People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Ibn Khaldoun University
Faculty of Applied Sciences
Electrical Engineering Department

End of Studies Dissertation
For obtaining the master's degree
Specialty : Embedded Systems Engineering

THEME

Autonomous drone for detection and tracking objects

Submitted by : *Abdessemed Sami*
Maati Mohamed

Supervised by : Mr Maamar Nour Eddine

According to the Jury :

Pr. Nasri Djilali	Pr	President
Dr. Belarbi Mustapha	MCA	Examinant
Dr. Safa Ahmed	MCA	Examinant

Academic year : 2023/2024

Acknowledgments

First and foremost, we would like to thank Allah SWT for granting us the strength and health to complete this project.

We would like to extend our deepest gratitude to our supervisor **Mr Maamar Nour Eddine**, especially for his availability, his deep understanding of our ideas, and his guidance, which has continuously motivated, encouraged, and steered us towards the path of success in this project.

We would also like to thank the members of the jury for dedicating their time to examine this project and provide critiques and suggestions to improve it.

Finally, we thank all the individuals who directly or indirectly contributed to the realization of this project, especially those who encouraged us with their ideas and provided feedback aimed at its improvement.

Dedications

We dedicate this work to our family

You gave us life, tenderness, and courage to succeed Everything we can offer you will not be able to express the love and gratitude we have for you. We truly admit that you are for us the light that guides us towards the path to success. It is to you that we owe our success
As a testimony, we offer you this modest work to thank you for the sacrifices you have made and the affection you have always shown us.

To our dear friends

We thank you for your friendships dear to our hearts, and we wish you all the happiness in the world.

Abstract

Our project consists of simulating and creating an autonomous drone hydrocopter, the aim of which is to identify objects and classify them, then after having identified the targets, the drone follows their suspicious movements, according to computer vision algorithms based on machine learning or deep learning. The purpose of this project is to monitor things or people for security purposes and protection against any suspicious operations such as theft or attack. This project is also used as an auxiliary tool in search and rescue operations.

Keywords : Hydrocopter, Autonomous, Algorithms, Computer vision, Suspicious movements, Machine learning, Deep learning.

Résumé

Notre projet consiste à simuler et réaliser un quadricoptère drone autonome, dont le but est d'identifier des objets et de les classer, puis après avoir identifié les cibles, le drone suivre leurs mouvements suspects, selon des algorithmes de vision par ordinateur basés sur le machine learning ou le deep learning. Le but de ce projet est de surveiller des choses ou des personnes à des fins de sécurité et de protection contre toute opération suspecte telle que vol ou attaque. Ce projet est également utilisé comme outil auxiliaire dans les opérations de recherche et de sauvetage.

Mots clé : Hydrocoptère, Autonome, Algorithmes, Vision par ordinateur, Mouvements suspects, Machine learning, Deep learning.

ملخص

يتكون مشروعنا من محاكاة وإنشاء طائرة بدون طيار هيدروكوبتر ذاتية التحكم ، والهدف منها هو تحديد الأشياء وتصنيفها، ثم بعد تحديد الأهداف، تتبع الطائرة بدون طيار تحركاتها المشبوهة، وفقًا لخوارزميات الرؤية الحاسوبية المعتمدة على التعلم الآلي أو التعلم العميق . الغرض من هذا المشروع هو مراقبة الأشياء أو الأشخاص لأغراض أمنية والحماية من أي عمليات مشبوهة مثل السرقة أو الهجوم. ويستخدم هذا المشروع أيضًا كأداة مساعدة في عمليات البحث والإنقاذ.

الكلمات المفتاحية : الطائرة بدون طيار، التحكم الذاتي، الخوارزميات، الرؤية الحاسوبية، الحركات المشبوهة، التعلم الآلي، التعلم العميق.

