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp.1631–1642, Stroudsburg, PA, October 2013. Association for Computational Linguistics
- Riedmiller, M., Gabel, T., Hafner, R. & Lange, S. (2009). Reinforcement learning for robot soccer. *Autonomous Robots*, 27, pp. 55–73.
- S. Liu , D. Sun.(2011).Optimal motion planning of a mobile robot with minimum energy consumption. 2011 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2011, pp. 43-48, doi: 10.1109/AIM.2011.6027010
- S. Nedeveschi, T. Marita, M. Vaida, R. Danescu, D. Frentiu, F. Oniga, C. Pocol, and D. Moga.(2002).Camera Calibration Method for Stereo Measurements. *Journal of Control Engineering and Applied Informatics (CEAI)*, vol. 4, no. 2, pp. 21-28.

- S. Praveen,(2020).efficient depth estimation using sparse stereo-vision with other perception techniques. *Coding Theory*, pp.. 111-127, 2020
- S. P. P. da Silva, J. S. Almeida, E. F. Ohata, J. J. P. C. Rodrigues, V. H. C. de Albuquerque and P. P. Rebouças Filho. (2020).Monocular Vision Aided Depth Map from RGB Images to Estimate of Localization and Support to Navigation of Mobile Robots. in *IEEE Sensors Journal*, vol. 20, no. 20, pp. 12040-12048, 15 Oct.15, 2020, doi: 10.1109/JSEN.2020.2964735.
- Schmidhuber, J.(2015). Deep learning in neural networks: an overview. *Neural Netw.* 61,pp. 85–117 (2015)
- Saravanan, M.; Kumar, P.S.; Sharma, A. (2019).IoT enabled indoor autonomous mobile robot using CNN and Q-learning. In *Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, Bali,Indonesia, 1–3 July 2019; pp. 7–13.
- Sattler, T., Leibe, B, Kobbelt, L.(2016). Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, 39, pp.1744–1756.
- Scaramuzza D, Martinelli A, Siegwart R. (2006).A toolbox for easily calibrating omnidirectional cameras. In: *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on 9 October 2006*; IEEE; pp. 5695-5701
- Shaohua Li, Huimin Huang, Yue Zhang, and Ming Liu.( 2015). An efficient multi-scale convolutional neural network for image classification based on pca. In *2015 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 57–62, June 2015.
- Shigeki Iida , Shin`ichi Yuta. (1990).Control of a Vehicle Subsystem for an Autonomous Mobil Robot with Power Wheled Steerings. *Proceedings of the IEEE International Workshop on Intelligent Motion Control*, vol.2, pp. 859-866, Istanbul, Turkey, 1990.
- Shaohua Li, Huimin Huang, Yue Zhang, and Ming Liu.(2015) .An efficient multi-scale convolutional neural network for image classification based on pca. In *2015 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 57–62, June 2015.
- Siegwart R , Nourbakhsh IR.(2004). *Introduction to autonomous mobile robots*. Massachusetts London, England: A Bradford Book, The MIT Press Cambridge, 2004. ISBN pp.0-262-19502-X
- Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv: pp.1409.1556*

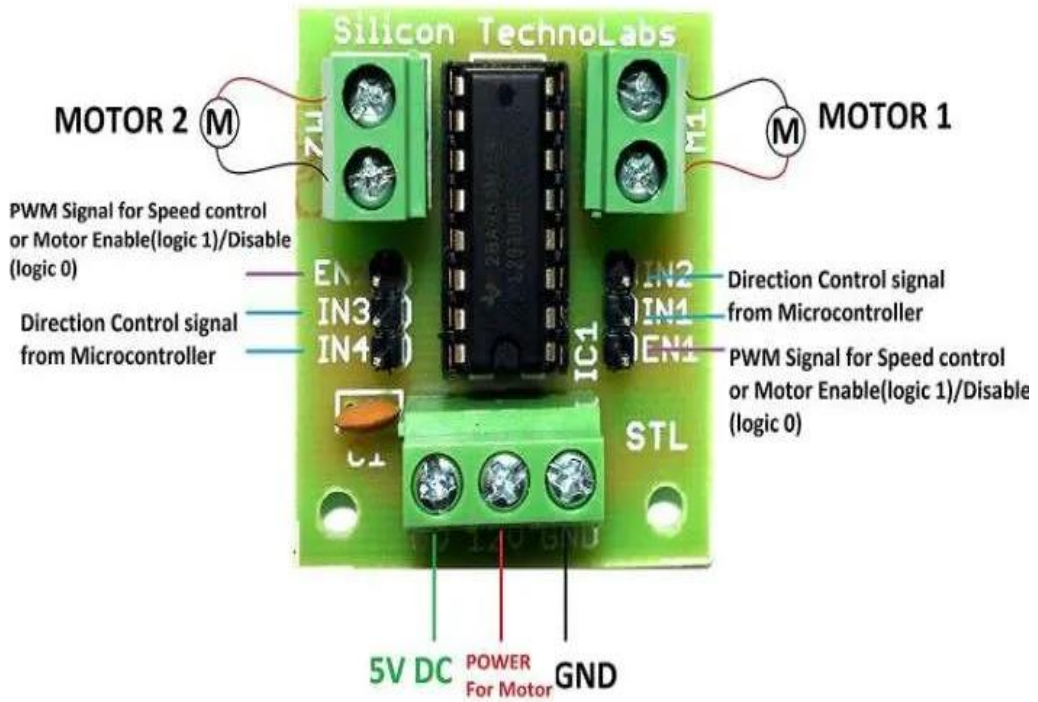
- T. Matsuzaki, O. Elfadil, K. Horiuchi, H. Shiratsuchi, and K. Mashiko.(2019). Embedded controller based learning system for DC motor contro. Int J Innov Comput Inform Control, vol. 15, no. 3, pp. 997-1007, 2019.
- T. Tykkälä. (2013).Real-time image-based RGB-D camera motion tracking and environment mapping,. Université Nice Sophia Antipolis, 2013.
- Toshiba Inc.(2001). Datasheet: TA8440H: DC Motor Full Bridge Driver, 2001.
- Tsai, R. Y.(1987).A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE Journal of Robotics and Automation, RA-3(4)/1987: pp.323-344.
- Tsai .(2003).Camera Calibration .[Internet]. 2003. Available from: [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/DIAS1/](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/DIAS1/) Accessed: pp.05-11-2003
- V.Casser, S. Pirk, R. Mahjourian, and A. Angelova .(2021).Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. pp. 8001-8008.
- W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue.(2019).Deep CNN for indoor localization in IoT-sensor systems. Sensors, vol. 19, no. 14, pp. 3127, 2019.
- V.Mnih, K Kavukcuoglu, David Silver, Alex Graves,Ioannis Antonoglou, Dean Wierstra, and Martin A. Riedmiller. (2013) . Playing atari with deep reinforcement learning. CoRR, abs/1312.5602, 2013.
- W. Sun , J. R. Cooperstock. (2005).Requirements for Camera Calibration: Must Accuracy Come with a High Price. in Proceedings of the Seventh IEEE Workshop on Applications of Computer Vision (IEEE, 2005),pp. 356–361
- Wu, H.Wu, X., Tian, G. (2018) .Indoor robot localization based on single RFID tag. Artif. Life Robot. 2018, 23,pp. 373–379. [CrossRef]
- X. Gao, L. Yan, and C. Gerada.(2021).Modeling and analysis in trajectory tracking control for wheeled mobile robots with wheel skidding and slipping. Disturbance rejection perspective. year 2021" pp. 222.
- Y. Ho,S. Wookey.(2019).The real-world-weight cross-entropy loss function Modeling the costs of mislabeling. IEEE Access, vol. 8, pp. 4806-4813, 2019
- Yann LeCun, Bernhard E. Boser, John S. Denker, Donnie Henderson, R. E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. (1990). Handwritten digit recognition with a back-propagation network. In D. S.

