



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Ingénierie de la Maintenance en Instrumentation

Thème

**Etude et réalisation d'un robot intelligent avec détection
par vision basée sur Raspberry Pi et Python**

Présenté et soutenu publiquement par :

SENNOUR Khalfallah

et

KHALED Norredine

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
BLKHODJ A LEILA	MAA	IMSI-Univ. D'Oran2	Présidente
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. D'Oran2	Encadreur
AISSANI NASSIMA	MCA	IMSI-Univ. D'Oran2	Examinatrice

Année 2020/2021



الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieure et de la Recherche Scientifique

جامعة وهران 2 محمد بن أحمد
Université d'Oran 2 Mohamed Ben Ahmed

معهد الصيانة و الأمن الصناعي
Institut de Maintenance et de Sécurité Industrielle

Département de Maintenance en Instrumentation

MÉMOIRE

Pour l'obtention du diplôme de Master

Filière : Génie Industriel

Spécialité : Ingénierie de la Maintenance en Instrumentation

Thème

**Study and realization of an intelligent robot with
vision detection based on Raspberry Pi and Python**

Présenté et soutenu publiquement par :

SENNOUR Khalfallah

et

KHALED Norredine

Devant le jury composé de :

Nom et Prénom	Grade	Etablissement	Qualité
BLKHODJA LEILA	MAA	IMSI-Univ. D'Oran2	Présidente
MEKKI Ibrahim El Khalil	MCA	IMSI-Univ. D'Oran2	Encadreur
AISSANI NASSIMA	MCA	IMSI-Univ. D'Oran2	Examinatrice

Année 2020/2021

Appriciation

*We thank befor all God "Allah", the Almighty, for having given us, health,
the courage, patience, desire and strength necessary, to complete our final project
studies;*

*First of all, we would like to sincerely thank our supervisor "Mr. MEKKI Ibrahim El
Khalil" Teacher at the Institute of Maintenance and Industrial Safety of Oran 2, for
having accepted to supervise this thesis, without forgetting their wise advice, their
encouragement and for his remarkable cooperation, and the support, throughout this
study...*

I also express my gratitude and sincere thanks to also the jury members

For agreeing to evaluate our modest work;

*Madame "AISSANI Nassima" teacher at the Institute of Maintenance and Industrial
Safety of the University of Oran 2, for agreeing to chair the jury*

*Madame "BELKHODJA LEILA " teacher at the Institute Maintenance and
Industrial Safety of the University of Oran 2, for agreeing to review this work*

We also we want thank our friend "BADI mohamed" and a students in this institute.

*Finally, we would also like to thank our families for their encouragement, and their
Support.*

Dedications

With the help of Almighty God, we were able to complete this work which I dedicate:

*To my **dear parents** who always trusted me .*

To my dear sisters.

To my fiancé.

To my best friends.

Khalfallah



Dedications

With the help of Almighty God, we were able to complete this work which I dedicate:

*To my **dear parents** who always trusted me .*

To my dear brothers.

To my friend “ Labri younes” and my other friends.

Nourreddine



Summary

Chapter I: Generalities about artificial intelligence

I.1 Introduction	3
I.2 Artificial Intelligence (AI).....	3
I.2.1 Definition	3
I.2.2 Operations of Artificial Intelligence	3
I.2.3 History of Artificial Intelligence	4
I.2.4 Types of artificial intelligences	5
I.3 Machine Learning (ML).....	5
I.3.1 Definition	5
I.3.2 The machine learning processes	6
I.3.3 Types of machine learning systems	7
I.4 Deep Learning	10
I.4.1 Definition	10
I.4.2 History of deep learning	11
I.4.3 Distinctive Features of Deep Learning	11
I.4.4 The neural network	12
I.4.4.1 The Layers of Neural Network	13
I.4.4.2 The types of neuron network	13

Chapter II: Computer vision

II.1 Introduction	16
II.2 The evolution of computer vision.....	16
II.3 Computer vision's principal	17
II.4 Applications of Computer Vision.....	18
II.5 the utilities of computer vision	20
II.6 the image and its representation	20
II.6.1 Binary Image	21
II.5 .2 Grayscale image	21
II.5 .3 Image color (RGB)	22
II.7 Image characteristics	22
II.7.1 Pixel	22
II.7.2 Image resolution	23
II.7.3 Image weight	23

II.7.2 Image noise	24
II.7.3 Image transparency	24
II.7.4 Image luminance	24
II.7.5 Contrast	24
II.7.6 Histogram	25
II.8 types of image formats:	25
II.8.1 Raster image files:	25
II.8.2 Vector image files:	25

Chapter III: Hardware and software tools

III.1 Introduction	27
III.2 Hardware.....	27
III.2 .1The Raspberry pi.....	27
III.2.1.1 History	27
III.2.1.2 Description of the raspberry pi board	27
III.2 .2 Materiel used	30
III.3 Software	31
III.3.1 python	31
We use python because	31
III.3.2 PyCharm IDE	32
III.3.2 libraries	32
Applications of OpenCV	33
III.3.2.2 Tensor Flow	34

Chapter IV: Implementation and programming

III.3.2.3 keras	35
IV.1 Introduction.....	37
IV.2 Hardware Components Connection.	37
IV.2 .1 List of Hardware.....	37
IV.2 .2 Assembly.....	38
IV.3 Applications.....	38
IV.3 .1 Application 01: Self Driving Lane Detection.....	38
IV.3 .1 .1 Introduction.....	38
IV.3 .1 .2 Algorithm.....	39
IV.3 .1 .3 Lane detection.....	40
IV.3 .2 Application 02: Object recognition.....	44
IV.3 .2 .1 Introduction.....	44

IV.3 .2 .2 Object recognition.....	44
IV.3 .2 Application 03: . traffic sings	47
IV.3 .2 .1 Introduction.	47
IV.3 .2 .2 Traffic sign detection and recognition using OpenCV and Python.....	47
IV.4 The Hardware implementation	50
Bibliography	54

List of tables and figures and abbreviations

List of tables

Table I. 1: A the brain and the neuron network.....	12
Table II. 1: RGB color example.....	22

List of figures

Figure I. 1: Artificial intelligence applications.....	Erreur ! Signet non défini. 4
Figure I. 2: The traditional approach.	06
Figure I. 3: Machine learning approach.	06
Figure I. 4: The process of machine learning.	07
Figure I. 5: Types of Machine Learning.	07
Figure I. 6: Regression.	08
Figure I. 7: Classification.	09
Figure I. 8: Clustering.	09
Figure I. 9: Reinforcement Learning.....	10
Figure I. 10: The relation between AI ML DL.	10
Figure I. 11: The difference between ML and DL.....	12
Figure I. 12: A node that receives three inputs.....	13
Figure I. 13: The structure of layers.....	13
Figure I. 14: Comparing a biological neuron to an artificial neuron.	13
Figure I. 15: Convolution neural network.....	14
Figure I. 16: Recurrent neural network.	14
Figure II. 1: The pixels labeled with numbers from 0–255.....	17
Figure II. 2: Storing image.	17
Figure II. 3: RGB image.....	18
Figure II. 4: Self driving cars.	18
Figure II. 5: Manufacturing.	19
Figure II. 6: Computer vision healthcare.	19
Figure II. 7: Surveillance camera.....	20
Figure II. 8: Binary image.	21
Figure II. 9: Grayscale image.....	22
Figure II. 10: RGB Color gradation.....	22
Figure II. 11: Pixels.	23
Figure II. 12: Image resolution.	23
Figure II. 13: Image noise and image clean.	24
Figure III. 1: Eben Upton (middle) and his colleagues.	27
Figure III. 2: Raspberry pi4 connection architecture.....	28
Figure III. 3: GPIO Connector Pin out.	29
Figure III. 4: The raspberry pi USB interfaces.....	29
Figure III. 5: The HDMI interface.	29
Figure III. 6: Audio and Composite interface.	30

Figure III. 7: RJ45.	30
Figure III. 8: The dc motor.....	30
Figure III. 9: Raspberry pi camera module.	30
Figure III. 10: The L298N module.	31
Figure III. 11: Logo of python.	31
Figure III. 12: Pycharm interface.	32
Figure III. 13: Open CV.....	33
Figure III. 14: Converting RGB to grayscale.....	33
Figure III. 15: Drawing shapes.....	33
Figure III. 16: Cropping and resizing images.	34
Figure III. 17: TensorFlow	34
Figure III. 18: Tensor flow Architecture.....	35
Figure III. 19: Keras.	35
Figure IV. 1: The hardware components.....	37
Figure IV. 2: The power.....	38
Figure IV. 3: The hardware components connection.....	38
Figure IV. 4: Block diagram of lane detection System.	39
Figure IV. 5: The modules of lane detection.	40
Figure IV. 6: Pixel summation.....	40
Figure IV. 7: Thresholding.	41
Figure IV. 8: Warping.....	41
Figure IV. 9: The curve direction.	42
Figure IV. 10: The lane detection module.....	43
Figure IV. 11: The webcam module.	43
Figure IV. 12: The motor module.....	43
Figure IV. 13: Overview of object Recognition computer vision tasks.....	44
Figure IV. 14: Model Detector.	45
Figure IV. 15: Import the necessary libraries.	45
Figure IV. 16: Construct the argument parse.	46
Figure IV. 17: Labels of network.....	46
Figure IV. 18: Load the Caffè model.....	46
Figure IV. 19: The Output.....	46
Figure IV. 20: CNN in traffic signs.	47
Figure IV. 21: The data loading.....	48
Figure IV. 22: Parameters.....	48
Figure IV. 23 : epochs of validation.	48
Figure IV. 24 : Loss and Acurracy curve.....	49
Figure IV. 25 : The traffic signs output.....	49
Figure IV. 26: Uploading the code	Erreur ! Signet non défini.
Figure IV. 27: Thonny environment.....	Erreur ! Signet non défini.
Figure IV. 28 : Image stack.....	Erreur ! Signet non défini.
Figure IV. 29: Car point of view.....	Erreur ! Signet non défini.

List of abbreviations

AI :	Artificial intelligence
ML:	Machine learning
DL:	Deep learning.
SSD:	Single Shot Detector.
JPEG:	Joint Photographic Expert Group.
GIF :	Graphics Interchange Format.
PNG :	Portable Graphics Format.
2D:	Two dimensional.
3D:	Three dimensional.
EPS:	Earnings per Share.
PDF:	Portable document format.
CSI:	Camera Serial Interface.
I2C:	Inter integrated circuit.
UART:	Universal Asynchronous Receiver Transmitter.
SPI:	Serial Peripheral Interface.
GPIO:	General purpose input/output.
DC:	Direct courant.
OpenCV:	Open Source Computer Vision
GPU:	Graphics processing unit.
API:	Application Programming Interface.
CNN:	Convolution neural network.
SVM:	support vector machine.

GENERAL INTRODUCTION

General introduction

General introduction

Traveling by car is currently one of the most dangerous forms of transportation, with over a million deaths annually worldwide. As nearly all car crashes (particularly fatal ones) are caused by driver error, driverless cars would effectively eliminate nearly all hazards associated with driving as well as driver fatalities and injuries.

But the world is changing, and with advancement in the field of computer vision and machine learning we can avoid or reduce this amount of deaths and car accidents happening these days by building self-driving cars.

A self-driving car is a vehicle equipped with an autopilot system and is capable of driving from one point to another without aid from a human operator.

The benefits of this cars are safe to travel, normal humans like take risks means rash driving but autonomous car obey rules and control speed of the vehicle and able to sleep in night time. Save fuel or electric power, pollution-free and reduce the vehicle thefts. It will surely reduce the majority of life threatening accidents happening all over the world due to careless driving errors made by humans.

In this thesis we made a prototype of a self-driving car and we represent three (03) applications:

The first is lane detection: Based on the vision of the camera and computer vision techniques to detect straight road and stay on it.

The second application is object detection in real time: One big application area of computer vision, the capability to locate and identify objects like (persons, animal, bicycle, even cars...), In order to protect their lives and avoid traffic accidents.

The third is traffic signs: To detecting the symbols (stop, limit speed, parking) in order to respect traffic laws.

We presented our work in 4 chapters as following:

Chapter I (Generalities about artificial intelligence): we talked about some Generalities of Artificial intelligence and its beginning its objectives as well as its fields of application like Machine learning and deep learning.

Chapter II (Computer vision): We have discussed the field of computer vision and it's evolution And some features about image processing.

Chapter III (Hardware and software tools): we presented the work environment (hardware and software) for the implementation of our project.

Chapter IV (Implementation and programming): in the final chapter we present the different stages of realization and the various algorithms and methods implemented with The improvement made also the tests and results obtained with the performance evaluation of the algorithms used.

Generalities about artificial intelligence

I.1 Introduction:

Artificial intelligence (AI) has boomed in recent years, we can see it infiltrate more and more into the daily lives of users of digital devices, there is almost no area where it cannot be found. Deep Learning is a new area of ML research, which was introduced with the aim of to bring ML closer to its main objective: artificial intelligence.

It concerns algorithms (neural networks) inspired by the structure and function of the brain. They can learn several levels of representation in order to model relationships complex between data.

I.2 Artificial Intelligence (AI)

I.2.1 Definition:

Artificial intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think like humans and mimic their actions. The term may also be applied to any machine that exhibits traits associated with a human mind such as learning and problem solving [1].

These machines are able to:

- Reason.
- Process large amounts of data.
- Discern patterns undetectable by the human eye.
- Understand and analyze these models.
- Interact with humans.
- Learn gradually.
- Continuously improve its performance.

Since 1950, when AI was created, AI has been changing. She has, even in January 2018, taken the step to go beyond intelligence human.

I.2.2 Operations of Artificial Intelligence:

According to Harry Shum, Chief Executive Officer of Microsoft, AI only works if there is presence of "a vast amount of data; extraordinary computing power, especially thanks to the cloud; and revolutionary algorithms, based on deep learning "

AI is applied today in various fields such as:

- Puzzle games.
- Mathematical research.
- Finance.
- Medicine.
- Personal assistants and home automation.
- Facial recognition and language comprehension.
- The robotics [2].

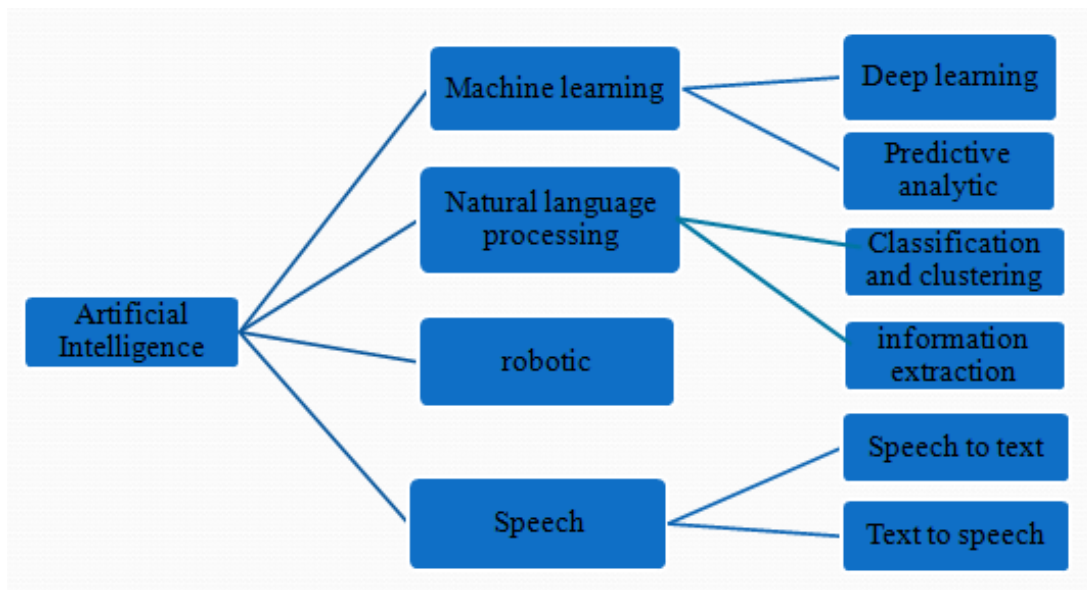


Figure I. 1: Artificial intelligence applications.

I.2.3 History of Artificial Intelligence:

Artificial intelligence is a field that has emerged recently and has around sixty years or so. This complex field which mixes sciences, theories and techniques has for ambition to imitate the cognitive and intelligent capacities of a human being.

It all started in 1940-1960; the idea of making the link between the functioning of machines and organic beings emerges. The ultimate goal of Norbert Wiener, pioneer of cybernetics, was to combine different fields which are mathematics, electronics and automation while combining the animal dimension.

In 1950, decimal logic and binary logic were set up in machines automated, the first real computers, which allow them to do what they want programmed. The scientist calls this effect an "imitation game". He starts the debate on the questioning of the limit between human and machine.

Between 1980 and 1990, expert systems appear which are based on an "engine of inference ", i.e. software that corresponds to an algorithm for simulating deductive reasoning, which was programmed to be a logical mirror of a human reasoning. Therefore, when entering data, the engine provides answers a high level of expertise.

In May 1997, IBM's expert system "Deep Blue" won in a chess game against the world champion Garry Kasparov. Deep Blue was designed based on a systematic brute force algorithm that is to say that all possible blows were assessed and weighted.

This phenomenon was marked by the defeat of the human face machine and remained very symbolic throughout history, although Deep blue was not actually managed to deal with only a very limited perimeter which is that of the rules of the chess game.

Since 2010, there has been a new boom due to the collection of massive data and new computing power resulting from two explanatory factors of this new era of AI [3].

I.2.4 Types of artificial intelligences:

Unlike the levels of artificial intelligences which are based on one's ability, these types are based on functionality, and there are four types (Reactivity, limited memory, theory of mind, self-awareness)

- **Type I: Responsiveness:**

It is the most basic stage of artificial intelligence, and the most developed currently: a machine capable of perceiving the surrounding world and acting according to these perceptions (the best illustration being Deep Blue, the computer of IBM who beat chess champion Garry Kasparov in 1996). According to some thinkers of artificial intelligence, research should focus on the development of this type of machine alone

- **Type II: Limited memory:**

Unlike reactive machines, machines with limited memory have the ability to rely on representations of the world to make a decision. Self-driving cars are a good example: with a repertoire of pre-programmed representations of the world, they are able to adjust their speed or trajectory depending on several factors.

- **Type III: Theory of mind:**

The third type of artificial intelligence marks a breaking point between the machines that exist and those that we will build in the future. Thanks to the theory of mind, robots will not only be able to grasp and classify representations of the world, but also to understand and prioritize the emotions that can influence human behavior, and to adapt their behavior accordingly

- **Type IV: Self consciousness:**

Self-awareness is the final step in the development of artificial intelligence. Self consciousness is about endowing machines with the capacity to form representations of themselves, Like a Type III extension of artificial intelligence, self conscious machines not only understand human emotions, but manage to apply them to themselves, while being able to predict those of others [4].