List of Figures

List of tables

List of Acronyms

2D	Two Dimensionals
AI	Artificial Intelligence
ACF	Aggregate channel features
BLDC	Brushless Motor Direct Currant
CCW	Counter Clockwise
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CT	Computed Tomography
CW	Clockwise
DC	Direct Currant
DIY	Do It Yourself
DJI	Dà-Jiāng Innovations Science and Technology Co
DPM	Deformable Parts Model
DSP	Digital Signal Processor
ESC	Electronic Speed Controller
ESP32	ESPRESSIF32
FC	Fully Connected
FPN	HyperText Transfer Protocol
FPS	Frames Per Second
GELAN	Generalized Efficient Layer Aggregate Network
GPIO	General Purpose Inputs Outputs
GPU	Graphics Processor Unit
GPS	Global Positioning System
GUI	Graphic User Interface
HALE	High Altitude Long Endurance
IC	Initial Conditions
IOT	Internet Of Things
ISM	Internet Protocol
JPEG	Joint Photographic Experts Group
LIPO	Lithium Polymer
LQR	Linear Quadratic Regulator
MALE	Medium Altitude Long Endurance
mAP	Named Entity Recognition
MEMS	Microelectromechanical Systems
MIPS	Million Instructions Per Second
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MPU	Magnetic Pickup Unit

NRF	Nordic Radio Frequency
PD	Proportional Derivative
PID	Proportional Integral Derivative
PGI	Programmable Gradient Information
PWM	Pulse Width Modulation
RCNN	Regions-Based Convolutional Neural Network
ReLU	Rectified Linear Unit
RF	Radio Frequency
SoC	System On Chip
SPP	Relation Extraction
STM	Radiology Information System
SVM	Support Vector Machine
SRAM	Static Random Access Memory
SSD	Single Short Detection
tanh	Hyperbolic Tangent
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once

Table of contents

General Introduction

"Any advanced technology is magical." This is what Sir Arthur Charles Clarke said about technology, it facilitates human life and enhances productivity and efficiency in many areas, whether they are health, economics, entertainment, or others. This project envisions an autonomous drone capable of identifying and tracking objects for the purpose of establishing a robust surveillance system or carrying out search and rescue operations.

This thesis aims to design an innovative quadcopter drone. The drone should be able to fly at high altitudes for extended periods and must be equipped with a high-resolution camera. The objective of this research is to develop a drone capable of identifying, classifying, and tracking objects through computer vision based on machine learning and deep learning.

This project is relevant to :

- Law enforcement and security officials : To protect citizens against criminal activities such as assaults, home and shop thefts, drug trafficking, or during search and rescue operations in natural disasters.
- Military : Through military reconnaissance and surveying operations.
- Landowners and farmers : To monitor their properties and produce.

The implementation of this project, from planning to simulation and realization, takes approximately 6 months, depending on the temporal phases. This is because the design of a drone follows several distinct phases, each with its own activities and deadlines, including the phases of planning and initial design, prototyping, testing and improvement, certification, and regulation. However, it is important to note that these phases may overlap, and the duration of each phase may vary depending on the project's complexity, available resources, and other factors.

Activities associated with a drone can take place in various locations depending on the specific purpose of the flight and the type of drone used, including industrial and construction areas, agricultural spaces, urban areas, military zones, search and rescue areas, etc

The aim of this work is to develop a quadrotor drone, followed by synthesis, simulation, and realization of this project.
The thesis is organized as follows :

- Chapter 1 : Presentation and description of the characteristics, benefits, and applications of a drone.

- Chapter 2 : It involves the modeling of the drone, as well as the study of the dynamics and aerodynamics of the quadcopter and the forces acting on it, such as gravity, drag, and thrust. As well as stability resulting from forces acting on it using a PID controller.
- Chapter 3 : Concerns algorithms for object detection and tracking through computer vision.
- Chapter 4 : Results of the simulation and realization of this drone.
- Conclusion.