Touretzky, editor, *Advances in Neural Information Processing Systems* 2, pp. 396–404. Morgan-Kaufmann, 1990

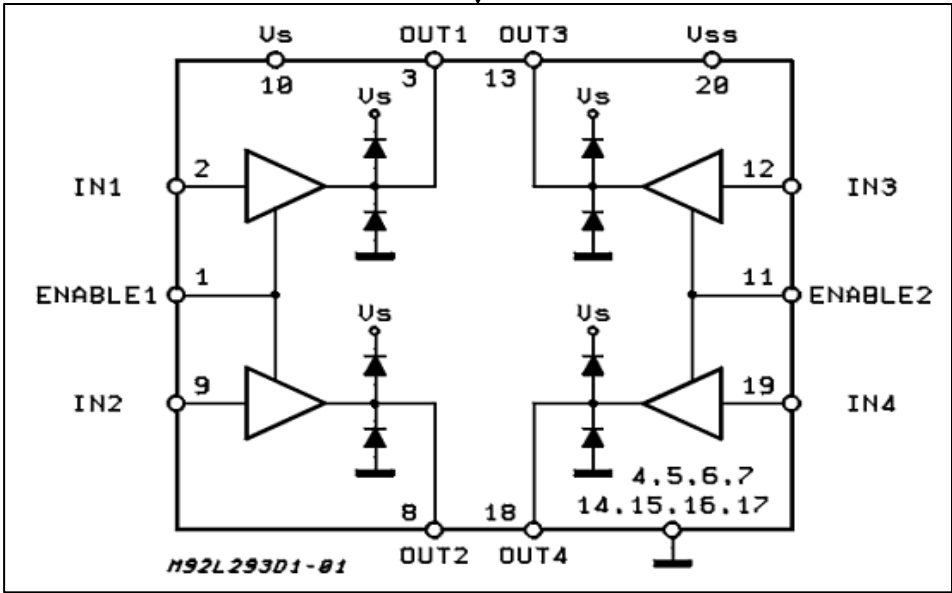
Zhang Z. (1999). Flexible Camera Calibration By Viewing a Plane From nknown Orientations. *International Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999, pp. 666-673.