I.3 Machine Learning (ML)

I.3.1 Definition:

Machine learning is the science and art of programming computers so they can learn from data., applying ML techniques to dig into large amounts of data can help discover patterns that were not immediately apparent. This is called data mining [5].

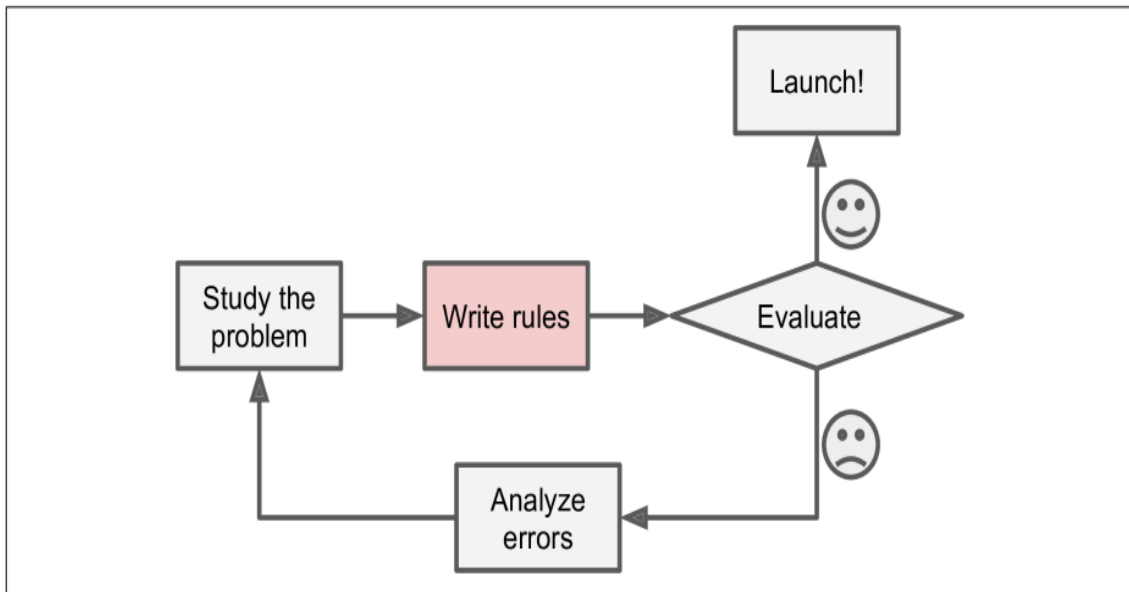


Figure I. 2: The traditional approach.

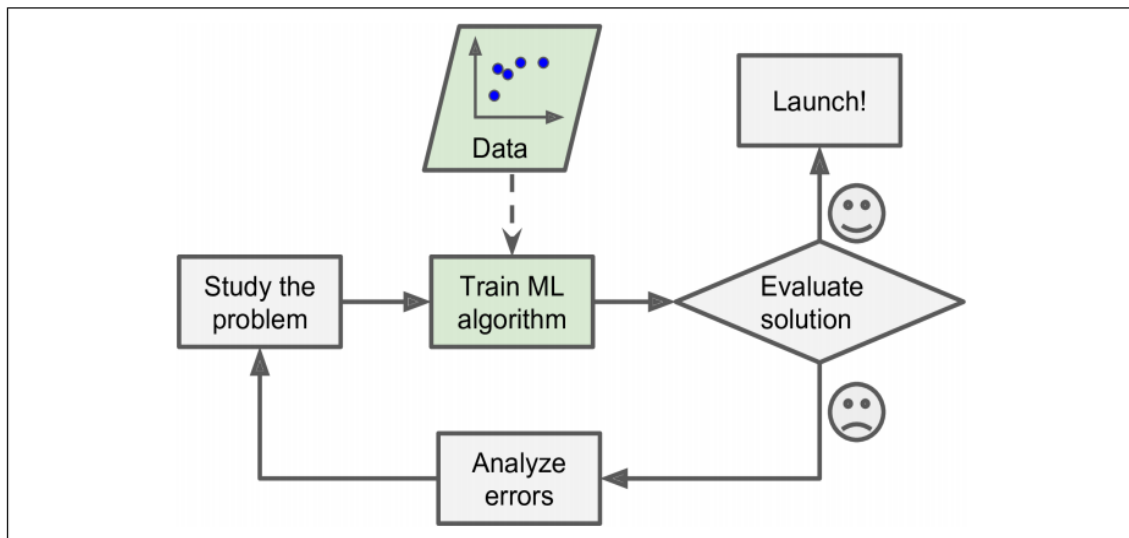


Figure I. 3: Machine learning approach.

I.3.2 The machine learning processes:

Machine Learning is a technique that figures out the “model” out of “data.” Here, the data literally means information such as documents, audio, images, etc. The “model” is the final product of Machine Learning. In short, Machine Learning is a modeling technique that involves data.

It helped solve many problems that were historically challenging for classical software development tools and approaches. For instance, years ago, machine learning engineers were able to create software that could predict breast cancer survival windows better than human experts. However building the features of the software required the efforts of dozens of engineers and breast cancer experts and took a lot of time develop[6].

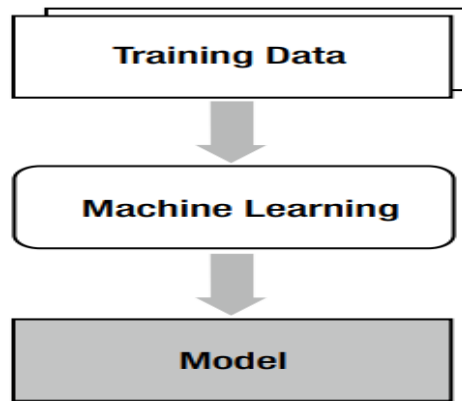


Figure I. 4: The process of machine learning.

Machine Learning is great for:

- Problems for which existing solutions require a lot of hand tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better.
- Complex problems for which there is no good solution at all using a traditional approach: the best Machine Learning techniques can find a solution.
- Fluctuating environments: a Machine Learning system can adapt to new data.
- Getting insights about complex problems and large amounts of data [7].

I.3.3 Types of machine learning systems:

There are many different types of Machine Learning techniques have been developed to solve problems in various fields. These Machine Learning techniques can be classified into three types depending on the training method.

- Supervised learning
- Unsupervised learning
- Reinforcement learning [8].

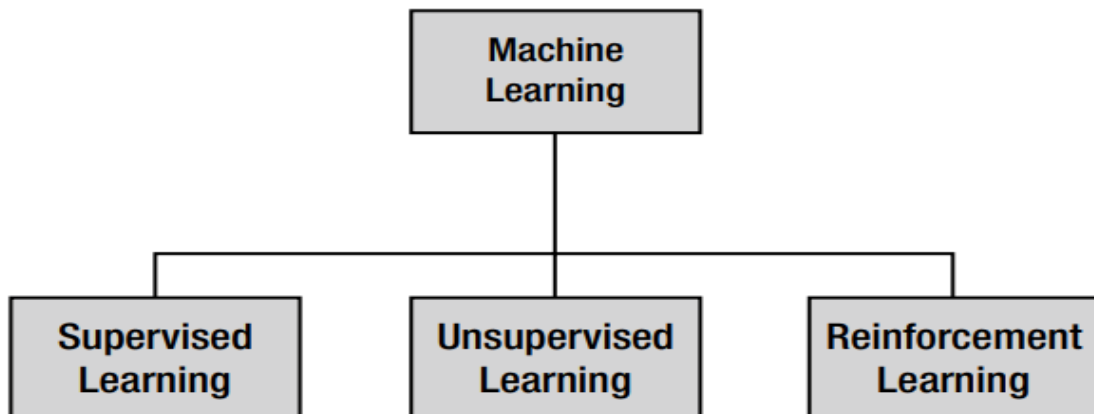


Figure I. 5: Types of Machine Learning.

Machine Learning systems can be classified according to the amount and type of supervision they get during training. There are three major categories:

- **Supervised learning:**

In supervised learning, the training data you feed to the algorithm includes the desired solutions, called label [8].

- **Regression analysis:**

Consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

Briefly, the goal of regression model is to build a mathematical equation that defines y as a function of the x variables [9].

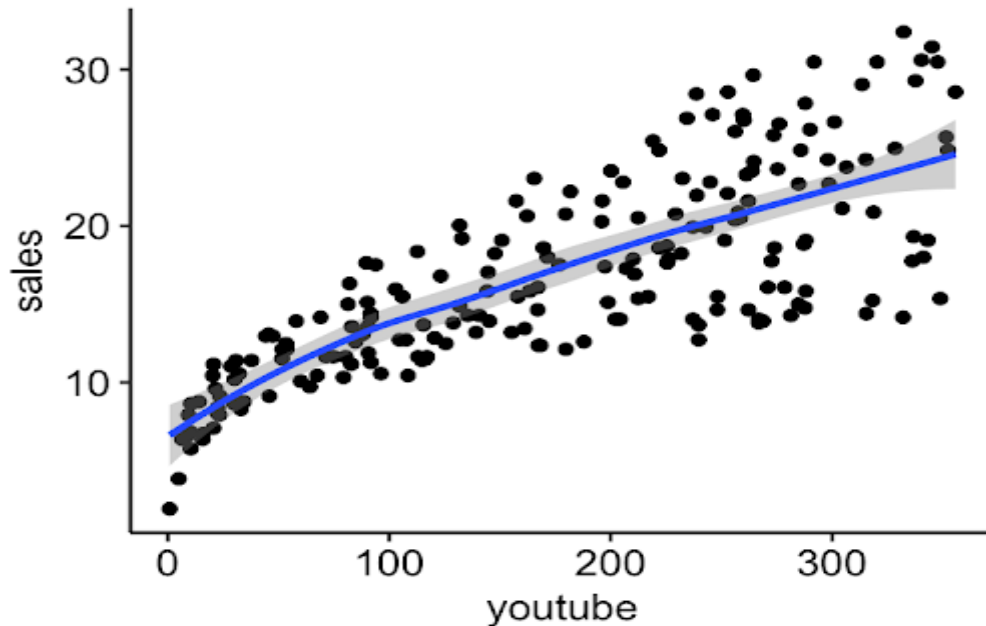


Figure I. 6: Regression.

- **Classification:**

The Classification algorithm is a Supervised Learning technique that is used to identify the category of new observations on the basis of training data. In Classification, a program learns from the given dataset or observations and then classifies new observation into a number of classes or groups.

The algorithm which implements the classification on a dataset is known as a classifier there are two types of Classifications:

- **Binary Classification:** If the classification problem has only two possible outcomes then it is called as Binary Classifier
Examples: YES or NO, MALE or FEMALE, SPAM or NOT SPAM, CAT or DOG, etc.
- **Multi-Classification:** If a classification problem has more than two outcomes, then it is called as Multi-class Classifier.
Example: Classifications of types of crops, Classification of types of music [10]

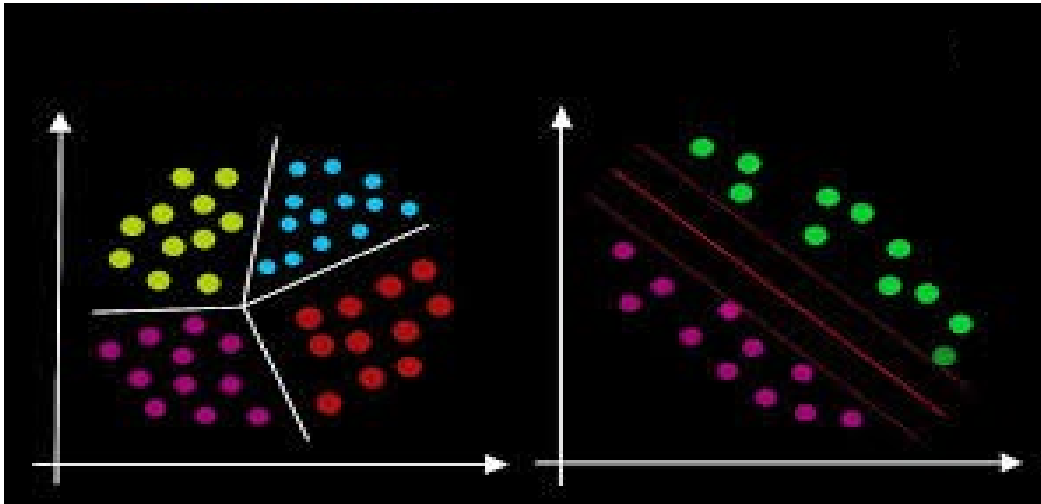


Figure I. 7: Classification.

➤ **Unsupervised learning:**

In unsupervised learning, as you might guess, the training data is unlabeled the system tries to learn without a teacher.

• **Clustering:**

Is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them. [11]

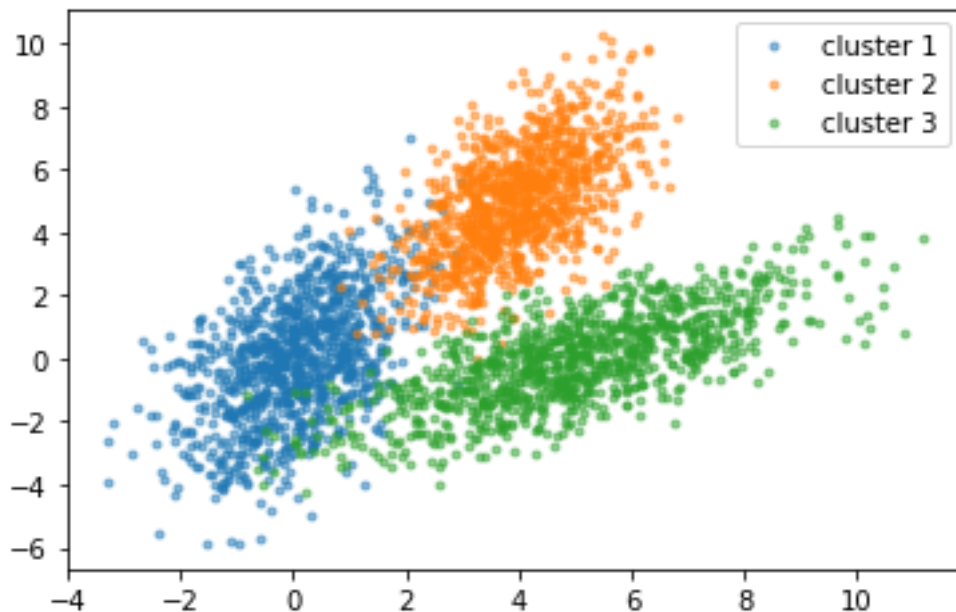


Figure I. 8: Clustering.

• **Reinforcement Learning:**

Reinforcement Learning is a very different beast. The learning system, called an agent in this context, can observe the environment, select and perform actions, and get rewards in. It must then learn by itself what is the best strategy, called a policy, to get the most reward over time ,a policy defines what action the agent should choose when it is in a given situation [12].

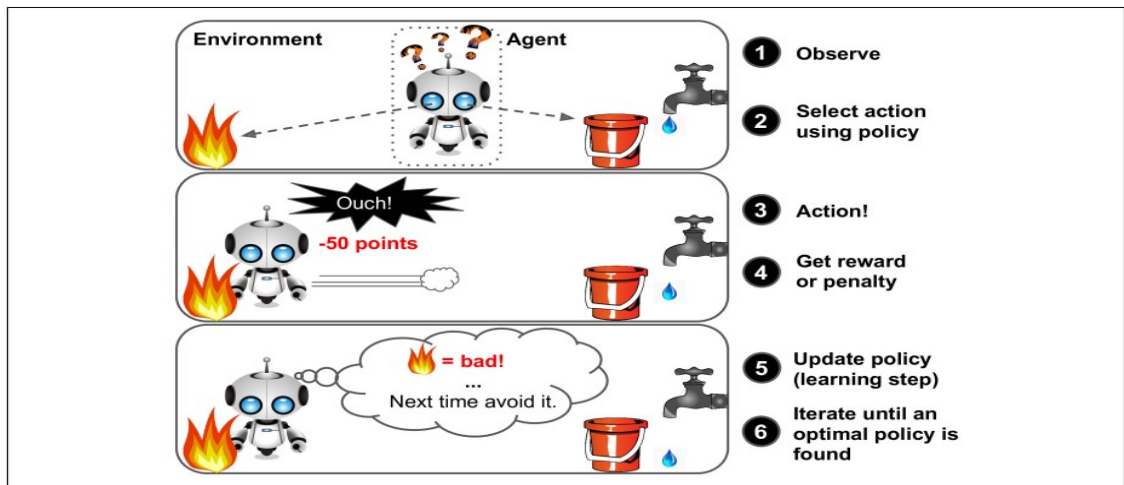


Figure I. 9: Reinforcement Learning.

I.4 Deep Learning:

In the past few years, artificial intelligence (AI) has been a subject of intense media hype. Machine learning, deep learning, and AI come up in countless articles, often outside of technology-minded publications. We're promised a future of intelligent chatbots, self-driving cars, and virtual assistants a future sometimes painted in a grim light and other times as utopian, where human jobs will be scarce and most economic activity will be handled by robots or AI agents. For a future or current practitioner of machine learning, it's important to be able to recognize the signal in the noise so that you can tell world-changing developments from overhyped press release.

I.4.1 Definition:

Deep Learning is a type of Machine Learning is very important, and that is why we are going through this lengthy review on how Artificial Intelligence, Machine Learning, and Deep Learning are related. Deep Learning has been in the spotlight recently as it has proficiently solved some problems that have challenged Artificial Intelligence [13].

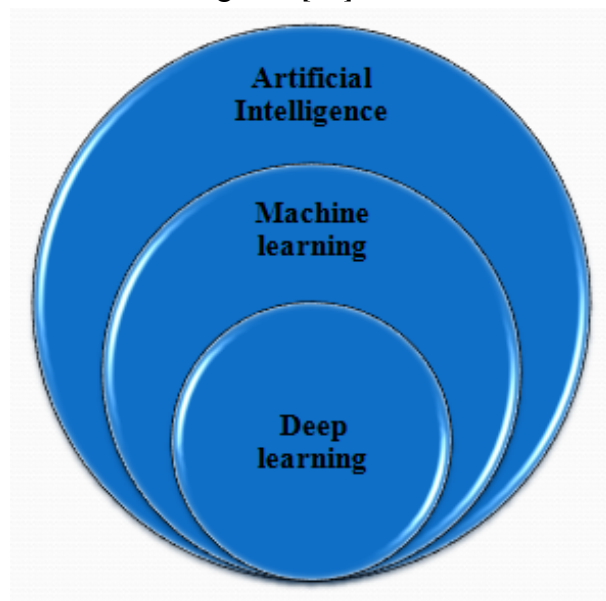


Figure I. 10: The relation between AI ML DL.