CHAPTER 1

Overview of Drones

1.1 Introduction

The use of drones has significantly evolved over the last decade. The study of these systems has attracted many researchers working in disciplines particularly associated with automation, electronics, mechanics, and aerodynamics. A drone (UAV, or unmanned aerial vehicle), or simply drone, is an unmanned flying vehicle capable of carrying cameras, sensors, communication equipment, or other devices. They are used in surveillance and security missions, as well as reconnaissance or information-gathering missions. Some of them are multirotors, and others are fixed-wing aircraft.

1.2 Drone history

The concept of unmanned aerial vehicles (UAVs) dates back to the mid-19th century, and the first recognized drone was an aerial target used by the British Royal Navy in 1917. Since then, drones have come a long way, and their applications have multiplied and expanded beyond military and reconnaissance areas[1].

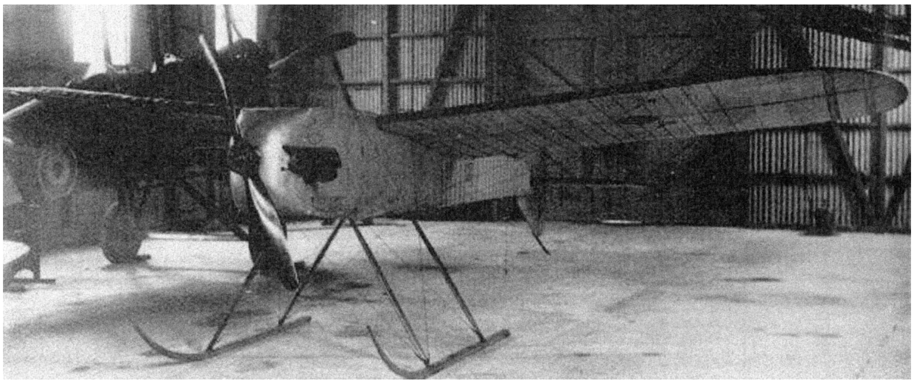


Figure 1.1: The first successful remote-controlled aircraft in 1917, the British "Aerial Target".[2]

This invention paved the way for the development of drones in the following years. Fast forward to the 21st century, and the drone market has undergone a radical change with the advent of quadcopters and multicopters. Drones gained popularity in 2006 when the Paparazzi project developed an open-source, affordable autopilot system for drones. This pivotal moment represents a turning point in drone development. The Chinese company DJI was founded and revolutionized the drone industry. The applications of drones have since expanded to include :

- Aerial photography.
- Racing and sports.

- Wildlife conservation.
- Agricultural and environmental monitoring.
- Emergency response.

Year	Drone model	Notable features
1917	Aerial Target	first recognisable drone
1918	Kettering Bug	first radio-controlled aircraft
1935	DH.82B Queen Bee	Inspire drone term
1960	MQM-57 Falconer	aerial reconnaissance
1990	Gnat 750	Advanced capacity of surveillance
2001	RQ-1 Predator	Army for miltary uses
2010	Parrot AR.Drone	first large public drone

Table 1.1: drones progress table [1]

1.3 Related works

The design of a quadcopter drone varies from one design to another depending on the flight controller, which contains the development board used to program the drone to stabilize and fly by controlling the motor speed, processing sensor data, and calculating adjustments to maintain stability and direction of the drone. The works that have already been designed include :

- **Design and Implementation of a Fire Detection and Survivor Drone Based on IoT :**
  This project involves designing and developing a firefighting drone system that can detect fires and assist rescue efforts in the event of a fire hazard or fire incident. This system is based on the Arduino UNO board as the flight controller and an ESP32-CAM module for the camera.[3]

- **Design and realization of a mini drone :**
  The objective of this work is to create a quad-rotor drone based on an Arduino Nano board, which is already programmed to stabilize (using PID) and control the quad-rotor in the form of PWM signals.[4]

- **VAGADRONE - intelligent and fully automatic drone based on Raspberry Pi and Android :**
  This article proposes the design of a drone based on the Raspberry Pi and the Navio+ board that can perform multiple missions, including autonomous flight and obstacle detection. Therefore, it has robust and efficient operation during navigation.[5]

- **Facial recognition drone :**
  A drone model based on the Raspberry Pi and Ardupilot (an open-source autopilot system) is presented in this article. The Raspberry Pi can take photos using a camera attached to the drone, which can then process them to recognize faces and upload all required information.[6]

1.4 Presentation of our project

There are many types of drones, including fixed-wing drones, rotary-wing drones, and multirotor drones, and their applications are also numerous, as they are used for surveillance systems, reconnaissance operations, etc.