# APPENDIX 1



L293D IC MOTOR DRIVER



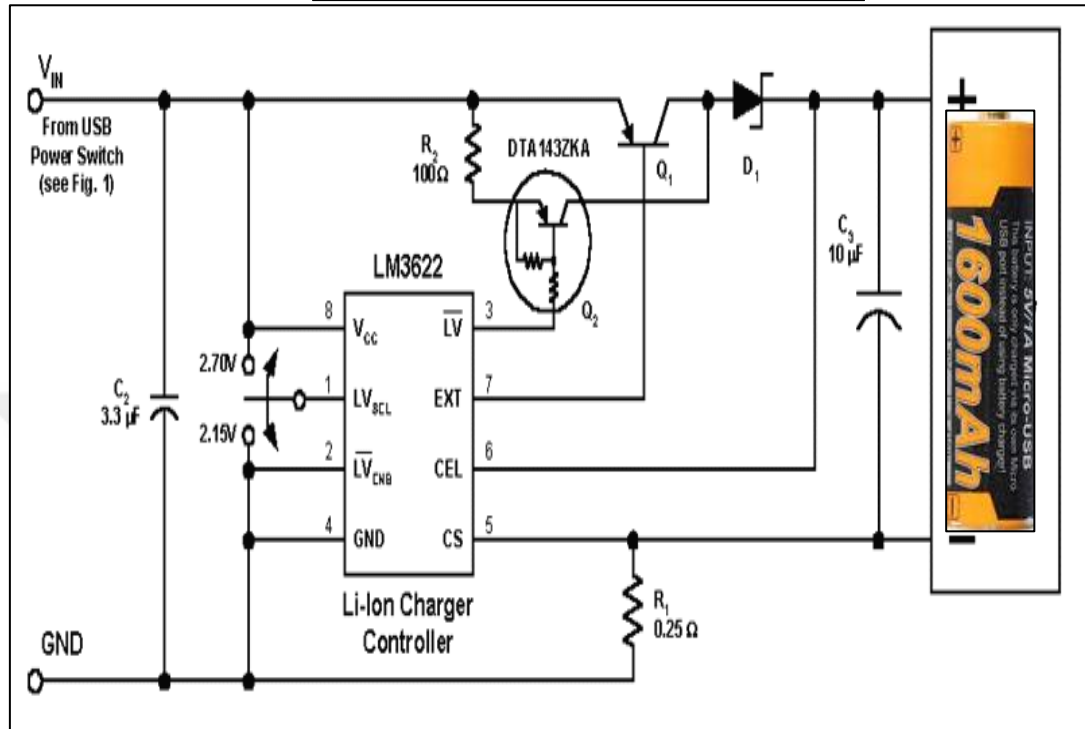
<https://www.datasheets.com>

## APPENDIX 2

Pin No.	Name	Function
a	Enable 1-2	When this pin is given HIGH or Logic 1, the left side of the IC works and when it is low, the left side doesn't work.
2	INP1	When this pin is given HIGH or logic 1, the output 1 becomes HIGH.
3	OUT 1	This pin is connected to one of the terminals of the motor 1.
4,5	GND	Should be connected to the circuit's ground.
6	OUT 2	This pin is connected to one of the terminals of the motor 1.
7	INP 2	When this pin is given HIGH or Logic 1, the output 2 becomes HIGH.
8	VCC 2	This is the voltage required to run the motor. IT can be greater than the IC voltage(VCC1).
16	VCC 1	It provides power to the l293D IC. So, this pin should be supplied with 5 V.
15	INP 4	When this pin is given HIGH or logic 1, the output 4 becomes HIGH.
14	OUT 4	This pin is connected to one of the terminals of the motor 2.
13,12	GND	Should be connected to the circuit's ground.
11	OUT 3	This pin is connected to one of the terminals of the motor 2.
10	INP 3	When this pin is given HIGH or logic 1, the output 3 becomes HIGH.
9	Enable 3-4	When this pin is given HIGH or Logic 1, the right side of the IC works and when it is low, the right side doesn't work.
<a href="https://www.datasheets.com/">https://www.datasheets.com/</a>		