I.4.2 History of deep learning:

Deep learning is based on the concept of artificial neural networks, or computational systems that mimic the way the human brain functions. And so, our brief history of deep learning must start with those neural networks.

1943: Warren McCulloch and Walter Pitts create a computational model for neural networks based on mathematics and algorithms called threshold logic.

1958: Frank Rosenblatt creates the perceptron, an algorithm for pattern recognition based on a two layer computer neural network using simple addition and subtraction. He also proposed additional layers with mathematical notations, but these wouldn't be realized until 1975.

1980: Kunihiko Fukushima proposes the Neoconitron, a hierarchical, multilayered artificial neural network that has been used for handwriting recognition and other pattern recognition problems.

1989: Scientists were able to create algorithms that used deep neural networks, but training times for the systems were measured in days, making them impractical for real-world use.

1992: Juyang Weng publishes Cresceptron, a method for performing 3D object recognition automatically from cluttered scenes.

2000: The term "deep learning" begins to gain popularity after a paper by Geoffrey Hinton and Ruslan Salakhutdinov showed how a many-layered neural network could be pre-trained one layer at a time.

2009: NIPS Workshop on Deep Learning for Speech Recognition discovers that with a large enough data set, the neural networks don't need pre-training, and the error rates drop significantly.

2012: Artificial pattern-recognition algorithms achieve human-level performance on certain tasks. And Google's deep learning algorithm discovers cats.

2014: Google buys UK artificial intelligence startup Deep mind for £400m.

2015: Facebook puts deep learning technology - called Deep Face - into operations to automatically tag and identify Facebook users in photographs. Algorithms perform superior face recognition tasks using deep networks that take into account 120 million parameters.

2016: Google Deep Mind's algorithm AlphaGo masters the art of the complex board game Go and beats the professional go player Lee Sedol at a highly publicized tournament in Seoul [14].

I.4.3 Distinctive Features of Deep Learning:

Big advantage with deep learning, and a key part in understanding why it's becoming popular, is that it's powered by massive amounts of data

Chapter I: Generalities about artificial intelligence

In traditional Machine learning techniques, most of the applied features need to be identified by a domain expert in order to reduce the complexity of the data and make patterns more visible to learning algorithms to work.

The biggest advantage Deep Learning algorithms as discussed before are that they try to learn high-level features from data in an incremental manner. This eliminates the need of domain expertise and hard core feature extraction.

Another major difference between Deep Learning and Machine Learning technique is the problem solving approach. Deep Learning techniques tend to solve the problem end to end, where as Machine learning techniques need the problem statements to break down to different parts to be solved first and then their results to be combine at final stage [15].

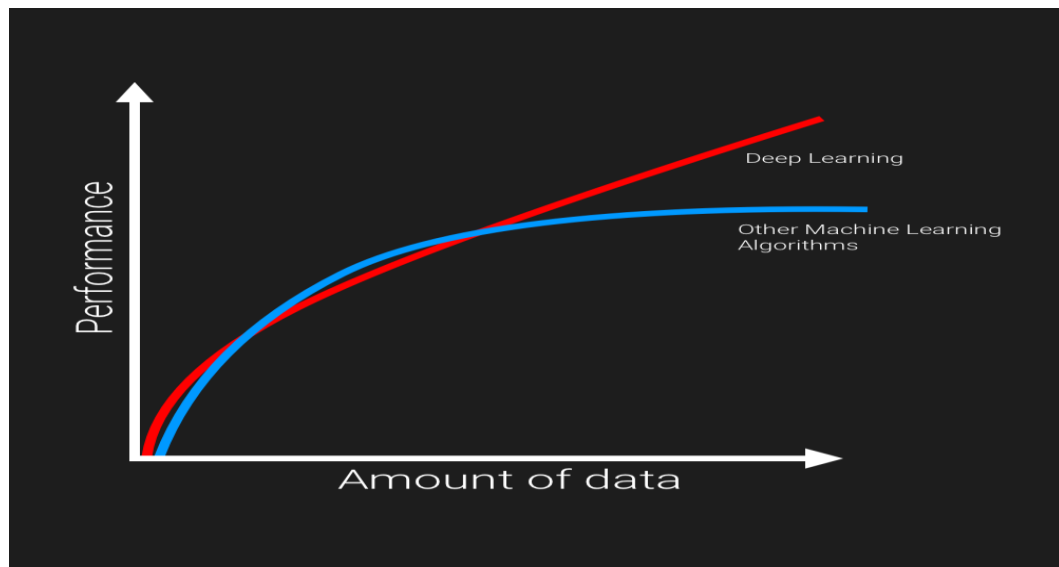


Figure I. 11: The difference between ML and DL.

I.4.4 The neural network:

The neural network imitates the mechanism of the brain. As the brain is composed of connections of numerous neurons, the neural network is constructed with connections of nodes, which are elements that correspond to the neurons of the brain. The neural network mimics the neurons' association, which is the most important mechanism of the brain, using the weight value.

The following table summarizes the analogy between the brain and neural network [16].

Table I. 1: A the brain and the neuron network

Brain	Neural Network
Neuron	node
Connection of neurons	Connection weight

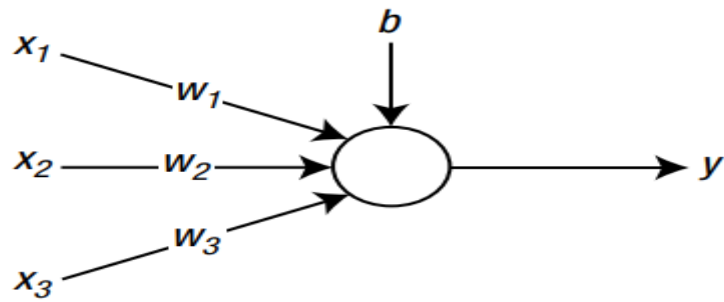


Figure I. 12: A node that receives three inputs.

I.4.4.1 The Layers of Neural Network:

As the brain is a gigantic network of the neurons, the neural network is a network of nodes. A variety of neural networks can be created depending on how the nodes are connected. One of the most commonly used neural network types employs a layered structure of nodes as shown in Figure below [17].

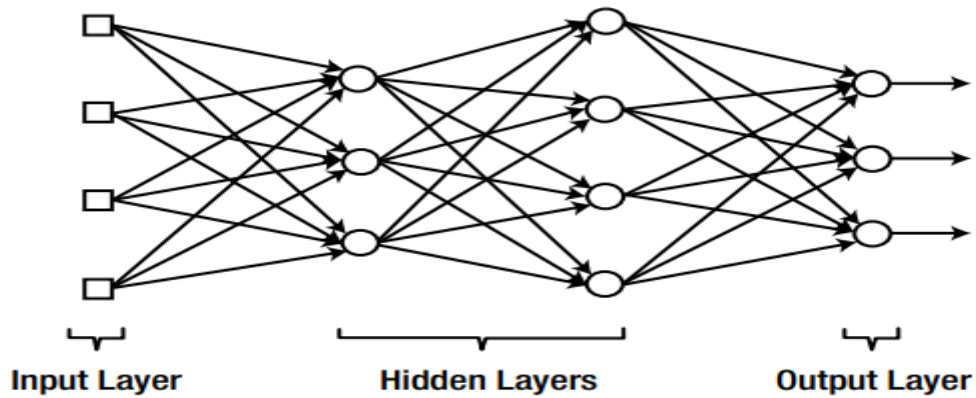


Figure I. 13: The structure of layers.

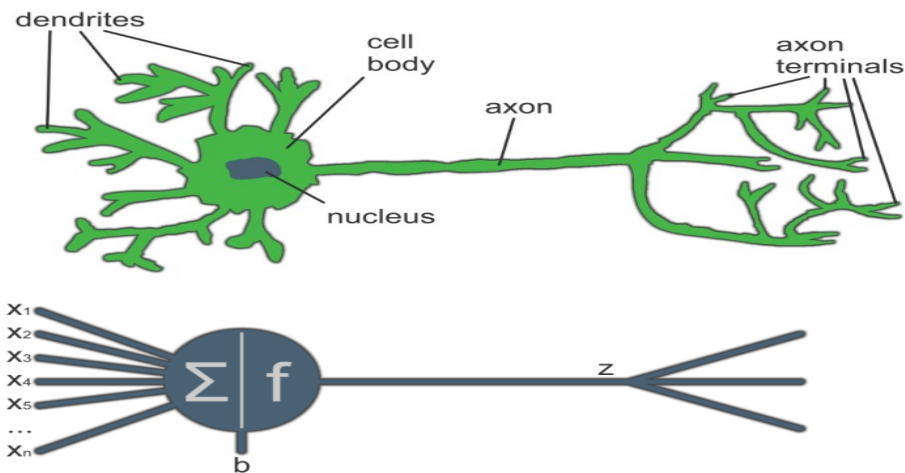


Figure I. 14: Comparing a biological neuron to an artificial neuron.

I.4.4.2 The types of neuron network:

There are many types of neural networks available or that might be in the development stage. They can be classified depending on their: Structure, Data flow, Neurons used and their density, Layers and their depth activation filters etc [18].

➤ Convolution neuron network

Convolution neural network contains a three-dimensional arrangement of neurons, instead of the standard two-dimensional array. The first layer is called a convolution layer. Each neuron in the convolution layer only processes the information from a small part of the visual field. Input features are taken in batch wise like a filter. The network understands the images in parts and can compute these operations multiple times to complete the full image processing. Processing involves conversion of the image from RGB or HSI scale to grey-scale. Furthering the changes in the pixel value will help to detect the edges and images can be classified into different categories [18].

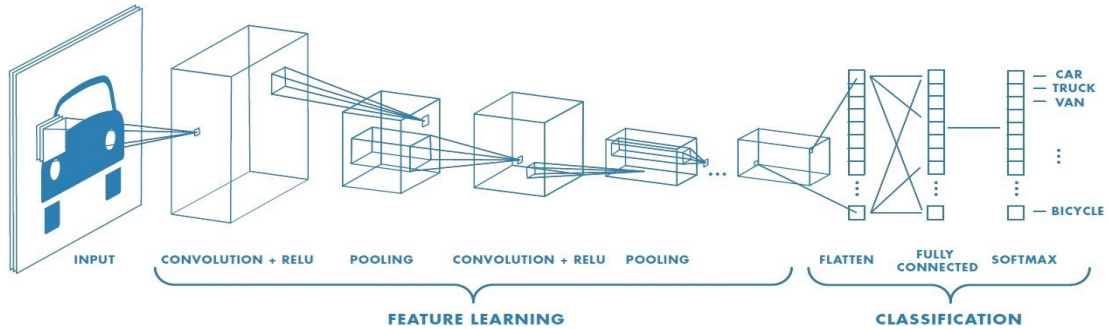


Figure I. 15: Convolution neural network.

➤ Recurrent neural networks

Recurrent Neural Network is fed back to the input to help in predicting the outcome of the layer. The first layer is typically a feed forward neural network followed by recurrent neural network layer where some information, it had in the previous time-step is remembered by a memory function. Forward propagation is implemented in this case. It stores information required for its future use. If the prediction is wrong, the learning rate is employed to make small changes. Hence, making it gradually increase towards making the right prediction during the back propagation [18].

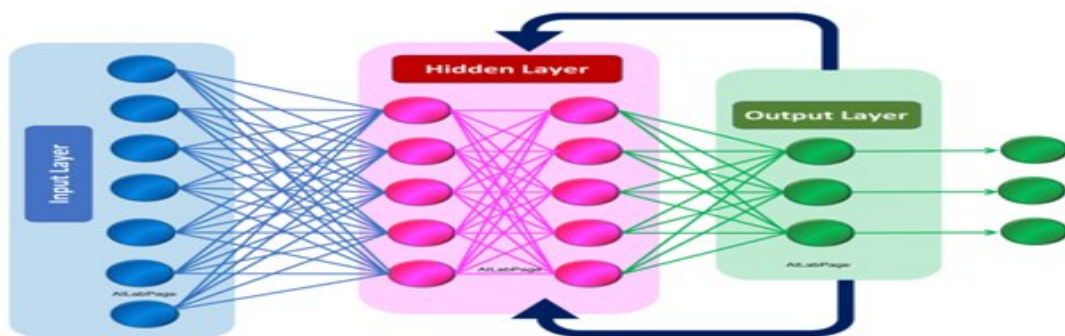


Figure I. 16: Recurrent neural network.

➤ Feed forward neural networks

A Feed forward neural network is an artificial neural network in which the connection between nodes does not form a cycle. The opposite of a feed forward neural network is a recurrent neural network, in which certain pathways are cycled. The feed forward model is the simplest form of neural network as information is only processed in one direction. While the data may pass through multiple hidden nodes, it always moves in one direction and never backwards [19].

Chapter II: Computer vision

Chapter II: Computer vision

II.1 introduction

Computer vision is a subfield of artificial intelligence. The purpose of computer vision is to program a computer to "understand" a scene or features in an image.

These goals are achieved by means of pattern recognition, statistical learning, projective geometry, image processing, graph theory and other fields. Cognitive computer vision is strongly related to cognitive psychology and biological computation.

In the related fields machine vision and medical imaging, systems using computer vision techniques are sold in markets worth billions of US dollars per year.

One interesting application of computer vision, commonly used in the creation of visual effects for cinema and broadcast, is camera tracking or match moving. Computer vision also finds its applications in medicine, military industry, security and surveillance, quality inspection, robotics, automotive industry and many other fields [20].

II.2 The evolution of computer vision

Before the advent of deep learning, the tasks that computer vision could perform were very limited and required a lot of manual coding and effort by developers and human operators. For instance, if you wanted to perform facial recognition, you would have to perform the following steps:

- **Create a database:** You had to capture individual images of all the subjects you wanted to track in a specific format.
- **Annotate images:** Then for every individual image, you would have to enter several key data points, such as distance between the eyes, the width of nose bridge, distance between upper-lip and nose, and dozens of other measurements that define the unique characteristics of each person.
- **Capture new images:** Next, you would have to capture new images, whether from photographs or video content. And then you had to go through the measurement process again, marking the key points on the image. You also had to factor in the angle the image was taken.

After all this manual work, the application would finally be able to compare the measurements in the new image with the ones stored in its database and tell you whether it corresponded with any of the profiles it was tracking. In fact, there was very little automation involved and most of the work was being done manually. And the error margin was still large.

Machine learning provided a different approach to solving computer vision problems. With machine learning, developers no longer needed to manually code every single rule into their vision applications. Instead they programmed "features", smaller applications that could detect specific patterns in images. They then used a statistical learning algorithm such as linear regression, logistic regression, decision trees or support vector machines (SVM) to detect patterns and classify images and detect objects in them [21].

Chapter II: Computer vision

II.3 Computer vision's principal

One of the major open questions in both Neuroscience and Machine Learning is: How exactly do our brains work, and how can we approximate that with our own algorithms? The reality is that there are very few working and comprehensive theories of brain computation; so despite the fact that Neural Nets are supposed to “mimic the way the brain works”, nobody is quite sure if that’s actually true.

The same paradox holds true for computer vision since we’re not decided on how the brain and eyes process images, it’s difficult to say how well the algorithms used in production approximate our own internal mental processes.

On a certain level Computer vision is all about pattern recognition. So one way to train a computer how to understand visual data is to feed it images, lots of images thousands, millions if possible that have been labeled, and then subject those to various software techniques, or algorithms, that allow the computer to hunt down patterns in all the elements that relate to those labels.

Below is a simple illustration of the grayscale image buffer which stores our image of Abraham Lincoln. Each pixel’s brightness is represented by a single 8-bit number, whose range is from 0 (black) to 255 (white):

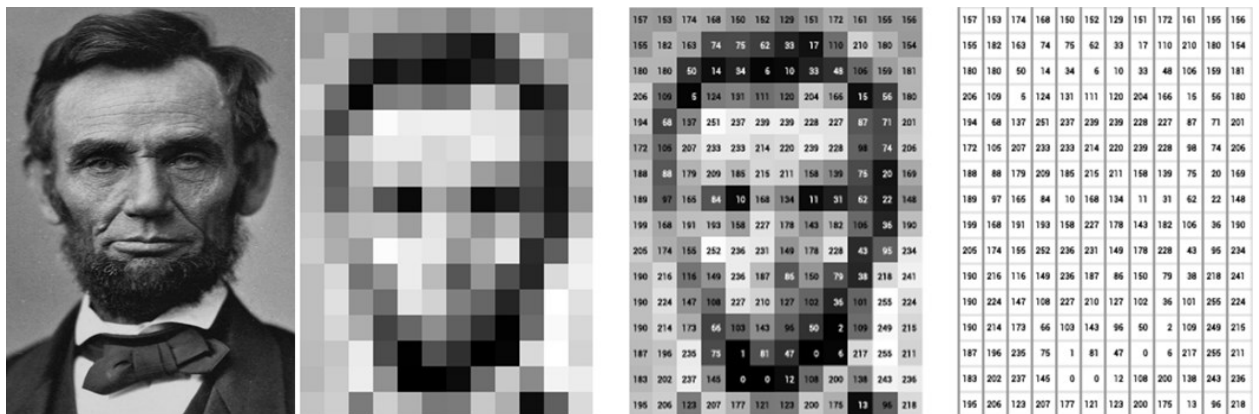


Figure II. 1: The pixels labeled with numbers from 0–255

This way of storing image data may run counter to your expectations, since the data certainly appears to be two dimensional when it is displayed. Yet, this is the case, since computer memory consists simply of an ever increasing linear list of address spaces.

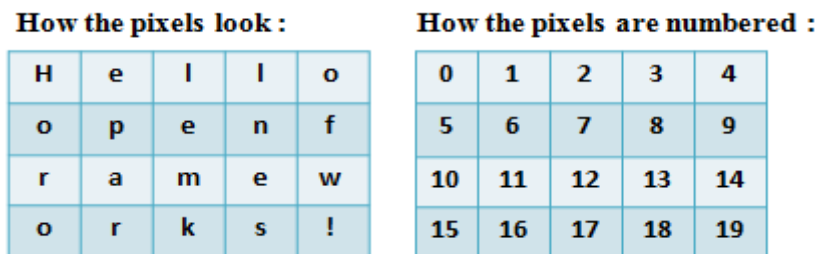


Figure II. 2: Storing image.

Now things start to get more complicated. Computers usually read color as a series of 3 values red, green, and blue (RGB) on that same (0–255) scale. Now, each pixel actually has 3 values for the computer to store in addition to its position. If we were to

Chapter II: Computer vision

colorize President Lincoln, that would lead to $12 \times 16 \times 3$ values, or 576 numbers computer to store in addition to its position if we were to colorize President Lincoln, that would lead to $12 \times 16 \times 3$ values, or 576 numbers.[21]

					
R:	255	0	0	0	255
G:	0	255	0	0	255
B:	0	0	255	0	255

Figure II. 3: RGB image.

II.4 Applications of Computer Vision

Automotive:

Some of the most famous applications of computer vision have been done by Tesla with their Autopilot function.