This project focuses specifically on quadrotor drones, which contain four motors and propellers, with the aim of creating an advanced security and surveillance system as well as applying reconnaissance and search operations. So how does this drone work to achieve these applications ?

This project is actually a composite project of several projects, as the design and realization of the drone can be considered separate projects, in addition to target identification and tracking, which are attributed to computer vision and to which many studies have also been applied. But one of the most important reasons that prompted us to design this project is that it is a unique idea that has never been implemented before. Most programmers who wanted to apply computer vision to a drone have implemented artificial intelligence algorithms on a pre-equipped drone. Therefore, it can be concluded that the drone project for identifying and tracking targets is a project composed of :

- **Designing and Building a Drone:**
  This requires an understanding of aerodynamic principles that will influence the drone's design, including wing or rotor configuration, stability, drag, etc. In addition to determining the payload that the drone will carry, such as cameras, sensors, scientific instruments, etc., with

the design of the control system, including sensors (gyroscopes, accelerometers, GPS), actuators, and a processor to stabilize and control the drone, let's not forget to integrate a navigation system, usually based on GPS, to allow the drone to move autonomously and follow pre-programmed trajectories.

- **Study and Application of Computer Vision in the Drone:** Computer vision relies on a set of techniques ranging from image acquisition to advanced analysis processes based on algorithms, typically artificial intelligence algorithms, including machine learning or deep learning. These techniques are applied to the drone for the purposes of object recognition, detection, and tracking.

1.4.1 Drones applications

Today, drones are used worldwide for many different applications, ranging from surveillance of suspicious and illegal activities to wildlife conservation and emergency response. This drone project targets the following applications :[7]

- Surveillance of Illegal Activities.
- Military Reconnaissance and Surveillance Operations.
- Agricultural application for monitoring crop health.
- Livestock management for monitoring and tracking livestock in fields in real-time.
- Search and rescue to locate missing individuals.

1.4.2 Drone Classifications

Drones can be classified based on their size, weight, propulsion method, and capacity to carry a payload. These classifications are summarized as follows :

- **Classification by weight and size :**[8]
  1. **Micro drones or very small drones:** Between 1 and 50 cm and less than 250 grams. These types of drones perform tasks that larger drones cannot perform, such as espionage tasks in hostile areas, for example. They are intended for such tasks due to their

very small size and low weight.

2. **Mini drones:** They have a slightly larger wingspan than micro drones. The size of the mini drone ranges from 50 cm to 2 meters and from 250 to 2500 grams. They facilitate the implementation of relatively low autonomy (from 10 to 30 minutes). Mini drones are often used for aerial recording and photography missions.

3. **Medium drones:** named in English (MALE: Medium Altitude Long Endurance). These are drones that resemble miniature airplanes. They are slightly heavier and measure slightly over 2 meters. Their weight is about 200 kg. They look a lot like airplanes but are lighter. Medium drones are used for military missions.

4. **Large drones** named in English (HALE, High Altitude Long Endurance). These are drones that resemble airplanes; due to their size and wingspan, their weight ranges from about 250 to 600 kilograms. They are capable of staying in flight for very long periods and collecting information over very long periods (between 12 and 48 hours). These are very heavy devices used exclusively for military missions.