## APPENDIX 3

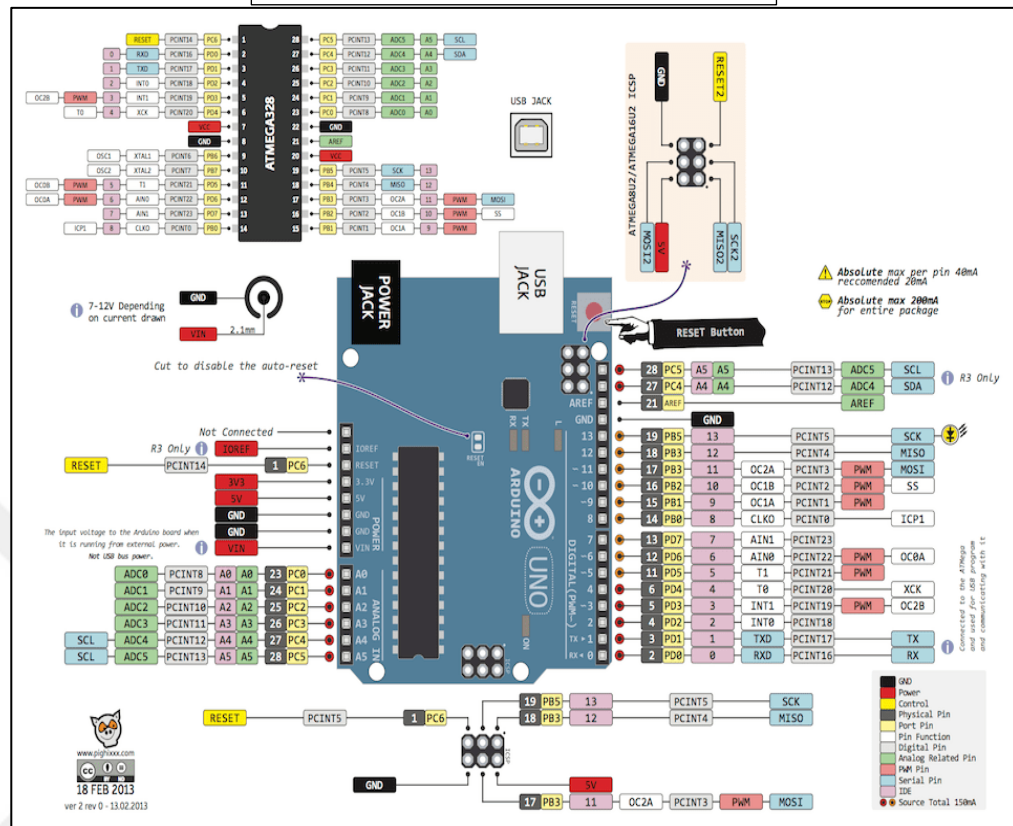
### Li-On Battery Charger via USB



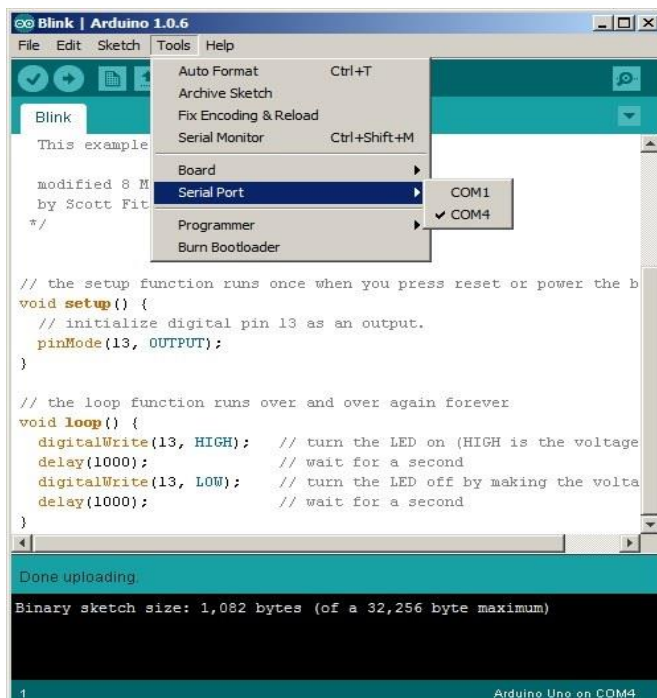
The USB port is a great power source for charging a single-cell li-on battery. It is capable of providing a maximum 5.25V and 500mAh. The circuit above is a USB powered single-cell battery charger. LM 3622 is used as a controller. This special-purpose IC has precise end-of-charge control and low-leakage current of the battery around 200nA. (Researcher's work)

# APPENDIX 4

## Arduino Uno Micro controller circuit



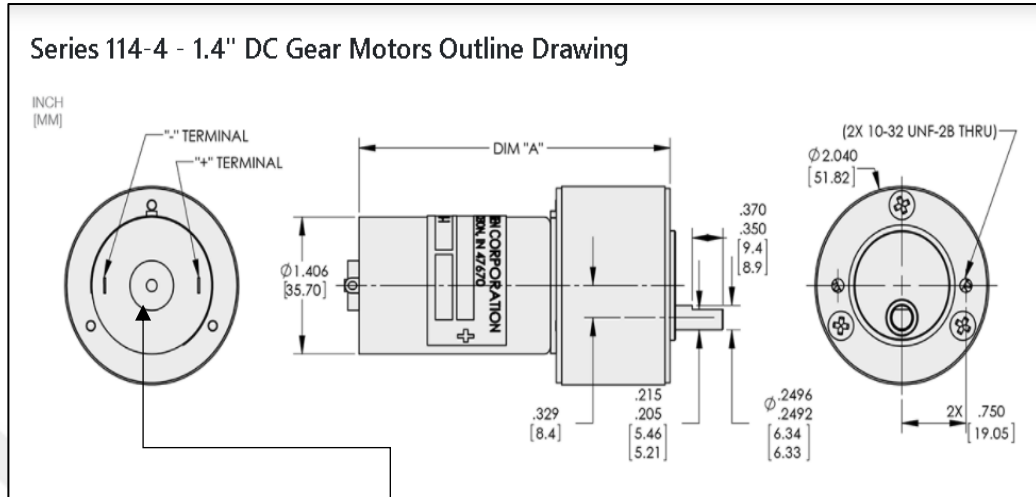
After you did all of these, plug your board, choose the new COM port on the Arduino IDE - Serial port menu, than your Arduino is ready to be programmed. (<https://www.datasheets.com>) and (Researcher's work)



#### Features:

- Atmega328 Microcontroller
- 7-12 V Input Voltage
- 14 Digital I/O pins
- 6 PWM Out
- 6 ADC In

## APPENDIX 5

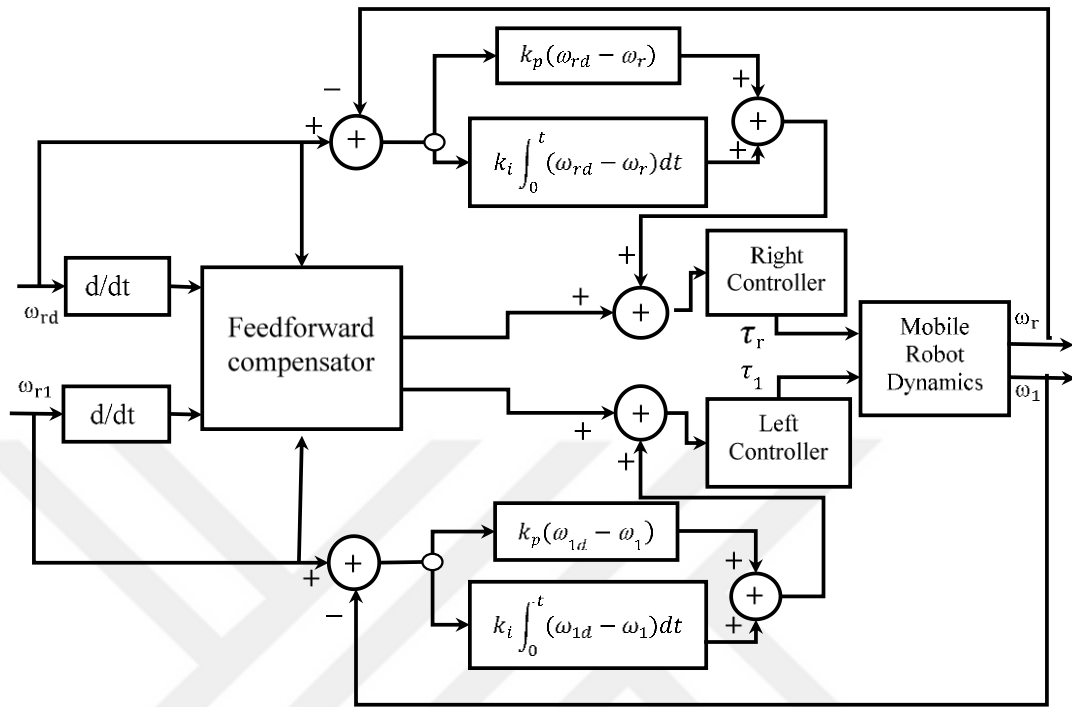


### TYPICAL APPLICATIONS:

- Printers
- Plotters
- Pumps
- Fans
- Cutting Tools
- Medical Devices
- Robotics