Figure II. 4: Self driving cars.

Manufacturing:

Computer vision coupled with sensors can work wonders for critical equipment. Today, the technology is being used to check on important plants or equipment in there.

Infrastructure faults and problems can be prevented with the help of computer vision that is wise enough to estimate its health and efficiency. Many companies are syncing predictive maintenance with their infrastructure to keep their tools in good shape.

Chapter II: Computer vision



Figure II. 5: Manufacturing.

Healthcare:

In healthcare, computer vision has the potential to bring in some real value. While computers won't completely replace healthcare personnel, there is a good possibility to complement routine diagnostics that require a lot of time and expertise of human physicians but don't contribute significantly to the final diagnosis. This way computer serves as a helping tool for the healthcare personnel.

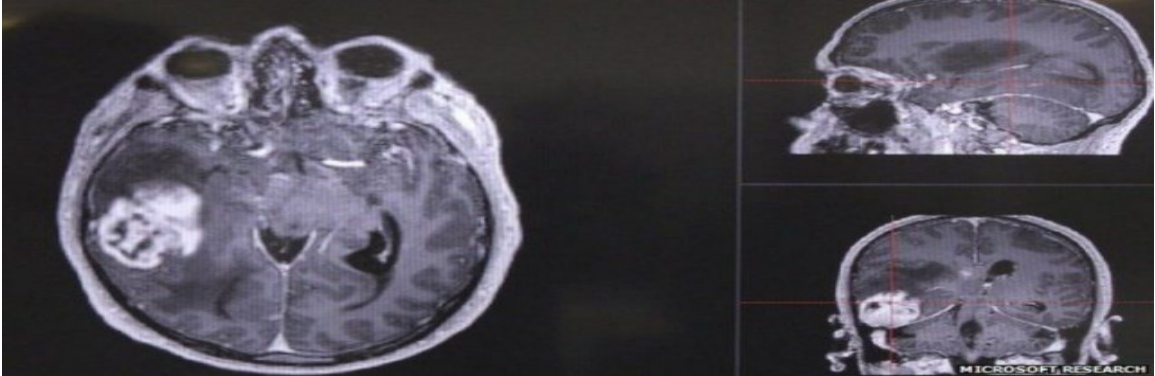


Figure II. 6: Computer vision healthcare.

Surveillance:

The innovation enables security of public places like parking lots, the subway, railways and bus stations, roads and highways, etc.

The application of computer vision for security purposes is diverse. It's face recognition, crowd detection, human abnormal behavior detection, illegal parking detection, speeding vehicle detection and more.

The technology helps strengthen security and prevent accidents of various kinds. Racetrack unveiled surveillance solutions that detect abnormal activities and inform managers to intervene [22].

Chapter II: Computer vision



Figure II. 7: Surveillance camera.

II.5 The utilities of computer vision:

Amazon:

Unveiled 18 AmazonGo stores where shoppers can bypass lines and pay for items right away. With computer vision, cameras are used to let employees know when something was taken off the shelves. It can also identify returned items or removed items from a shopping cart. Your Amazon prime account will be charged once you finish filling your “virtual basket.”

Computer vision in retail stores it can also improve security. Tracking each person inside the store all time makes sure each shopper pays for the merchandise.

Facebook:

Uses facial recognition (Deep Face) when automatically tagging photos that are posted to your profile. After negative feedback from many audiences due to privacy, Facebook only allows the recognition if you opt into it.

Military Space:

Countries across the globe are embedding AI into weapons, transportation, and target recognition, healthcare in the field, simulation training, and other systems used on land, air, sea, and space. AI systems based on these platforms are less reliant on human input because they enhance performance while requiring less maintenance. With current systems, AI decreases cyber-attacks and can protect networks, computers, programs, and data from any unauthorized access [23].

II.6 the image and its representation:

Humans derive a great deal of information about the world through their visual sense. Light reflects objects and sometimes passes through objects to create an image on the retina of each eye. From this pair of images much of the structure of the 3D environment is derived.

Chapter II: Computer vision

The important components are thus a scene of objects, illumination of the objects, and sensing the illumination re acting o the objects (or passing through them).

The major purpose of this chapter is to describe how sensors produce digital images of 2D or 3D scenes.

The 2D digital image is an array of intensity samples re acted from or transmitted through objects: this image is processed by a machine or computer program in order to make decisions about the scene. often, a 2D image represents a projection of a 3D scene, this is the most common representation used in machine vision [24].

II.6.1 Binary Image:

Binary images are images whose pixels have only two possible intensity values. Numerically, the two values are often 0 for black, and either 1 for white [25].

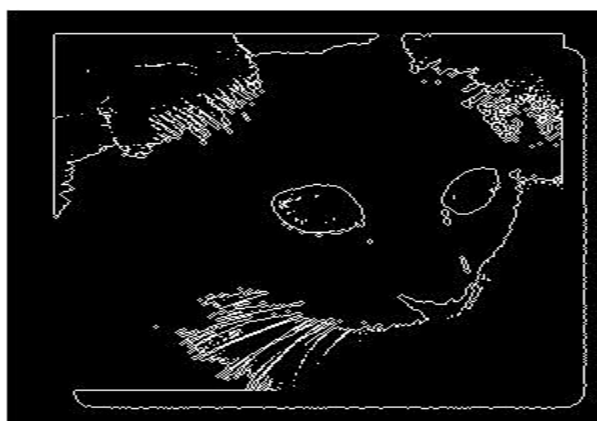


Figure II. 8: Binary image.

II.5 .2 Grayscale image:

A grayscale (or gray level) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel.

In a gray-scale image, each pixel has a value between 0 and 255, where zero corresponds to “black” and 255 corresponds to “white”. The values in between 0 and 255 are varying shades of gray, where values closer to 0 are darker and values closer to 255 are lighter.

Gray-scale images are very common, in part because much of today’s display and image capture hardware can only support 8-bit images.

In addition, gray-scale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images [26].



Figure II. 9: Grayscale image

II.5 .3 Image color (RGB):

In the most common color space, RGB (Red Green Blue), colors are represented in terms of their red, green, and blue components. In more technical terms, RGB describes a color as a tuple of three components. Each component can take a value between 0 and 255, where the tuple (0, 0, 0) represents black and (255, 255, 255) represents white.

RGB is considered an “additive” color space, and colors can be imagined as being produced from shining quantities of red, blue, and green light onto a black background. [27]

Here are a few more examples of colors in RGB

Table II. 1: RGB color example.

Color	RGB values
Red	(255, 0, 0)
Orange	(255, 128, 0)
Pink	(255, 153, 255)



Figure II. 10: RGB Color gradation.

II.7 Image characteristics:

II.7.1 Pixel:

In digital imaging, a pixel (or picture element) is the smallest item of information in an image. Pixels are arranged in a 2-dimensional grid, represented using squares. Each pixel is a sample of an original image, where more samples typically provide more-accurate representations of the original. The intensity of each pixel is variable; in color

Chapter II: Computer vision

systems, each pixel has typically three or four components such as red, green, and blue, or cyan, magenta, yellow, and black. [28].

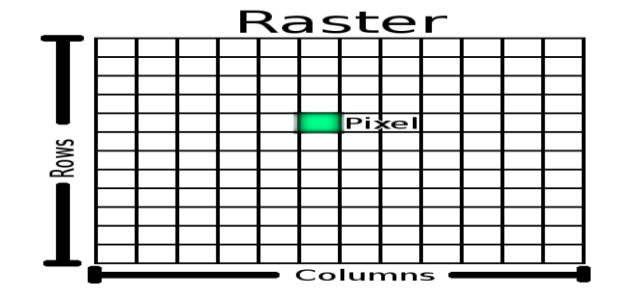


Figure II. 11: Pixels.

II.7.2 Image resolution:

The term resolution is often used as a pixel count in digital imaging. When the pixel counts are referred to as resolution, the convention is to describe the pixel resolution with the set of two numbers.

The first number is the number of pixel columns (width) and the second is the number of pixel rows (height), for example as 640 by 480. Another popular convention is to cite resolution as the total number of pixels in the image, typically given as number of megapixels, which can be calculated by multiplying pixel columns by pixel rows and dividing by one million. An image that is 2048 pixels in width and 1536 pixels in height has a total of $2048 \times 1536 = 3,145,728$ pixels or 3.1 megapixels.

One could refer to it as 2048 by 1536 or a 3.1-megapixel image. Other conventions include describing pixels per length unit or pixels per area unit, such as pixels per inch or per square inch. [28].

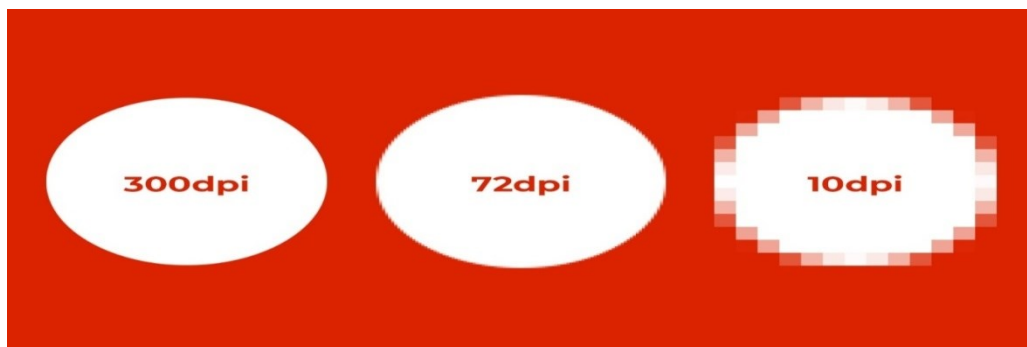


Figure II. 12: Image resolution.

II.7.3 Image weight:

This is the size of the image. Since the latter is represented in the form of a matrix whose values represent the pixel intensity), the number of columns (W) multiplied by the number of lines (H) gives the total number of pixels in the image.

For a 640x480 color image:

Number of pixels = $640 \times 480 = 307200$

Weight of each pixel = 3 bytes

Image weight = $307200 \times 3 = 921600$ bytes = 900 KB [29].

Chapter II: Computer vision

II.7.2 Image noise:

The original meaning of "noise" was "unwanted signal"; unwanted electrical fluctuations in signals received by AM radios caused audible acoustic noise ("static"). By analogy, unwanted electrical fluctuations are also called "noise".

Image noise can range from almost imperceptible specks on a digital photograph taken in good light, to optical and radio astronomical images that are almost entirely noise, from which a small amount of information can be derived by sophisticated processing. Such a noise level would be unacceptable in a photograph since it would be impossible even to determine the subject [30].

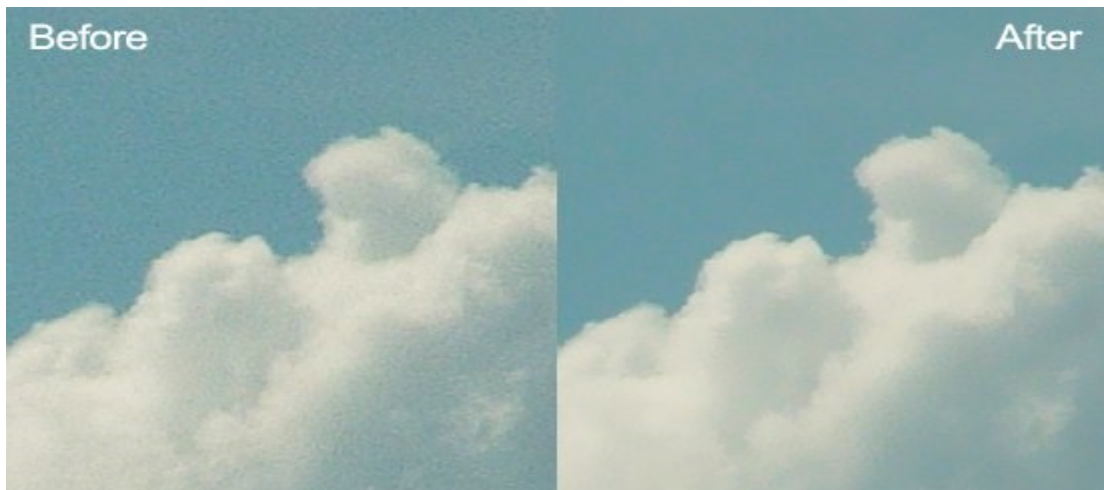


Figure II. 13: Image noise and image clean.

II.7.3 Image transparency:

Transparency is possible in a number of graphics file formats. The term "transparency" is used in various ways by different people, but at its simplest there is "full transparency" i.e. something that is completely invisible.

Only part of a graphic should be fully transparent, or there would be nothing to see. More complex is "partial transparency" or "translucency" where the effect is achieved that a graphic is partially transparent in the same way as colored glass. Since ultimately a printed page or computer or television screen can only be one color at a point, partial transparency is always simulated at some level by mixing colors. There are many different ways to mix colors, so in some cases transparency is ambiguous [31].

II.7.4 Image luminance:

In luminance, each pixel is measured in bits, with 0 meaning total darkness and 1 meaning total white. The bit depth of an image has become synonymous with luminance. When capturing images with a camera, the color and light information in an image [32].

II.7.5 Contrast:

Contrast is the difference in luminance or color that makes an object (or its representation in an image or display) distinguishable. In visual perception of the real

Chapter II: Computer vision

world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. The human visual system is more sensitive to contrast than absolute luminance; we can perceive the world similarly regardless of the huge changes in illumination over the day or from place to place. The maximum contrast of an image is the contrast ratio or dynamic range [33].

II.7.6 Histogram:

An image histogram is a graphical representation of the number of pixels in an image as a function of their intensity.

Histograms are made up of bins, each bin representing a certain intensity value range. The histogram is computed by examining all pixels in the image and assigning each to a bin depending on the pixel intensity. The final value of a bin is the number of pixels assigned to it. The number of bins in which the whole intensity range is divided is usually in the order of the square root of the number of pixels.

Image histograms are an important tool for inspecting images. They allow you to spot Background and grey value range at a glance. Also clipping and Quantization Noise in image values can be spotted immediately [34].

II.8 types of image formats:

II.8 .1 Raster Image Files:

Raster images are constructed by a series of pixels, or individual blocks, to form an image. **JPEG**, **GIF**, and **PNG** are all raster image extensions. Every photo you find online or in print is a raster image. Pixels have a defined proportion based on their resolution (high or low), and when the pixels are stretched to fill space they were not originally intended to fit, they become distorted, resulting in blurry or unclear images. [35].

II.8 .2 Vector Image Files:

Vector images are far more flexible. They are constructed using proportional formulas rather than pixels. EPS, AI and PDF are perfect for creating graphics that require frequent resizing [36].

Chapter III: Hardware and software tools

Chapter III: Hardware and software tools

III.1 Introduction:

In this chapter we will talk about all the component parts of our project and all the programming software .

III.2 Hardware

III.2 .1The Raspberry pi

The Raspberry Pi is a series of single board computers. They are low cost high performance and the size of a credit card. The Raspberry Pi was developed in the UK by the Raspberry Pi Foundation. The Raspberry Pi Foundation's goal is to "advance the education of adults and children, particularly in the field of computers, computer science and related subjects."Many people have used Raspberry Pi's to make things like cameras, video game consoles, robots, web servers and media canterers [37] .

III.2.1.1 History:

In 2006, Eben Upton and her colleagues at Cambridge University noticed that Aspirants of a computer science degree had less experience than those in the 1990s. They attributed this decline, among other factors, to "the emergence of PCs in individuals and game consoles that replaced the Amiga, BBC Micro, Spectrum ZX and Commodore 64, machines with which the previous generation had learned to program". The computer has become so important to everyone in the household can also discourage young people from venturing into this field, because of the risk of damaging a important tool for the whole family. But recently, the technologies that have enabled mobile phones and tablets to become cheaper and more and more powerful, have launched the Raspberry Pi into the world of "ultra-economical-but-super-easy-to-use" boards [38].



Figure III. 1: Eben Upton (middle) and his colleagues.

III.2.1.2 Description of the raspberry pi board:

The Raspberry Pi 4 Model B (Pi4B) is the first of a new generation of Raspberry Pi computers supporting more RAM and with significantly enhanced CPU, GPU and I/O performance; all within a similar form factor, power envelope and cost as the previous generation Raspberry Pi 3B+.

The Pi4B is available with either 1, 2 and 4 Gigabytes of LPDDR4 SDRAM [39].

Chapter III: Hardware and software tools

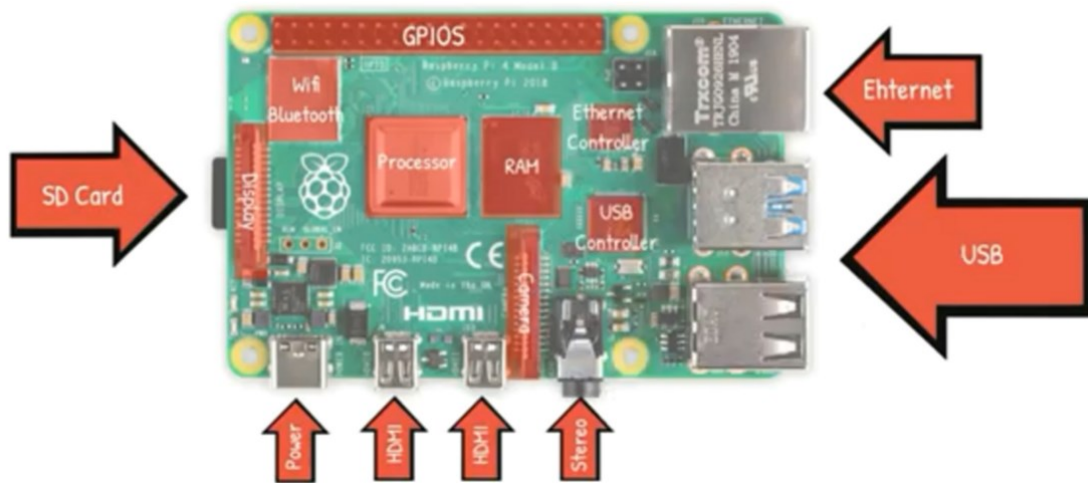


Figure III. 2: Raspberry pi4 connection architecture.

Features

- Quad core 64-bit ARM-Cortex A72 running at 1.5GHz
- 8 Gigabyte LPDDR4 RAM options
- H.265 (HEVC) hardware decode (up to 4Kp60)
- H.264 hardware decode (up to 1080p60)
- Video Core VI 3D Graphics
- Supports dual HDMI display output up to 4Kp60

Interfaces

- Bluetooth 5.0 with BLE
- 1x SD Card
- 2x micro-HDMI ports supporting dual displays up to 4Kp60 resolution
- 2x USB ports /2x USB3 ports
- 1x Gigabit Ethernet port
- 1x Raspberry Pi camera port (CSI)
- 40x user GPIO
- Power Requirements: The Pi4B requires a good quality USB C power supply capable of delivering 5V at 3A. [39]

Operating system:

The following operating systems are officially supported by all the models of Raspberry Pi (Raspbian, windows 10 IOT core , Kali linux...) ,We chose in this project the operating system (raspbian)

Raspbian : It is a free operating system based on Debian that is optimized for the Raspberry Pi hardware,Raspbian is more than a pure OS. It comes with over 35,000 packages and precompiled software for Raspberry Pi

Chapter III: Hardware and software tools

Peripherals

GPIO Interface

(general purpose input / output) supporting various interface options and GPIO pins can be switched (multiplexed) into various other modes backed by dedicated peripheral blocks such as I2C, UART and SPI [39].