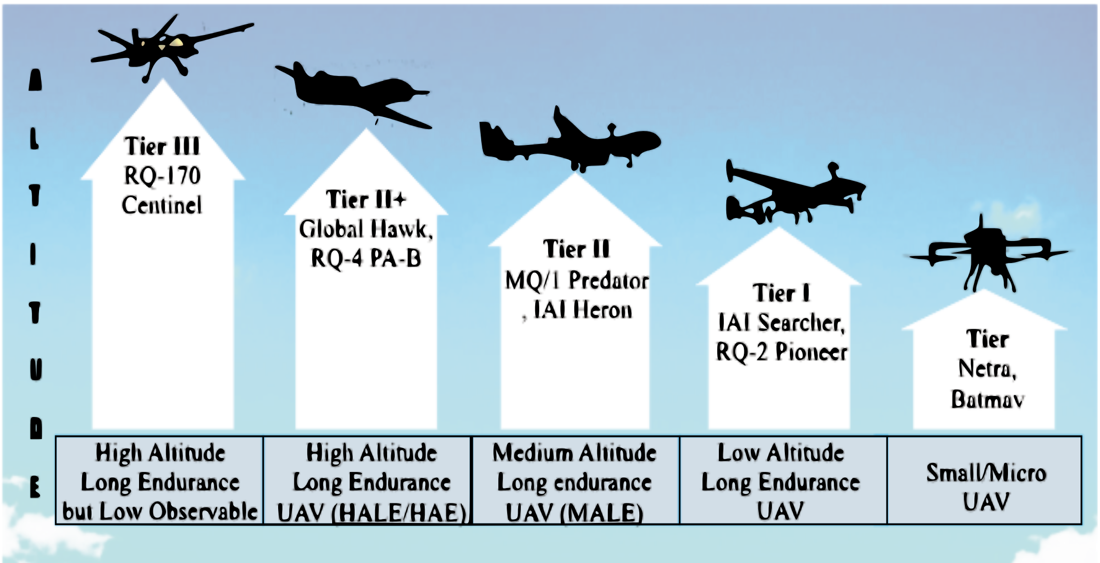


Figure 1.2: drones classification based on size[9]

- **Classification by propulsion method :**[10]

  1. **Fixed-wing drones :**
     These drones use fixed wings for their mode of movement. They can be either :
     - Heavier-than-air (airplane type).
     - lighter-than-air (airship type).

  2. **Flapping-wing drones :**
     They can be either :
     - Bird type.
     - insect type.

  3. **Rotary-wing drones :**
     Rotary-wing drones are newer than fixed-wing drones. They use rotors to generate the lift needed for flight. These rotors are rotating blades that produce lift when air passes through them. Rotary-wing drones are generally smaller and lighter than fixed-wing drones. They are also more agile and easier to pilot. Among their features are vertical takeoff and landing, and they are capable of flying at low speed and low altitude. They are divided into five categories:

     - **Monorotor :** A single rotor is used to provide lift and propulsion. Monorotor drones are generally small and light, and they are often used for recreational or aerial photography purposes.

     - **Biropters:** There are many configurations with two rotors, including the classic helicopter with a main rotor and a tail rotor. The rotors of non-cyclic-pitch aircraft, such as helicopters, are rotated using ailerons. There are also devices with two rotors rotating in opposite directions and ailerons bathed in the airflow of the rotors.

     - **Trirotors:** The tri-rotor consists of two rotors at the front, rotating in opposite directions to change pitch, and one rotor at the back to adjust roll. It is less efficient in flight than the quadrotor.

     - **Quadrotors:** have four rotors; they are the most common multirotor drones. They are easy to pilot and are available in a variety of sizes and prices.

– **Hexarotors:**A flying machine with six rotors is called a hexarotor.

– **Octarotors:** Eight rotors are used. Although they are heavier and larger than hexarotors, octarotors are even more stable.