<https://www.datasheets.com>. (Researcher's work)

## APPENDIX 6



**Block diagram of the proposed controller (G. Mester,2006)**

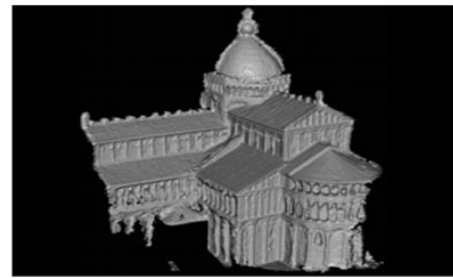
## APPENDIX 7



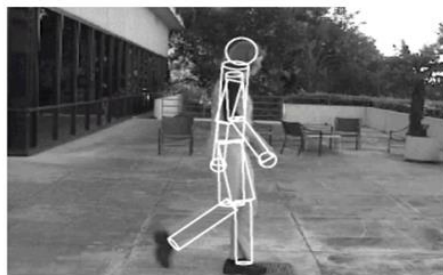
**Figure( A) Human optical system has no problem interpreting images .  
( R.Siegwart, 2011:1)**



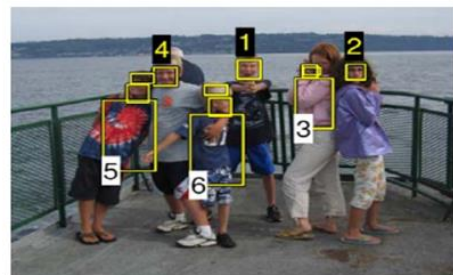
(a)



(b)



(c)



(d)

**Figure (B) Some types of algorithms and applications of computer vision. (1) A structure of motion algorithms can reconstruct a scattered 3D point model of a large complex landscape (2) stereo matching algorithms can construct a detailed 3D model of a building (3), (4) Facial detection algorithms.( R.Siegwart, 2011:2)**

## APPENDIX 8



(1) BRETT is learning how to screw a cap



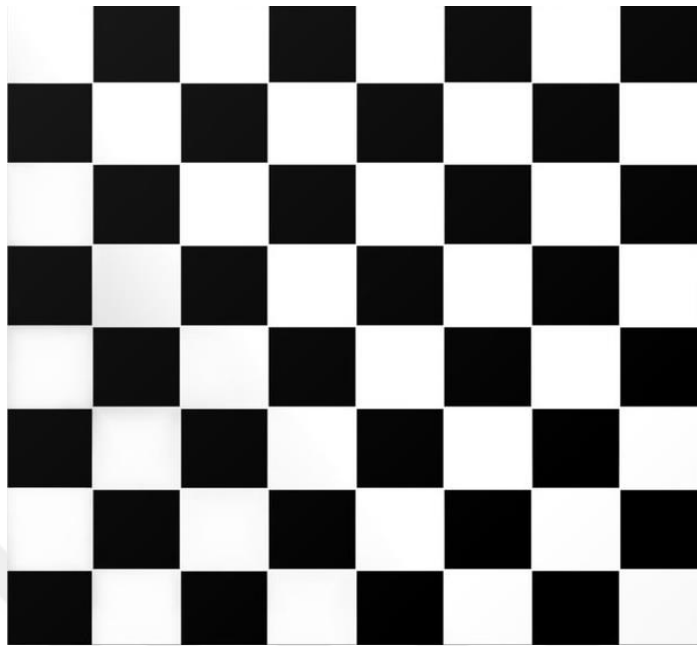
(2) Learning to flip an artificial pancake.