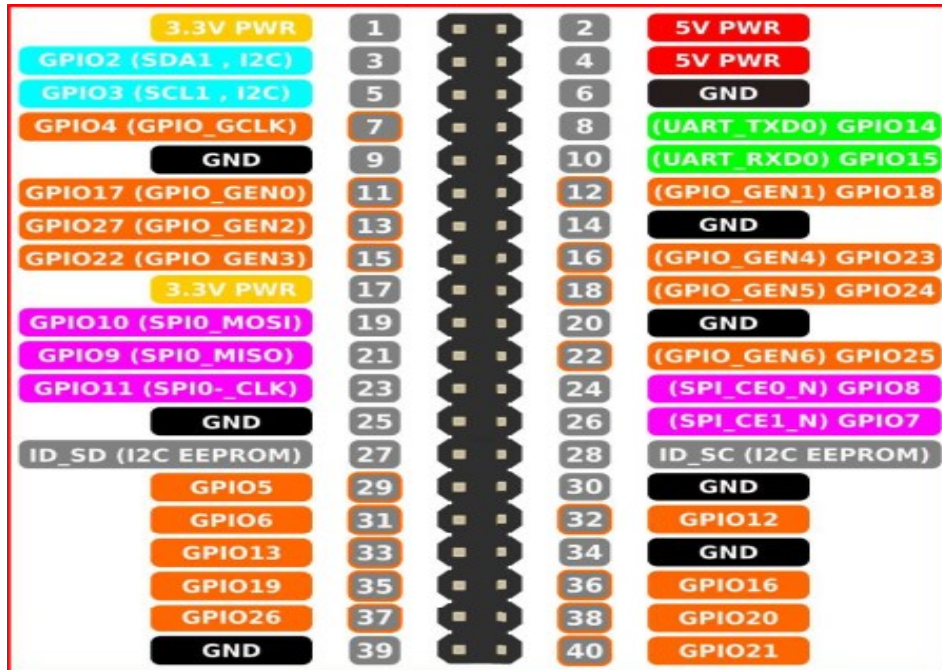


Figure III. 3: GPIO Connector Pin out.

USB

The Pi4B has 2x USB2 and 2x USB3 type-A sockets. Downstream USB current is limited to

Approximately 1.1A in aggregate over the four sockets



Figure III. 4: The raspberry pi USB interfaces.

HDMI

The Pi4B has 2x micro-HDMI ports, both of which support CEC and HDMI 2.0 with Resolutions up to 4Kp60

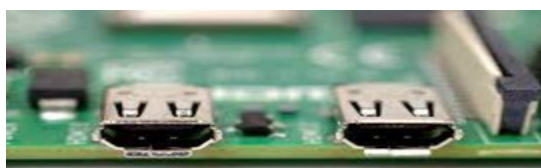


Figure III. 5: The HDMI interface.

Chapter III: Hardware and software tools

Camera and display interfaces

The Pi4B has 1x Raspberry Pi 2-lane MIPI CSI Camera and 1x Raspberry Pi 2-lane MIPI DSI Display connector, These connectors are backwards compatible with legacy Raspberry Pi boards, and support all of the available Raspberry Pi camera and display

Peripherals

➤ Audio and Composite (TV Out)

The Pi4B supports near CD quality analogue audio output and composite TV



Figure III. 6: Audio and Composite interface.

Ethernet/RJ45

The raspberry Pi has an Ethernet socket



Figure III. 7: Rj45.

III.2 .2 Materiel used:

➤ Dc motor

The small DC motor runs off any battery or solar cell.



Figure III. 8: The dc motor.

➤ Camera

In this project we have to use high image capturing digital camera to get the clear images of vehicles. It connects directly to the Camera Serial Interface (CSI) port in the Raspberry Pi 4 For practical purpose, we have used following camera.

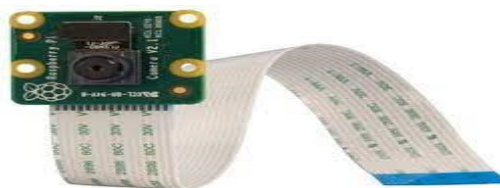


Figure III. 9: Raspberry pi camera module.

Chapter III: Hardware and software tools

➤ H-Bridge Motor Driver

The L298N is a dual H-Bridge motor driver which allows speed and direction control of two DC motors at the same time. The module can drive DC motors that have voltages between 5 and 35V, with a peak current up to 2A [40].

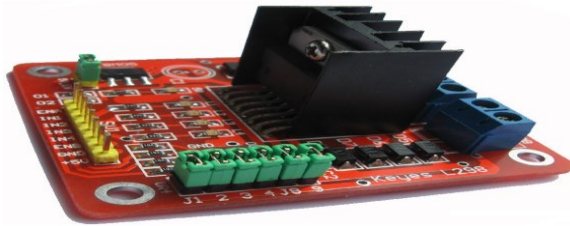


Figure III. 10: The L298N module.

III.3 Software

III.3.1 python:

Python: Dynamic programming language which supports several different programming paradigms and Open source general-purpose language:

- Procedural programming
- Object oriented programming
- Functional programming



Figure III. 11: Logo of python.

Is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

- Web development (server-side).
- Software development.
- Mathematics.
- System scripting.

We use python because:

- Big community
- Works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).
- Can be treated in a procedural way, an object-oriented way or a functional way
- Contains powerful libraries in artificial intelligence like: Opencv, tensorflow, pytorch, etc [41].

Chapter III: Hardware and software tools

III.3.2 PyCharm IDE:

PyCharm is a hybrid-platform developed by JetBrains as an IDE (integrated development environment) for Python. It is commonly used for Python application development. Some of the unicorn organizations such as Twitter, Facebook, Amazon, and Pinterest use PyCharm as their Python IDE .

An IDE is a software application that provides comprehensive facilities to computer programmers for software development. An IDE normally consists of at least a source code editor, build automation tools and a debugger [42].

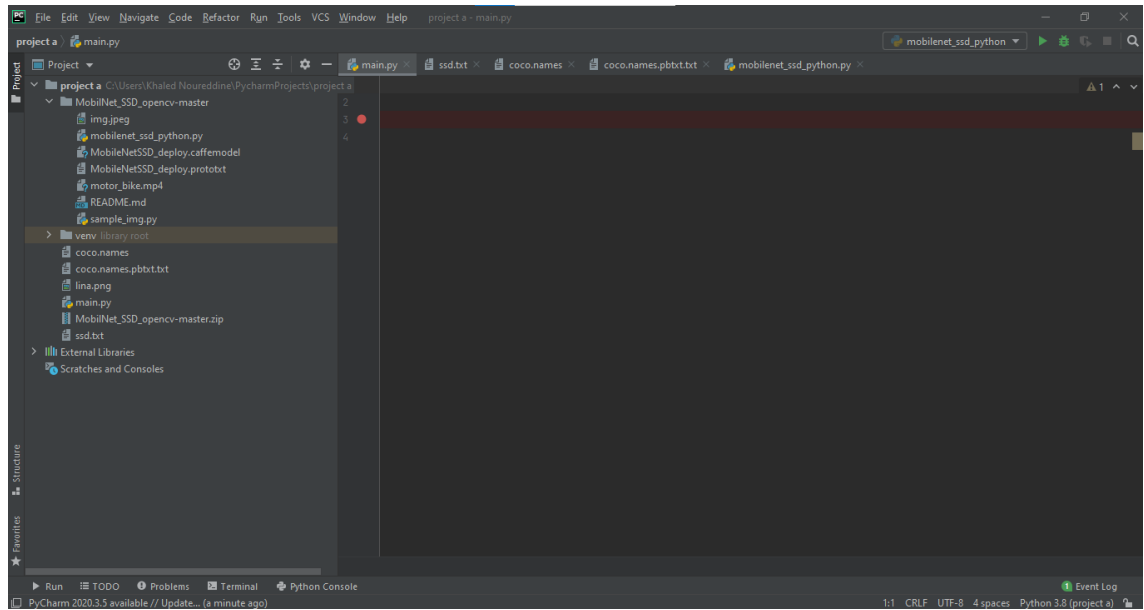


Figure III. 12: Pycharm interface.

III.3.2 libraries:

III.3.2.1 OpenCV

OpenCV was started at Intel in 1999 by Gary Bradsky, and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel's Russian software OpenCV team [43].

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms [44].



Figure III. 13: Open CV.

Applications of OpenCV :

Convert RGB to grayscale image:

if the color has no significance in the images to classify then it's better to go for grey scale images to avoid false classification and complexities.

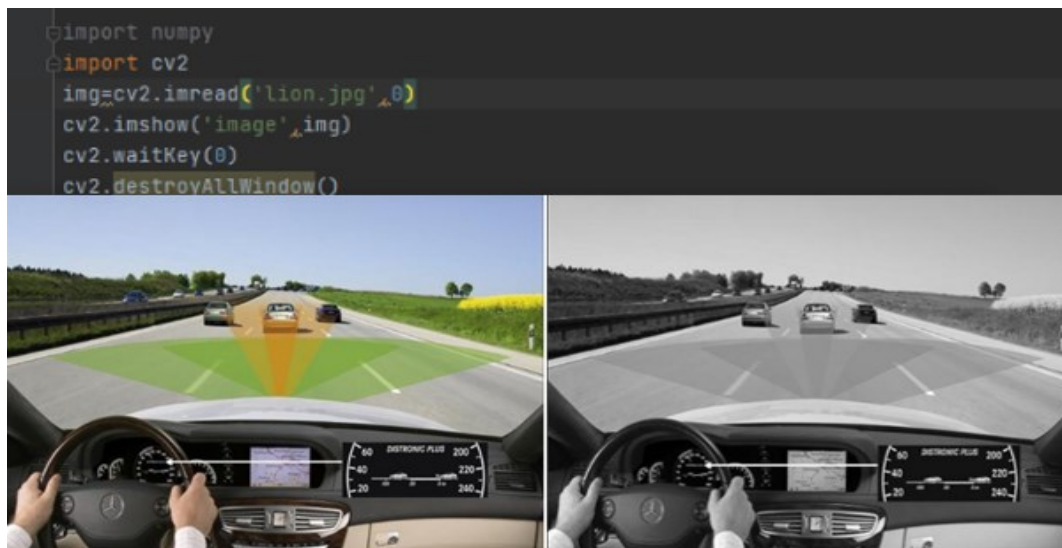


Figure III. 14: Converting RGB to grayscale.

Draw lines and shapes:

In open cv we can draw shapes with a simple and easy functions.

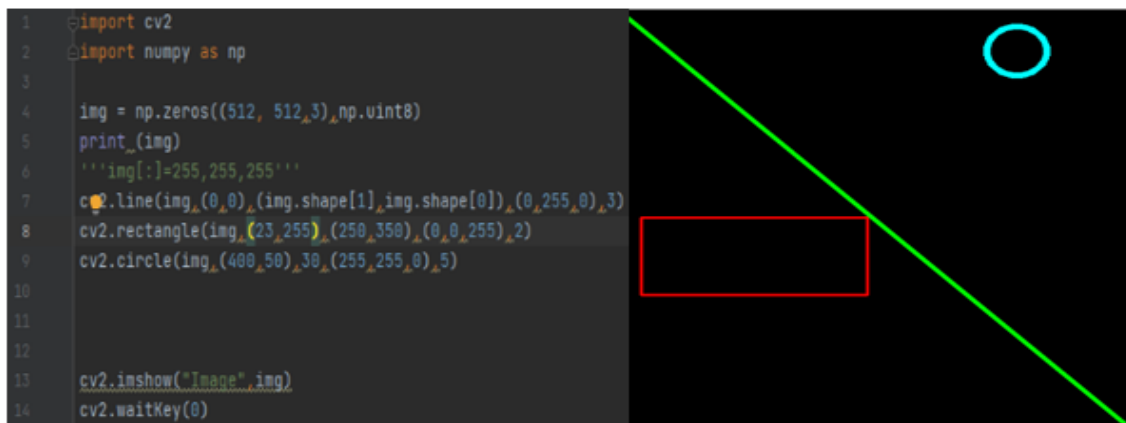


Figure III. 15: Drawing shapes.

Chapter III: Hardware and software tools

Crop and Resize Images: This function is used to delete unimportant details in the image.

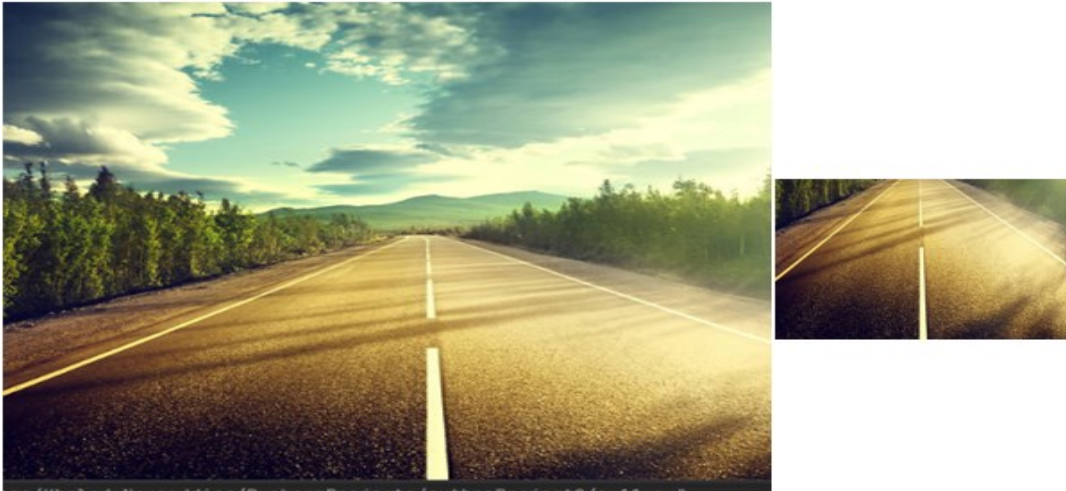


Figure III. 16: Cropping and resizing images.

III.3.2.2 Tensor Flow:

Tensor Flow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

The important features of Tensor Flow:

- It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
- It includes a programming support of deep neural networks and machine learning techniques.
- It includes a high scalable feature of computation with various data sets.
- Tensor Flow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used. [45]

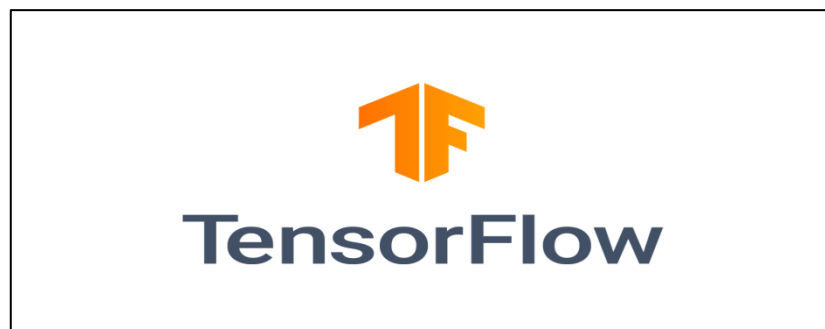


Figure III. 17: TensorFlow .

Chapter III: Hardware and software tools

Tensor flow Architecture:

Tensor Flow Serving is a flexible, high-performance serving system for machine learning models, designed for production environments. Tensor Flow Serving makes it easy to deploy new algorithms and experiments, while keeping the same server architecture and APIs. Tensor Flow Serving provides out of the box integration with Tensor Flow models, but can be easily extended to serve other types of models. [46]

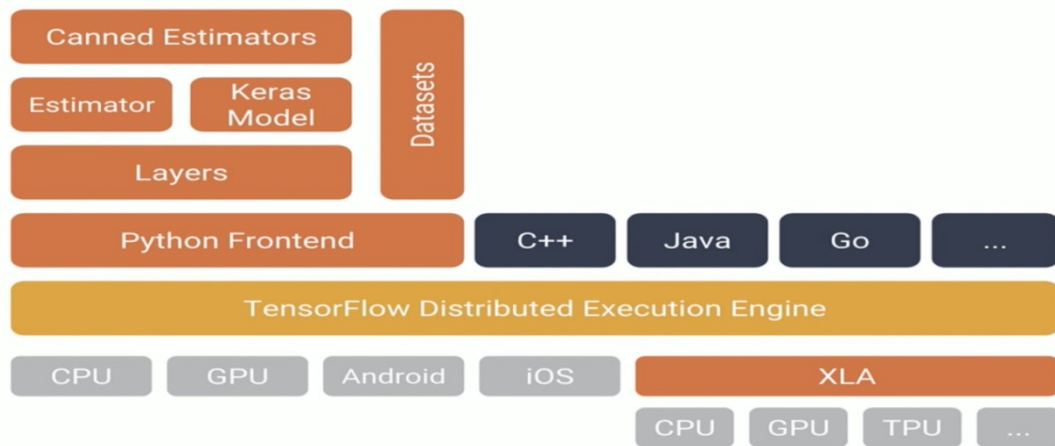


Figure III. 18: Tensor flow Architecture.

III.3.2.3 keras :

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research. [47]



Figure III. 19: keras.

Chapter IV: Implementation and programming

Chapter IV: Hardware and software tools

IV.1 Introduction:

In this chapter we will describe the components necessary to build the self driving car and running all of the software on this car, and we will see some applications on it.

IV.2 Hardware Components Connection:

IV.2 .1 List of Hardware:

- Pre-built Robot Car Chassis (2Wheels).
- Raspberry pi Model B+ (With Wi-Fi Module).
- Power Bank.
- L298N H bridge Motor driver.
- Jumper Wires to connect individual components.
- Pi Camera.

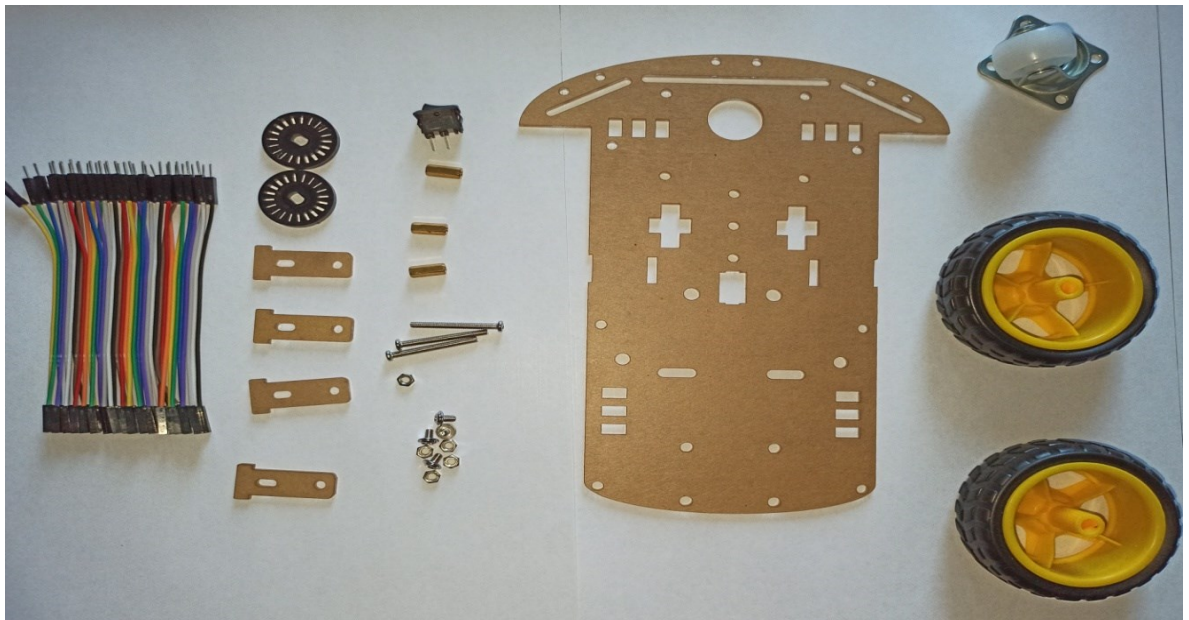


Figure IV. 1:The hardware components.

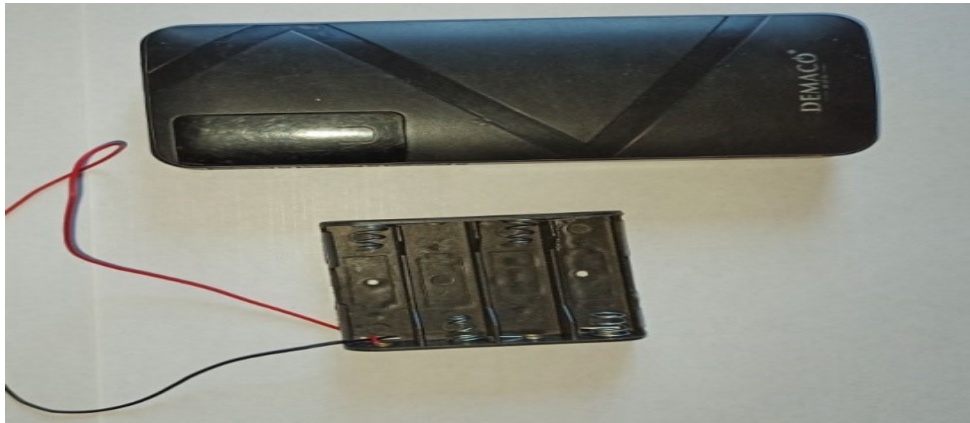


Figure IV. 2:The power.

IV.2 .2 Assembly:

The 2 wheels of the chassis are connected separately to 2 DC motors. Then another serial connection is made between the adjacent wheels so that they move in the same direction. These motors are connected to the L298N Dual H bridge motor driver which helps in regulating the speed. Then connecting the raspberry pi to the camera.



Figure IV. 3: The hardware components connection

IV.3 Applications:

In this chapter we will see three difference applications (Lane detection, object detection, traffic signs) and we will explain how it's work.

IV.3 .1 Applications 1: Self Driving Lane Detection car using Python and Opencv on Raspberry Pi.

IV.3 .1 .1 Introduction:

Driver Assistant System is designed to assist drivers in the perception of any dangerous situations before, to avoid accidents after sensing and understanding the environment around itself. To date there have been numerous studies into the

Chapter IV: Hardware and software tools

recognition. Traffic accidents have become one of the most serious problems. The reason is that most accidents happen due to negligence of the driver. Rash and negligent driving could push other drivers and passengers in danger on the roads. More and more accidents can be avoided if such dangerous driving condition is detected early and warned to other drivers. Most of the roads, cameras and speed sensors are used for monitoring and identifying drivers who exceed the permissible speed limit on roads and motorways. This simplistic approach, and there are no restrictions.

IV.3 .1 .2 Algorithm:

The block diagram of a proposed lane detection system on Raspberry Pi is shown in the figure **FigureIV.4**.

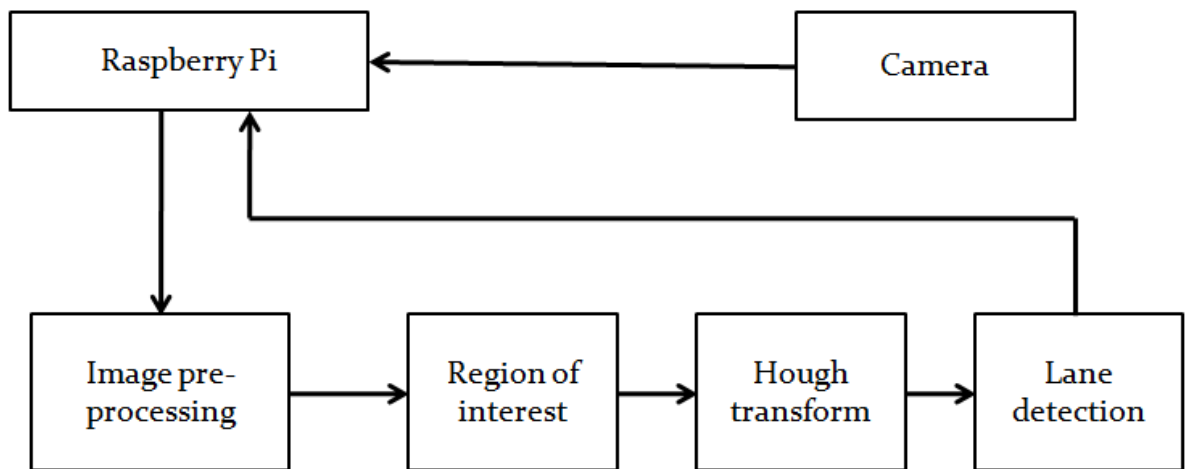


Figure IV. 4: Block diagram of lane detection System.

Each blocks of a block diagram are explained one by one below:

- Capture input image: Hardware like Camera is used to take input image.
- Image Preprocessing: To enhance the quality of image, we need to preprocess it. The processes like noise reduction, edge detection contrast and color management are applied.
- . Region of interest: In determining the computational complexity of lane identification. Here only the selected are as is detected or taken for the next level of processing. These selected images are then used for lane detection using a proposed algorithm. The selection I reduce the processing time of the frames.
- . Hough Transform: The Hough Transform is implemented on images after the canny edges detection has taken place so as to obtain the image pixels that are desired ones. So here in this system to detect the lanes marking from the image data, Hough Transform is used.
- . Lane Detection: Here, the Lane will be marked with a separate color. Two important algorithms Canny Edge Detection and Hough Transform are used to implement Lane Detection System.

Chapter IV: Hardware and software tools

IV.3 .1 .3 Lane detection:

In this project we are having a main module that will first of all connect to webcam module. The images captured in the webcam module are send to main module and are used by lane module to detect the lanes from the image and sends us back the curve values on how much we need to turn the wheels of the robot . We also have a utility file that is linked to lane module. The reason for this is we don't want to write all the code in one module if it is too long. We will write all the supporting functions in the utility file and we can relate to lane module. Once the curve is reached the main module, we can send this curve to motor module which will turn the motors based on the speed we have provided or based on the turn we have provided or received from the lane module.

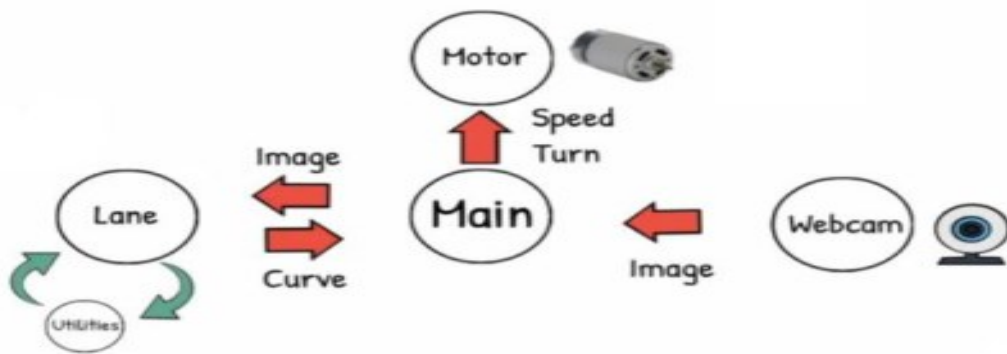


Figure IV. 5: The modules of lane detection.

We are going to use pixel summation method. The idea is that we are summing up the pixels in a column. Black is basically 0 and white is 255. Now because we are using 8 bit integers, we have 2 power 8 values that are from 0 to 256 values. We gave white as 255 and black as 0.

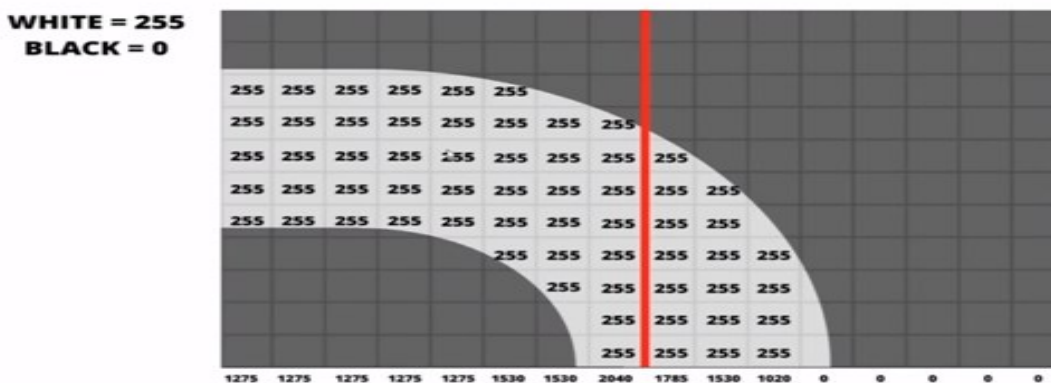


Figure IV. 6: Pixel summation.

so with summing up the pixels in the column and with little bit of math we can not only determine whether we have to turn left or right or go straight but also how much we have to turn left or right . So that is the value of the curve. This is not a complicated

Chapter IV: Hardware and software tools

process and we are not using fancy algorithms. It will be a easy process and we will go step by step and we will see the output of each steps.

Even though we are going to run our code in raspberry pi we are going to write our code in pc and the reason for that is it makes the coding process much easier and its much faster to debug and once the module is prepared we can add it to our existing module and run it .

a) Step 01: Thresholding.

Now the idea here is to get the path using either Color or Edge Detection. In our case we are using regular A4 White paper as path, therefore we can simply use the color detection to find the path. We will write the Thresholding function in the Utilities file.

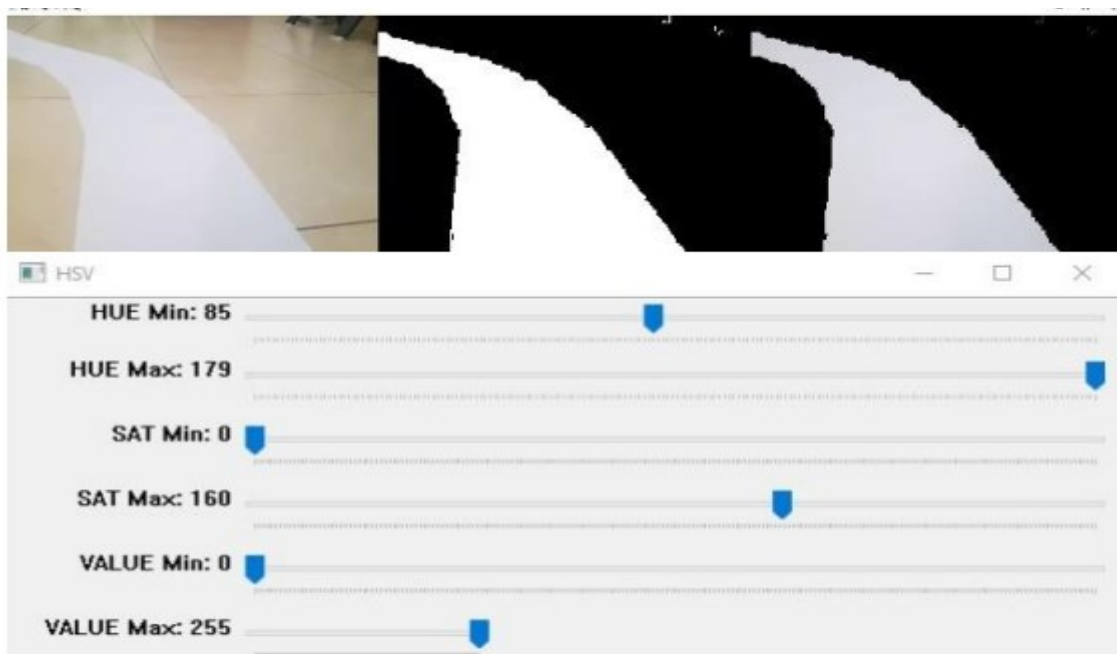


Figure IV. 7: Thresholding.

b) Step 02: warping.

We don't want to process the whole image because we just want to know the curve on the path right now and not a few seconds ahead. So we can simply crop our image, but this is not enough since we want to look at the road as if we were watching from the top. This is known as a bird eye view and it is important because it will allow us to easily find the curve. To warp the image we need to define the initial points. These points we can determine manually. So to make this process easier we could use track bars to experiment with different values. The idea is to get a rectangle shape when the road is straight.



Figure IV. 8: Warping.

c) Step 03: Histogram

Now comes the most important part, finding the curve in our path. To do this we will use the summation of pixels. Given that our warped image is now binary i.e. it has either black or white pixels, we can sum the pixel values in the y direction. All the white pixels with 255 values and all the black with 0, now if we sum the pixels in first column it yields $255+255+255+255+255 = 1275$ We apply this method to each of the columns. In our original Image we have 480 pixels in the width. Therefore we will have 480 values. After summation we can look at how many values are above a certain threshold.

The figure below shows the 3 cases where this method would work. We can clearly see that when the curve is right the numbers of pixels on the right hand side are more than the left hand .and when the curve is left the numbers of pixels on the left hand side are more than the right hand and when it's straight the numbers of pixels are approximately same on both sides.

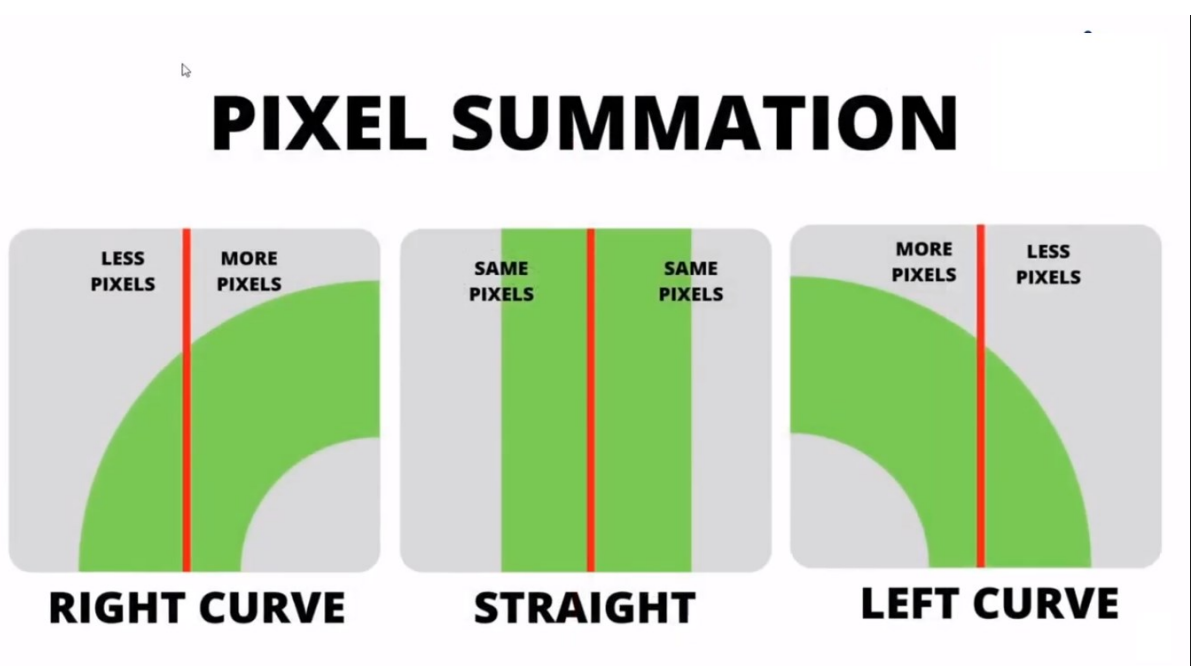


Figure IV. 9: The curve direction.

d) Step 04: Averaging

Once we have the curve value we will append it in a list so that we can average this value. Averaging will allow smooth motion and will avoid any dramatic movements

e) Step 05: display

Now we can add options to display the final result. We will add an input argument to our main 'getLaneCurve' function so that we can have the flexibility of turning it on and off, since raspberry pi would run at very slow speeds if we display and run at the same time.

➤ The coding:

```
#####step 1
imgThres = utlis.thresholding(img)
#####step 2
hI, wI, c = img.shape
points = utlis.valTrackbars()
imgWarp = utlis.warpImg(imgThres, points, wT, hT)
imgWarpPoints = utlis.drawPoints(imgCopy, points)
##### STEP 3
middlePoint, imgHist = utlis.getHistogram(imgWarp, display= True, minPer=0.5, region=4)
curveAveragePoint, imgHist = utlis.getHistogram(imgWarp, display=True, minPer=0.9)
curveRaw = curveAveragePoint - middlePoint
#####step 4
curveList.append(curveRaw)
if len(curveList) > avgVal:
    curveList.pop(0)
curve = int(sum(curveList) / len(curveList))

##### STEP 5
if display != 0:
    imgInvWarp = utlis.warpImg(imgWarp, points, wT, hT, inv=True)
    imgInvWarp = cv2.cvtColor(imgInvWarp, cv2.COLOR_GRAY2BGR)
    imgInvWarp[0:hT // 3, 0:wT] = 0, 0, 0
    imgLaneColor = np.zeros_like(img)
    imgLaneColor[:] = 0, 255, 0
    imgLaneColor = cv2.bitwise_and(imgInvWarp, imgLaneColor)
    imgResult = cv2.addWeighted(imgResult, 1, imgLaneColor, 1, 0)
    midY = 450
```

Figure IV. 10: The lane detection module.

```
cap = cv2.VideoCapture(0)
def getImg(display= False, size=[480,240]):
    _, img = cap.read()
    img = cv2.resize(img, (size[0], size[1]))
    if display:
        cv2.imshow('IMG', img)
    return img
```

Figure IV. 11: The webcam module.


```
motor.move(0.5,0,2)
motor.stop(2)
motor.move(-0.5,0,2)
motor.stop(2)
```

Figure IV. 12: The motor module.