**Figure ( A 1,2) Various applications of robot learning. .( C.Xia, 2015:8)**

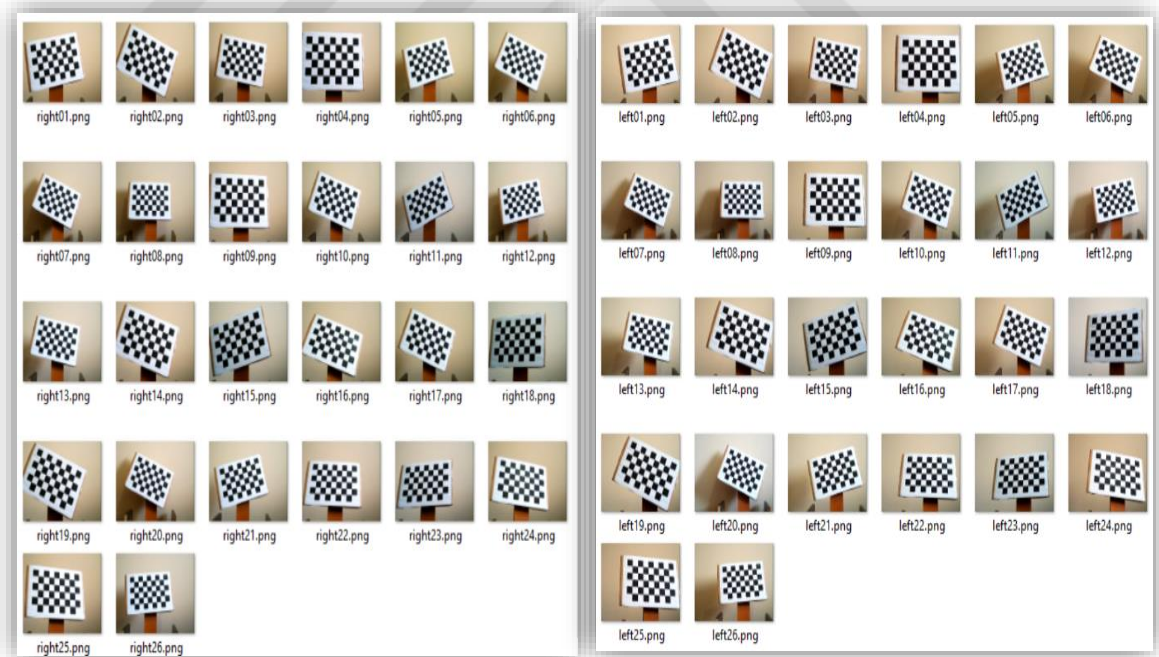


**Figure B. The stereo system hardware contains web camera in same feature  
(Researcher's work)**

## APPENDIX 9



**Figure (A) the checkerboard pattern 8 x8 cell. (Researcher's work)**

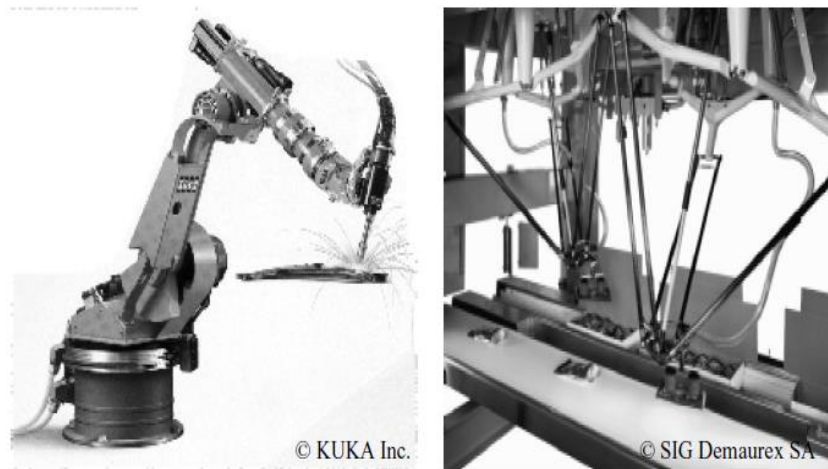


a: The Image from Right camera

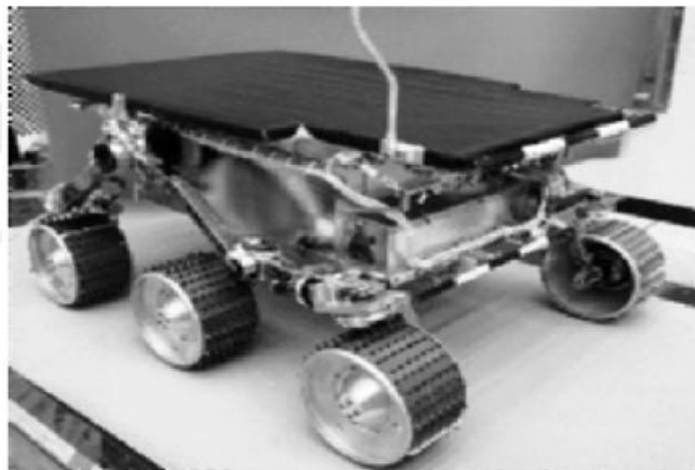
b: The Image from Left camera

**Figure (B) the images of a checkerboard for a-right, b-left images  
(Researcher's work)**

## APPENDIX 10

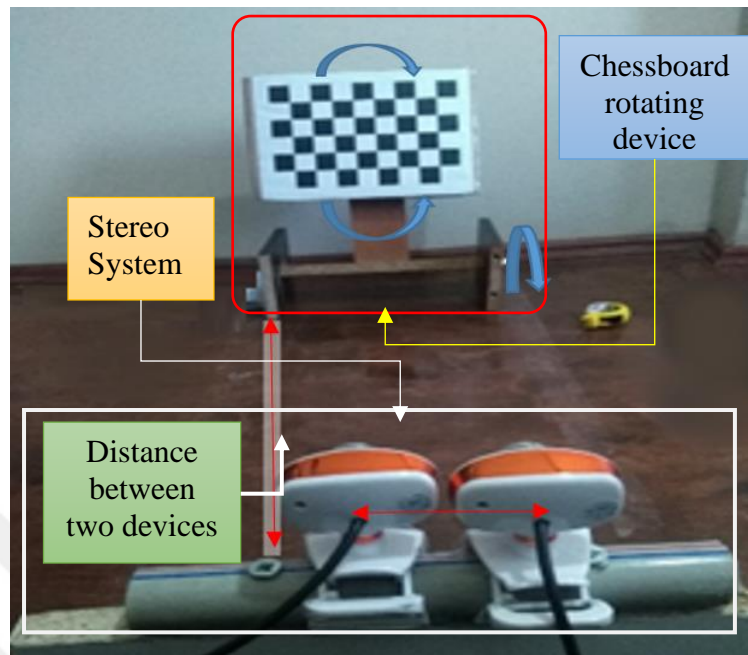


**Figure (A) Picture of auto assembly plant-spot welding robot of KUKA and a parallel robot Delta of SIG Demarex.( R.Siegwart, 2011)**