IV.3 .2 Applications 02: Object recognition

IV.3 .2 .1 Introduction:

Computer vision is a field of computer science which enables machines not only to see but also to process and analyze digital images and videos. One big application area of computer vision is object detection, the capability of a computer to locate and identify objects in an image. With the recent advances in deep learning and convolution neural networks (CNN) computers can from large image datasets learn to recognize and track objects seen in images and videos with great accuracy.

This thesis studies the possibility of implementing an object detector on a single board computer, the Raspberry Pi4B, capable of maintaining real-time frame rate while keeping high precision [50]

IV.3 .2 .2 Object recognition

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs.

Image classification involves predicting the class of one object in an image. Object localization refers to identifying the location of one or more objects in an image and drawing bounding box around their extent.

Object detection combines these two tasks and localizes and classifies one or more objects in an image

We can distinguish between these three computer vision tasks:

- **Image Classification:** Predict the type or class of an object in an image.
Input: An image with a single object, such as a photograph.
Output: A class label (e.g. one or more integers that are mapped to class labels).
- **Object Localization:** Locate the presence of objects in an image and indicate their location with a bounding box.
Input: An image with one or more objects, such as a photograph.
Output: One or more bounding boxes (e.g. defined by a point, width, and height).
- **Object Detection:** Locate the presence of objects with a bounding box and types or classes of the located objects in an image.
Input: An image with one or more objects, such as a photograph.
Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box from this breakdown, we can see that object recognition refers to a suite of challenging computer vision tasks [51]

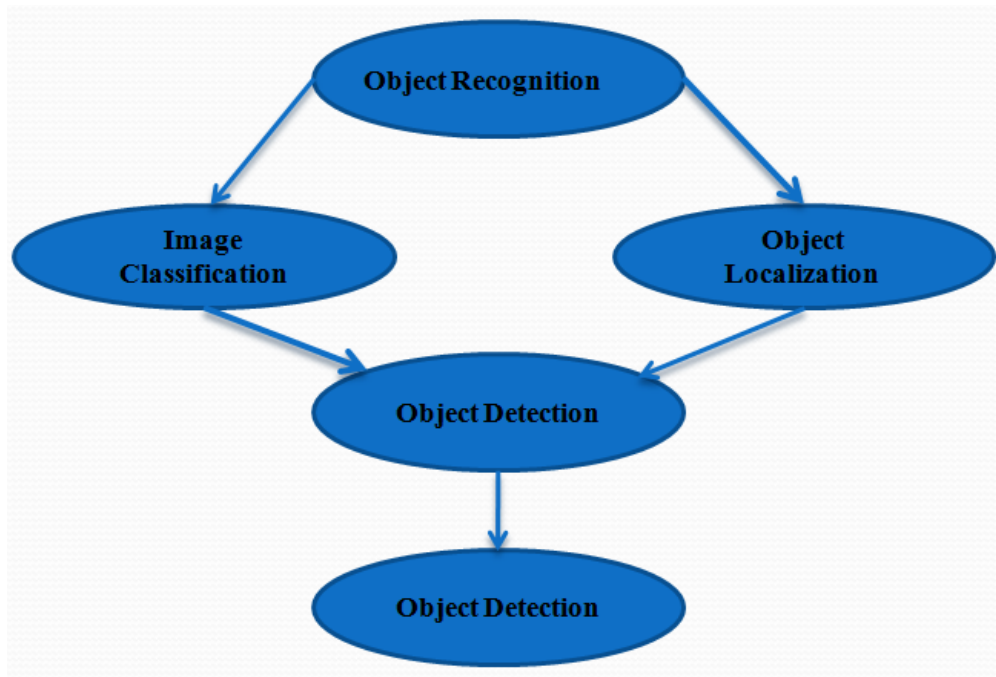


Figure IV. 13: Overview of object Recognition computer vision tasks.

Object recognition using neuron network in deep learning:

➤ **Convolution Neural Networks (CNN):**

Convolution Neural Networks as they are made up of neurons that have learnable weights and biases are similar to ordinary Neural Networks. Each neuron receives some inputs, performs a dot product and optionally follows it with a non-linearity activation function.

The overall functionality of the network is like having an image on one end and class as an output on the other end in simple words, images are sent to the input side and the output side to classify them into classes based on a probability score.[49]

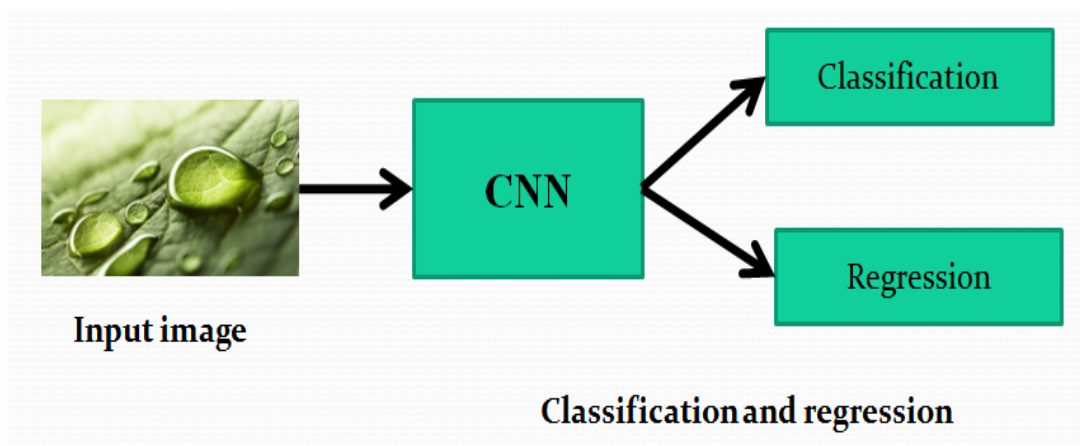


Figure IV. 14: Model Detector.

Chapter IV: Hardware and software tools

➤ SSD mobile net:

SSD: is a single-shot detector. It has no delegated region proposal network and predicts the boundary boxes and the classes directly from feature maps in one single pass. To improve accuracy, SSD introduces: small convolution filters to predict object classes and offsets to default boundary boxes.

Mobile Net: is a light-weight deep neural network architecture designed for mobiles and embedded vision applications. Single Shot object detection or SSD takes one single shot to detect multiple objects within the image [50]

The coding:

First, we import the necessary packages so that our model can function - DNN module is included in cv2.

```
1  #Import the necessary libraries
2  import numpy as np
3  import argparse
4  import cv2
```

Figure IV. 15: Import the necessary libraries.

Next, we analyze our command line arguments

```
7  parser = argparse.ArgumentParser(
8      description='Script to run MobileNet-SSD object detection network ')
9  parser.add_argument("--video", help="path to video file. If empty, camera's stream will be used")
10 parser.add_argument("--prototxt", default="MobileNetSSD_deploy.prototxt",
11                    help='Path to text network file: '
12                        'MobileNetSSD_deploy.prototxt for Caffe model or ')
13 parser.add_argument("--weights", default="MobileNetSSD_deploy.caffemodel",
14                    help='Path to weights: '
15                        'MobileNetSSD_deploy.caffemodel for Caffe model or ')
16 parser.add_argument("--thr", default=0.2, type=float, help="confidence threshold to filter out weak detections")
17 args = parser.parse_args()
```

Figure IV. 16: Construct the argument parse.

Then, let's initialize the list of class labels that Mobile Net SSD has been trained for detect.

```
22 classNames = {0: 'background',
23              1: 'aeroplane', 2: 'bicycle', 3: 'bird', 4: 'boat',
24              5: 'bottle', 6: 'bus', 7: 'car', 8: 'cat', 9: 'chair',
25              10: 'cow', 11: 'diningtable', 12: 'dog', 13: 'horse',
26              14: 'motorbike', 15: 'person', 16: 'pottedplant',
27              17: 'sheep', 18: 'sofa', 19: 'train', 20: 'phone' }
```

Figure IV. 17: Labels of network.

Chapter IV: Hardware and software tools

Now we need to load our model. To do this, we use the function: OpenCV readNetFrom

```
36 net = cv2.dnn.readNetFromCaffe(args.prototxt, args.weights)
```

Figure IV. 18: Load the Caffe model.

Next, we loaded our query image and prepare our blob, which we will transmit through the neural network and get the detections and predictions. The "blob" is a preprocessing step (it's to give the network an image in the right format in order to that he accepts it).

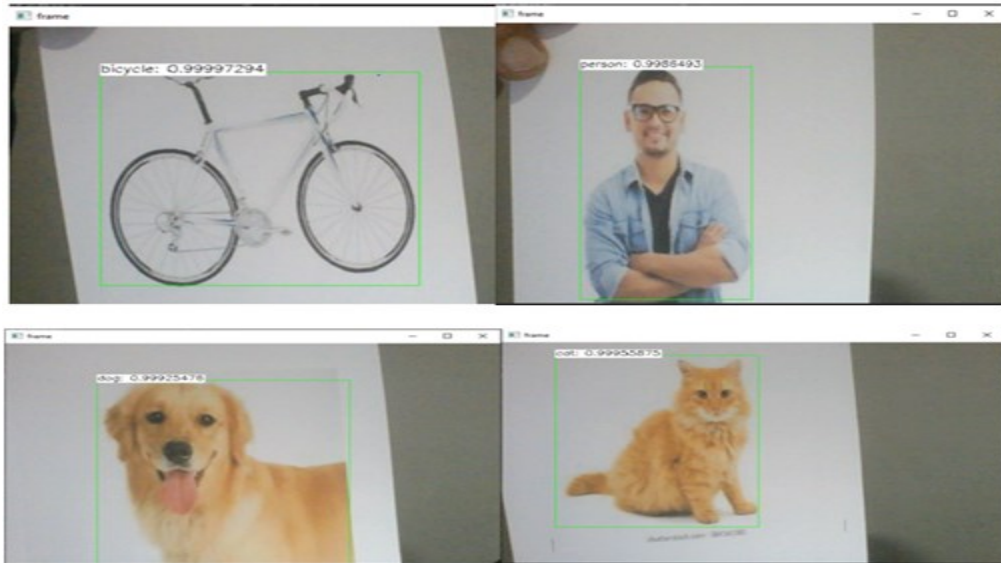


Figure IV. 19: The Output.

IV.3 .3 Application 03: traffic sings

IV.3.3.1 Introduction:

The goal of this application is to train a model that can recognize traffic sings, on this basis, it's can make decisions related to respecting the laws and Keeping users safe.

In this application we use deep learning to train the car to classify traffic signs with 95% accuracy.

IV.3.3.2 Traffic sign detection and recognition using OpenCV and Python:

This application detects and recognizes traffic signs in video sequences recorded by an onboard vehicle camera. Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn a driver, and command or prohibit certain actions.

A fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver and significantly increase driving safety and comfort. Automatic recognition of traffic signs is also important for automated intelligent driving vehicle or driver assistance systems. This paper presents a study to recognize traffic sign patterns using OpenCV technique.

Chapter IV: Hardware and software tools

The images are extracted, detected and recognized by preprocessing with several image processing techniques, such as, threshold techniques, Gaussian filter, canny edge detection, Then the stages are performed to detect and recognize the traffic sign patterns. The system is trained and validated to find the best network architecture.

The experimental results show the highly accurate classifications of traffic sign patterns with complex background images and the computational cost of the proposed method [51].

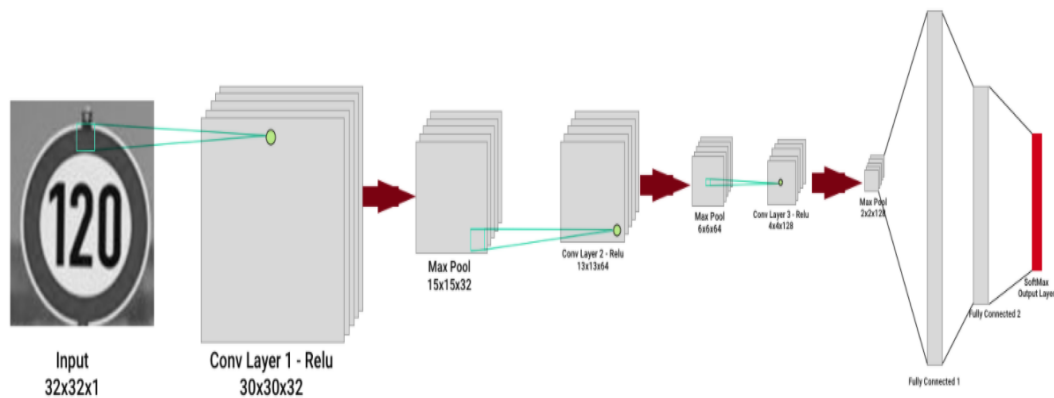


Figure IV. 20: CNN in traffic signs.

Dataset:

The dataset used for this application it's very famous consists of 43 different traffic signs classes have about 35000 pictures.

```
##### Parameters #####
path = "myData" # folder with all the class folders
labelFile = 'labels.csv' # file with all names of classes
batch_size_val = 50 # how many to process together
steps_per_epoch_val = 2000
epochs_val = 15
imageDimensions = (32, 32, 3)
testRatio = 0.2 # if 1000 images split will 200 for testing
validationRatio = 0.2 # if 1000 images 20% of remaining 800 will be 160 for validation
#####
```

Figure IV. 21: The data loading.

Training the model:

➤ Step one:

We put all the parameters for training.

Chapter IV: Hardware and software tools

```
##### Parameters #####  
path = "myData" # folder with all the class folders  
labelFile = 'labels.csv' # file with all names of classes  
batch_size_val = 50 # how many to process together  
steps_per_epoch_val = 2000  
epochs_val = 15  
imageDimensions = (32, 32, 3)  
testRatio = 0.2 # if 1000 images split will 200 for testing  
validationRatio = 0.2 # if 1000 images 20% of remaining 800 will be 160 for validation  
#####
```

Figure IV. 22: Parameters.

➤ Step two:

We have 30 epochs each one of them takes about two minutes to finish the training.

```
Run: 1 x  
Epoch 27/30  
400/400 [=====] - 67s 167ms/step - loss: 0.1494 - accuracy: 0.9520 - val_loss: 0.0191 - val_accuracy: 0.9944  
Epoch 28/30  
400/400 [=====] - 66s 164ms/step - loss: 0.1554 - accuracy: 0.9527 - val_loss: 0.0204 - val_accuracy: 0.9952  
Epoch 29/30  
400/400 [=====] - 66s 166ms/step - loss: 0.1272 - accuracy: 0.9624 - val_loss: 0.0240 - val_accuracy: 0.9932  
Epoch 30/30  
400/400 [=====] - 67s 166ms/step - loss: 0.1304 - accuracy: 0.9592 - val_loss: 0.0155 - val_accuracy: 0.9957
```

Figure IV. 23 : Epochs of validation.

➤ Step three:

And finally we display the Accuracy and the Loss after this we will save the modal in a pickel file.

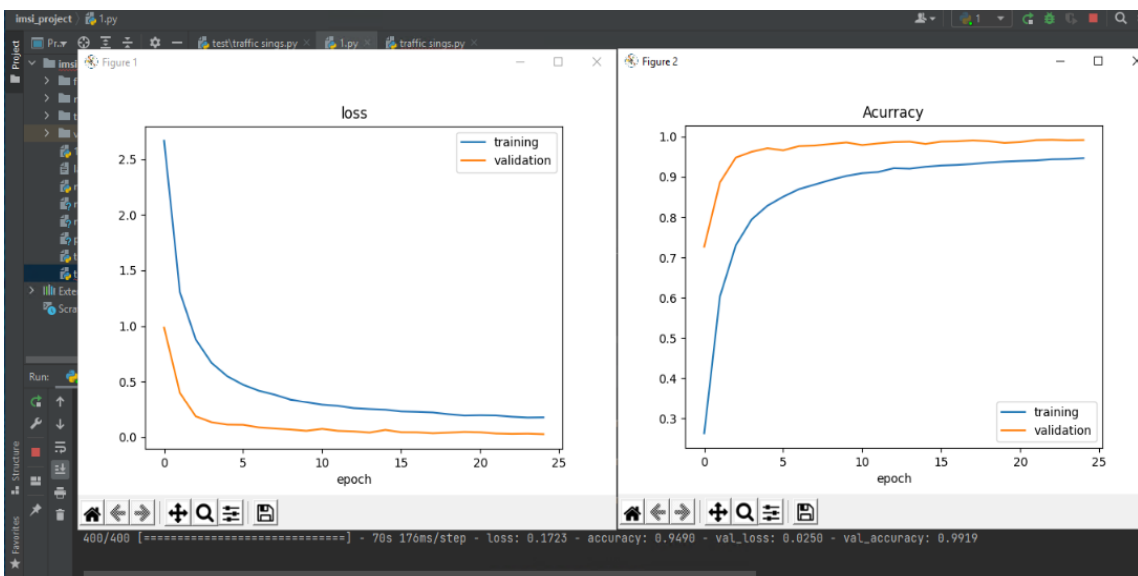


Figure IV. 24 : Loss and accuracy curve.



Figure IV. 25 : The traffic signs output.

IV.4 The Hardware Implementation:

In this part we will show how to implement this code in raspberry pi, and how can we run it, and try to find the best parameters to work properly.

➤ **Step 1: Install library:**

We need to install these libraries (OpenCV 4.4.0, Numpy1.12, Rpi.GPIO, Pandas)

➤ **Step 2: Uploading the code in raspberry pi:**

In this part we Copy the file that contains the entire module (motor, main Robot, webcam, lane detection, utilis) and place it in the same folder in raspberry pi.

Chapter IV: Hardware and software tools

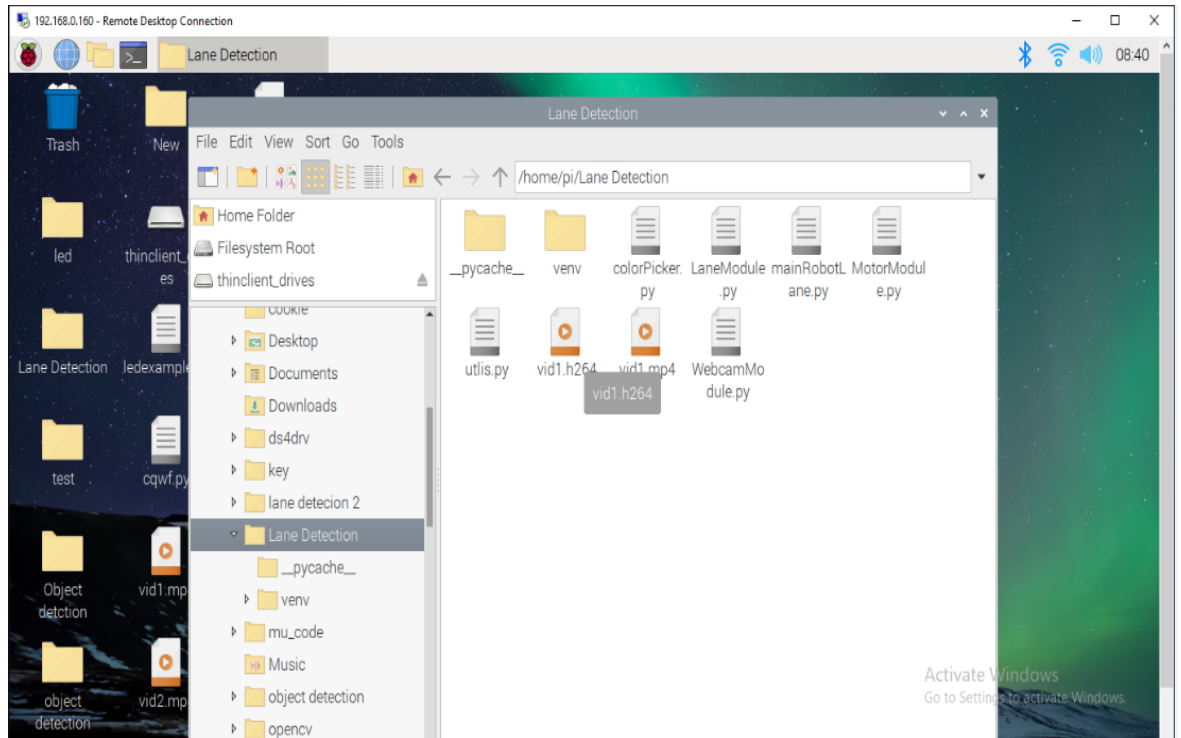


Figure IV. 26: Uploading the code .

Then open all the modules in python environment Thonny IDE or mu IDE.

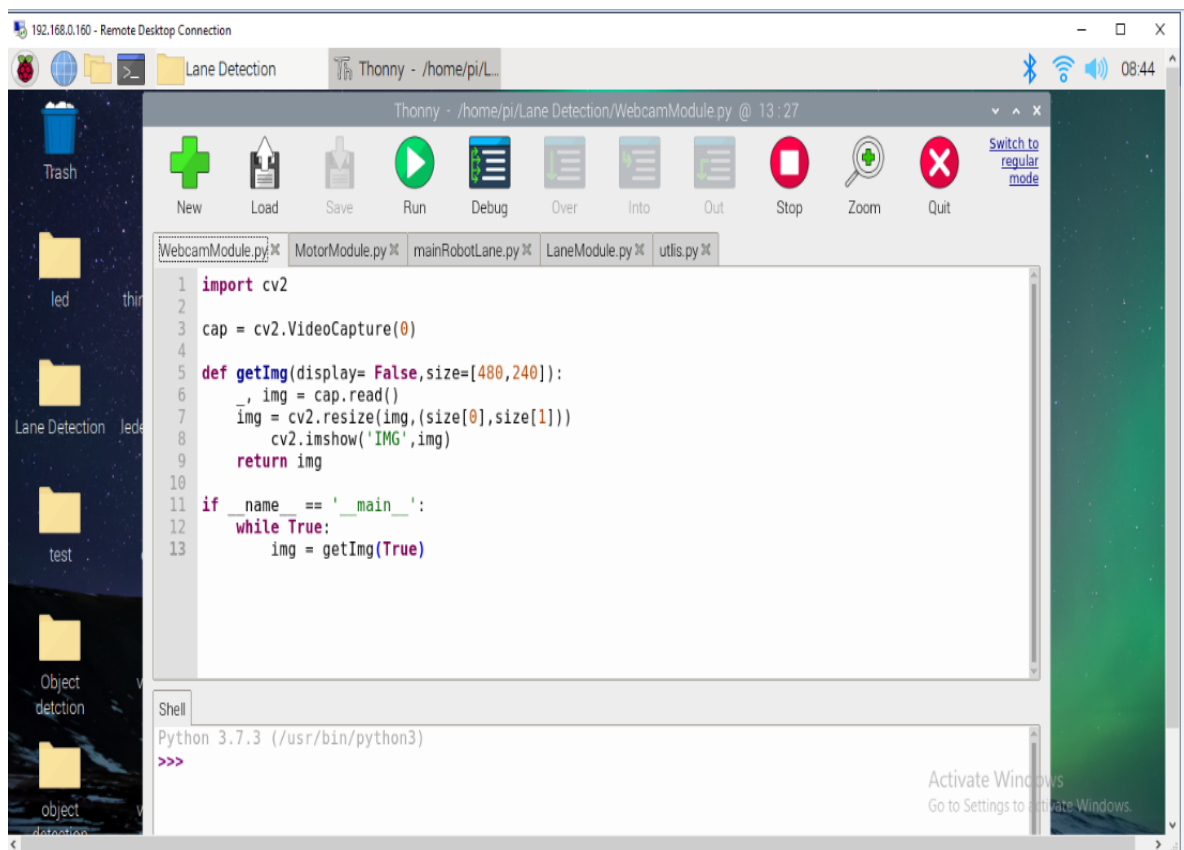


Figure IV. 27: Thonny environment.

Chapter IV: Hardware and software tools

➤ Step 3: execute the code:

Execute the modules (motor, utilis, and lane detection) and check that there are no errors.

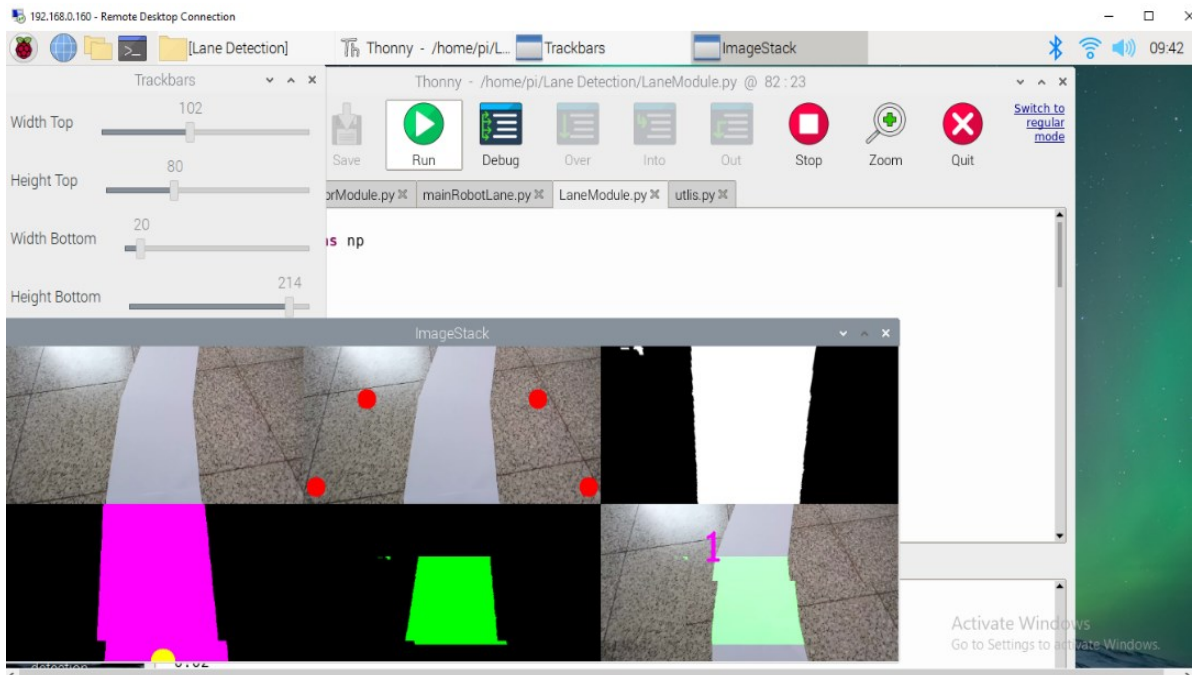


Figure IV. 28 : Image stack.

Lastly run the main robot and observing the car can find and follow the curve .

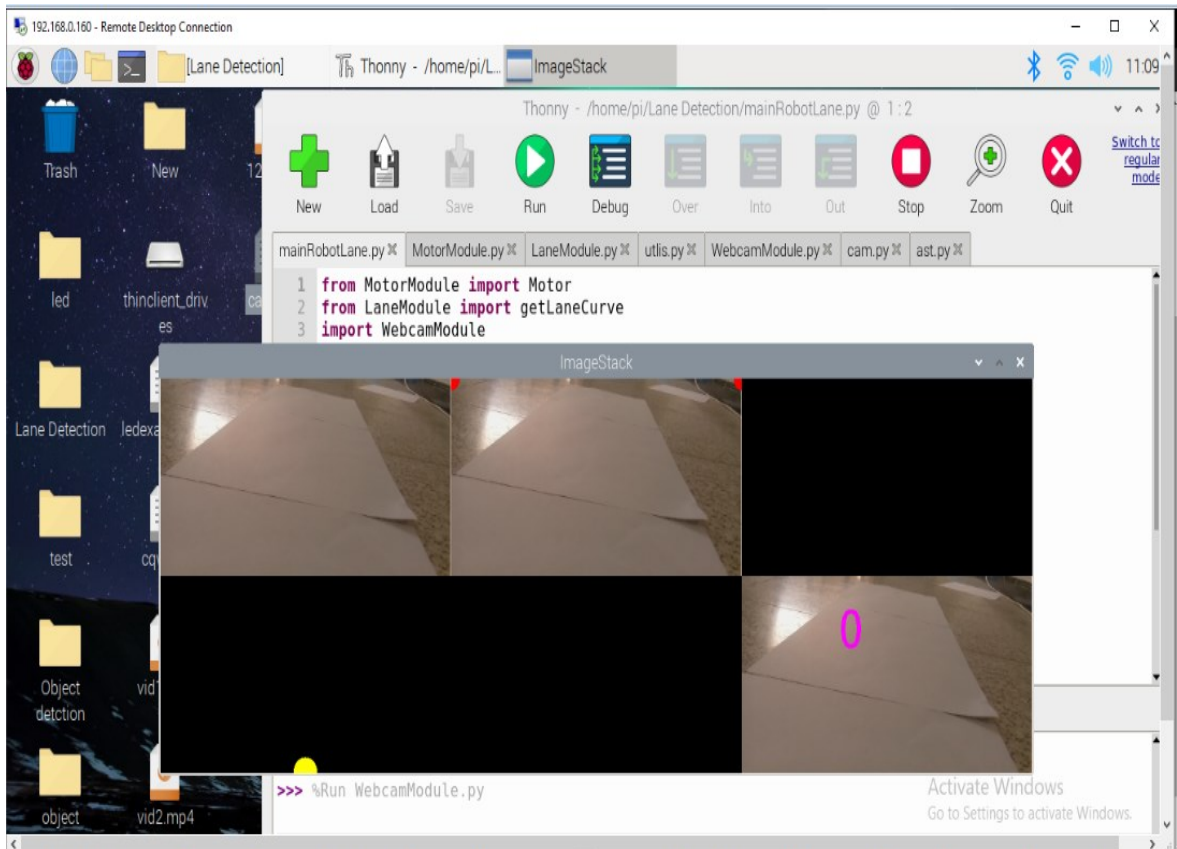


Figure IV. 29: Car point of view.

Chapter IV: Hardware and software tools

➤ **Problems and encountered :**

Although the software part functioned perfectly, the hardware part didn't function as planned; and the big hardware problems that we encountered are :

- Weak camera sensitivity, so we couldn't detect the curve lane properly, which held us back significantly and we couldn't collect our data (pictures of the lane)
- the rechargeable batteries life was short.
- we had to use a power bank and it was heavy.

Other problems that we faced:

- We had no prior knowledge of this vast field
- We took a very long time to find the proper tools
- The 3D printer took a long time to print the prototypes, so we wasted several days

GENERAL CONCLUSION

GENERAL CONCLUSION

➤ **Conclusion and Future Scope:**

To summarize this thesis, the system aims to improve the safety of self-driving technology using new techniques. It is mainly about implementing a self-driving car that can make its own decisions accurately.

The system is considered to be a prototype car that contains raspberry pi 4 and camera to perceive the surrounding environment. The methodologies used in the system are lane detection using image filtering methods along with Hough transform feature extraction.

The result of lane detection is that the car moves accurately in its path according to the signals that are being sent to the motors from the road lane detection algorithm.

The goal of this project was the car can moves and detects the road with a high accuracy and is able to make the right decisions moving forward, slowing down, or stopping based on the traffic signs

During the development process, we reached our goal which was to apply the idea on a small car prototype. So we aim that our future work will consider applying this system in a real car.

Bibliography

Bibliography

- [1] : <https://www.investopedia.com/terms/a/artificial-intelligence-ai.asp>
- [2] : Phil Kim,2017, MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence,p8
- [3] : ...<https://siecledigital.fr/22/03/2020>.
- [4] : ...<https://archives.rsln.fr/23/03/2020>.
- [5] : Aurélien Géron , June 2019 , Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Concepts, Tools, and Techniques to Build Intelligent Systems, p04.
- [6]: Phil Kim,2017, MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence, p03.
- [7]: Géron , June 2019 , Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Concepts, Tools, and Techniques to Build Intelligent Systems, p07.
- [8] : Géron , June 2019 , Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Concepts, Tools, and Techniques to Build Intelligent Systems, p08.
- [9] [http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning#:~:text=Regression%20analysis%20consists%20of%20a,multiple%20predictor%20variables%20\(x\).&text=It%20assumes%20a%20linear%20relationship%20between%20the%20outcome%20and%20the%20predictor%20variables](http://www.sthda.com/english/wiki/regression-analysis-essentials-for-machine-learning#:~:text=Regression%20analysis%20consists%20of%20a,multiple%20predictor%20variables%20(x).&text=It%20assumes%20a%20linear%20relationship%20between%20the%20outcome%20and%20the%20predictor%20variables).
- [10]: <https://www.javatpoint.com/classification-algorithm-in-machine-learning>
- [11]: <https://www.geeksforgeeks.org/clustering-in-machine-learning>
- [12]: Géron , June 2019 , Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow, Concepts, Tools, and Techniques to Build Intelligent Systems, p14.
- [13]: Phil Kim,2017, MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence,p15
- [14]:<https://www.forbes.com/sites/bernardmarr/2016/03/22/a-short-history-of-deep-learning-everyone-should-read/?sh=70713ede5561>
- [15]: <https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063>

Bibliography

- [16]: Phil Kim,2017, MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence,p20
- [17]: Phil Kim,2017, MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence,p22
- [18]: <https://www.mygreatlearning.com/blog/types-of-neural-networks/#typesofneuralnetworks>
- [19]: <https://deeptai.org/machine-learning-glossary-and-terms/feed-forward-neural-network>
- [20]: https://computervision.fandom.com/wiki/Computer_vision
- [21] : <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e>
- [22]: https://indatalabs.com/blog/applications-computer-vision-across-industries?cli_action=1625154158.847
- [23]: <https://towardsdatascience.com/an-overview-of-computer-vision-1f75c2ab1b66>
- [24]:https://www.cse.unr.edu/~bebis/CS485/Handouts/ShapiroChapt2_ImageFormation.pdf
- [25]: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/binimage.htm>
- [26]: <https://medium.com/@rndayala/image-processing-basics-20054aa26365>
- [27]: by:Rebecca is a PhD student in computer vision and artificial intelligence
<https://realpython.com/python-opencv-color-spaces/>
- [28]: Abstract:Processing and Scanning Film from around the world. Located in Carlsbad, California.
<https://northcoastphoto.com/digital-explained>
- [29]: Richelieu R.A. Costa, Petroleum Engineering A plugin for computing the pore/grain network tortuosity of a porous medium from 2D/3D MicroCT image.
- [30]: https://en.wikipedia.org/wiki/Image_noise
- [31]: [https://en.wikipedia.org/wiki/Transparency_\(graphic\)](https://en.wikipedia.org/wiki/Transparency_(graphic))
- [32]: Vincent Tabora , Editor HD-PRO, DevOps

Bibliography

- [33]: [https://en.wikipedia.org/wiki/Contrast_\(vision\)#:~:text=The%20human%20visual%20system%20is,contrast%20ratio%20or%20dynamic%20range](https://en.wikipedia.org/wiki/Contrast_(vision)#:~:text=The%20human%20visual%20system%20is,contrast%20ratio%20or%20dynamic%20range)
- [34]: <https://svi.nl/ImageHistogram#:~:text=Image%20histograms>
- [35]: <https://www.computerhope.com/jargon/r/raster.htm>
- [36]: <https://bstigmafree.org/how-and-when-to-use-different-types-of-image-files>
- [37]: https://en.wikipedia.org/wiki/Raspberry_Pi
- [38]: Matt Richardson & Shawn Wallace “Getting started with Raspberry Pi “, Maker Media Inc.2014
- [39]: Raspberry Pi 4 Model B Datasheet ,Copyright Raspberry Pi (Trading) Ltd. 2019
- [40] : <https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-1298n-pwm-h-bridge>
- [41] : PYTHON Introduction to the Basics ,March 2021 S. Linner, M. Lischewski, M. Richerzhagen Forschungszentrum Jülich
- [42] : <https://intellipaat.com/blog/what-is-pycharm/>
- [43] : https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html
- [44] : <https://opencv.org/about/>
- [45]: https://www.tutorialspoint.com/tensorflow/tensorflow_introduction.htm
- [46]: <http://webindiasolutions.biz/webindiasolutions.co.in/predmac/>
- [47]: <https://keras.io/about/>
- [48] : <https://machinelearningmastery.com/object-recognition-with-deep-learning>
- [49]: Aliasgar Haji ,Self Driving RC Car using Behavioral Cloning.
- [50]: https://www.google.com/search?client=opera&hs=oRI&sxsrf=ALeKk02chXL21YGCG0y440GTKXQJyZTipA:1623944325788&q=What+is+SSD+network%3F&sa=X&ved=2ahUKEwj4g7GFgJ_xAhWGjKQKHfZjDTUQzmd6BAgDEAk
- [51]: <https://machinelearningmastery.com/object-recognition-with-deep-learning>
- [52]: <https://www.pantechsolutions.net/traffic-sign-detection-and-recognition-using-opencv-and-python>