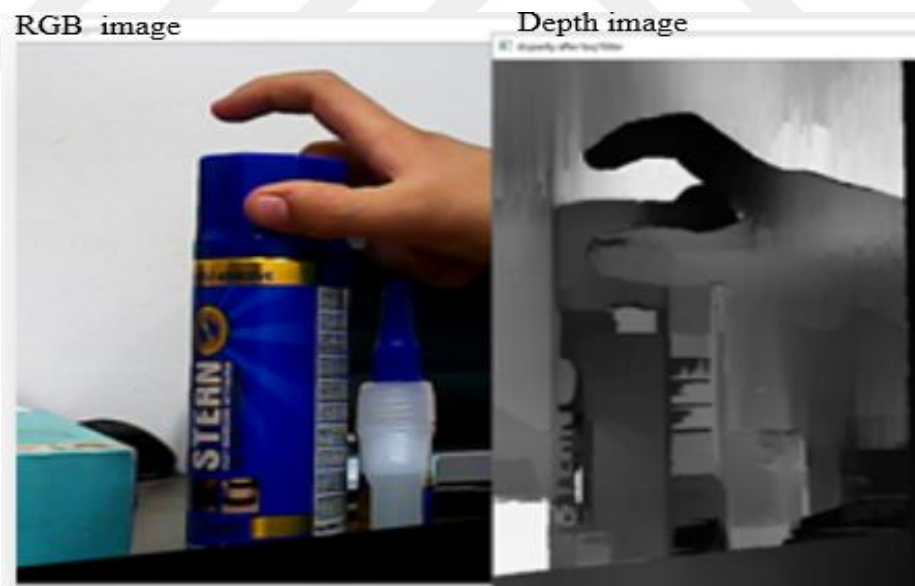


**Figure(B) The mobile robot Sojourner was use during the Pathfinder mission to explore Mars in summer 1997. .( R.Siegwart, 2011)**

## APPENDIX 11



**Figure (A) the system uses to take a picture of the chessboard with stereo system. (Researcher's work)**



**Figure (A) The RGB image and the depth image after calibration (Researcher's work).**

## APPENDIX 12



Figure (A) Diverse Environments (Researcher's work)

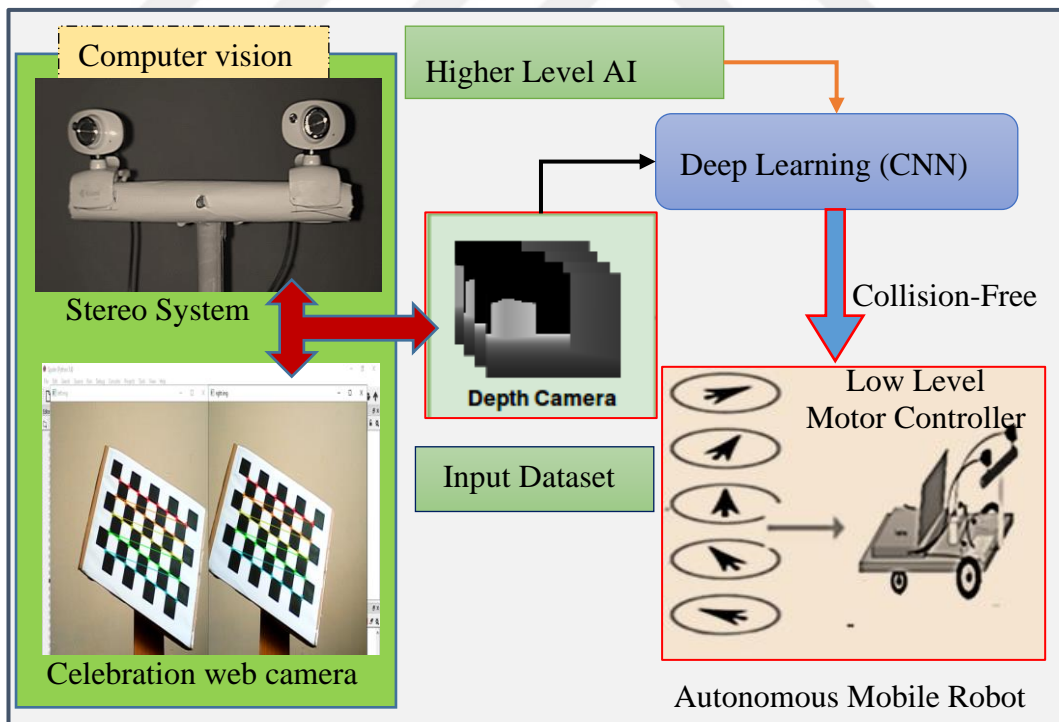
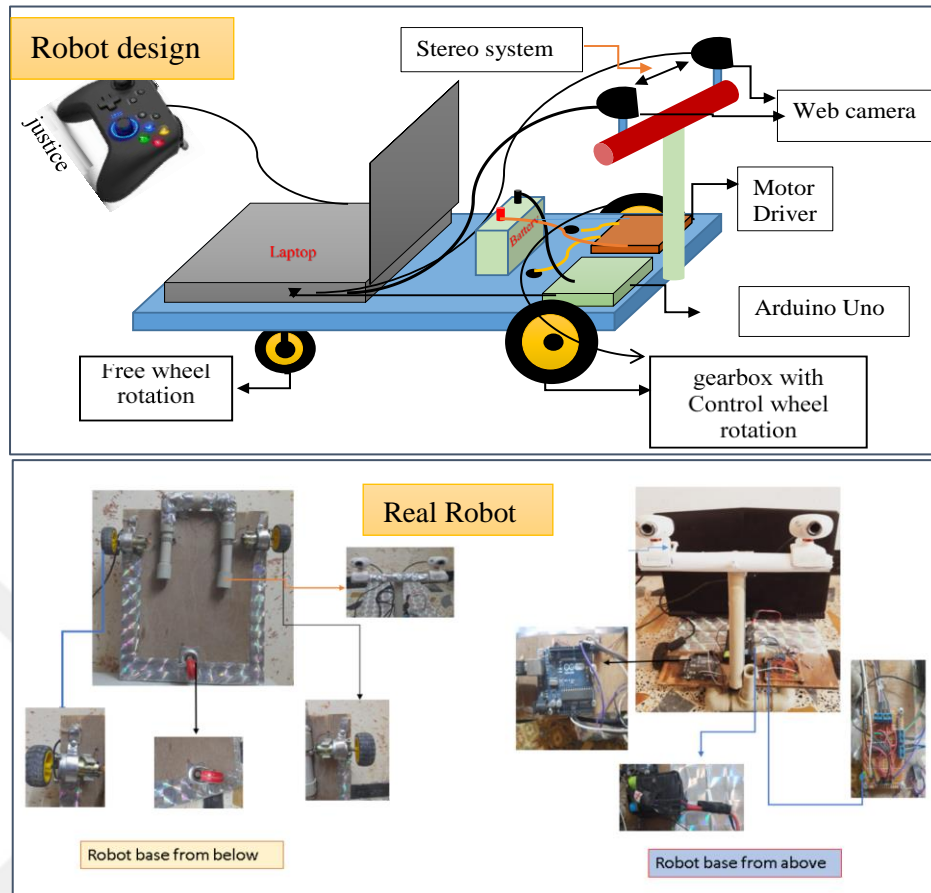


Figure B High-level overview of the deep learning system developed (Researcher's work)

## APPENDIX 13



**Figure A show the real and design robot HCMR  
(Researcher's work)**

My new dataset Aim building use stereo system install on HCMRuse depth image:-  
This data is built by navigating the mobile robot in 6 different internal environments.  
This data includes deep images with its control markers that depend on the scene in  
the image. This data divided into random images after it compiled into one file into  
two parts of the first training, which represent the 70%. The second section is test data,  
which is 30%. Having accomplished this step, we are working to separate the images  
of depth from the control markers in both parts.

[https://drive.google.com/file/d/1h3cd-5jS4nDYYvpF\\_HzF0fJYbz20mB0w/view?usp=sharing](https://drive.google.com/file/d/1h3cd-5jS4nDYYvpF_HzF0fJYbz20mB0w/view?usp=sharing)

## APPENDIX 14

### Software packages and open source libraries plus designed program

no	Package name	work
1	arduino-1.8.16	Program and configuration of Adriano
2	DS4Windows	Application use to install justiceplay4
3	Rezizeimge.py	Change size images us python
4	motor_end2.py	Work to control all part of robot HCMR
5	<pre>import pyfirmata from time import sleep import click import os #import motor_end import pprint import pygame import numpy as np import cv2 import glob import sys import math import time from matplotlib import pyplot as plt import hs_qulization as hs from tkinter import * from tkinter import simpledialog import csv</pre>	This library are use in control program
6	<pre>from __future__ import absolute_import from __future__ import division import seaborn as sns import numpy as np import torch import matplotlib.pyplot as plt from torch.autograd import Variable import torch.nn.functional as F import torch.optim as optim import torch.nn as nn from model import CNNModel from sklearn.metrics import confusion_matrix import itertools</pre>	This libraries are use in CNN training
7	<pre>import pyfirmata from time import sleep import click import os #import motor_end import pprint import pygame import numpy as np import cv2 import glob import math from matplotlib import pyplot as plt import hs_qulization as hs from tkinter import * from tkinter import simpledialog tensorflow</pre>	This libraries are use in calibration image

## CURRICULUM VITAE



<b>Full Name</b>	<b>Walead kaled Seaman</b>
<b>Academic Rank</b>	<b>PhD student in Istanbul sabahattin zaim university/faculty of engineering and natural sciences and teacher at the University of Tikrit, Faculty of Computers and Mathematics, Department of Computers</b>
<b>Field (General)</b>	<b>Computer science</b>
<b>Field (Specialization)</b>	<b>AI applications</b>
<b>Email</b>	
<b>College</b>	<b>Computers and mathematics</b>

<b>Research Interest</b>
Training and teaching mobile robots on navigation in environments unknown to them. independently using the combination of computer vision and deep learning technology

<b>Work Experience</b>
<ol style="list-style-type: none"> <li>1. Maintenance and repair of computers and their attachments programmatically and electronically</li> <li>2. Experience in building and training mobile robots through the use of Python</li> <li>3. Experience in database design and management</li> <li>4. Experience in building telecommunications networks</li> <li>5. Experience in processing medical images through the use of computer vision</li> <li>6. Experience in the following languages (C++, python, java, C #, MATLAB, Pascal, basic, HTML)</li> <li>7. Sustainable energy expertise</li> <li>8. Building electronic remote control circuits</li> </ol>

<b>Links of Academic webpages</b>			
Pueblos	Orcid	Research Gate	Google Scholar
Link <a href="https://publons.com/researcher/4482833/walead-sleman/">https://publons.com/researcher/4482833/walead-sleman/</a>	<a href="https://orcid.org/0000-0001-5446-3180">https://orcid.org/0000-0001-5446-3180</a>	<a href="https://www.researchgate.net/profile/Walead-Khalid">https://www.researchgate.net/profile/Walead-Khalid</a>	<a href="https://scholar.google.com/citations?user=FMYY2ZQsAAAAJ&amp;hl=arLink">https://scholar.google.com/citations?user=FMYY2ZQsAAAAJ&amp;hl=arLink</a>