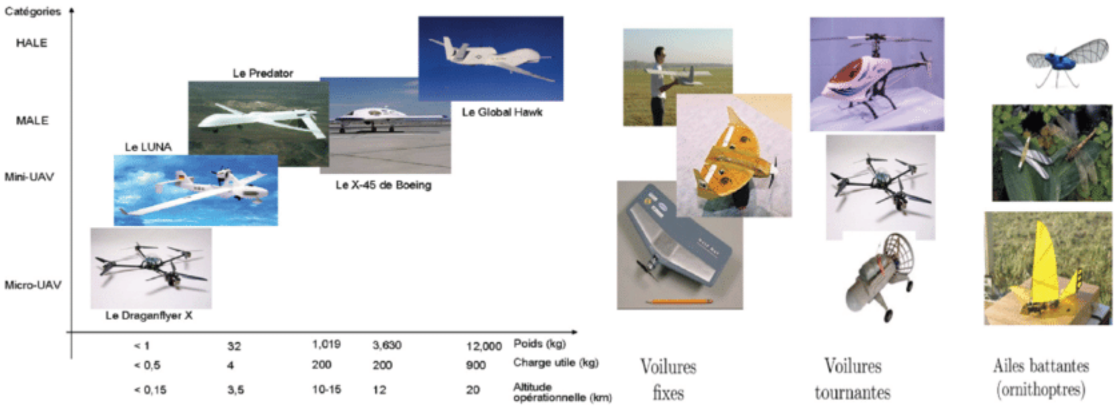


Figure 1.3: the categories of drones based on the propulsion method[11]

1.5 Drone Description

1.5.1 Drone Characteristics

The technical characteristics of a drone are the elements that determine its performance and capabilities. It is important to choose the drone that suits your needs and understand how it works. Among the most important technical characteristics found in drones are:[12]

- Flight performance.
- Camera specifications.
- Battery autonomy.
- Controller and range.
- GPS and navigation systems.
- Size and portability.
- Build quality and durability.
- Security features.

- Because patients differ biologically, it is used to locate organs in real-time during surgery.
- calculation of social distance violations between individuals during the COVID-19 epidemic.

1.5.2 Drone architectures

The architecture of a drone system can be characterized by four types of systems :

1. **Control system :**
   The control system is the brain of the drone and is divided into two systems: one system that controls the stability of the drone based on a smart card and a radio communication receiver, and another system representing the drone remote control based on a smart card and a radio communication transmitter. Each of them is represented as follows:

   - **Drone control system**
     (a) **ESP32 microcontroller:**
     ESP32 is a microcontroller based on the Xtensa dual-core architecture (or single-core). It contains a 32-bit LX microprocessor running at 160 or 240 MHz and providing up to 600 DMIPS, integrating Wi-Fi and Bluetooth management (up to LE 5.0 and 5.11) in dual mode, and a DSP with a co-processor for ultra-low power consumption (ULP).[13]

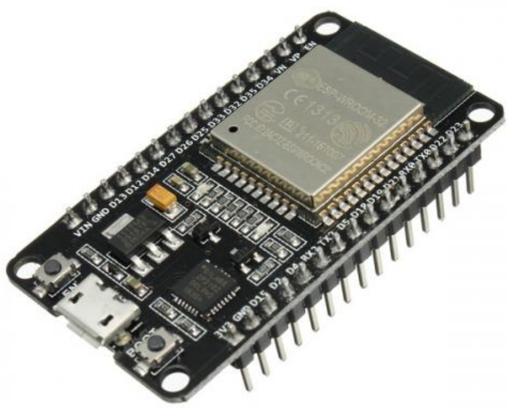


Figure 1.4: ESP32-WROOM micro-controller[14]

(b) **NRF24L01 with antenna :**
NRF24L01 is a radio transceiver module used to send and receive data at the 2.4 to 2.5 GHz ISM operating frequency. and it can also operate on 125 different channels. While transmitted data is received by the RF receiver. The operating frequency range of both the RF transmitter and the RF receiver is the same. The antenna is used to extend the communication range between the transmitter and the receiver from a few meters up to 2 kilometers. It operates in three modes :[15]

- **Transmission mode.**
- **Reception mode.**
- **Transceiver mode.**

Figure 1.5: NRF24L01 module with antenna[16]

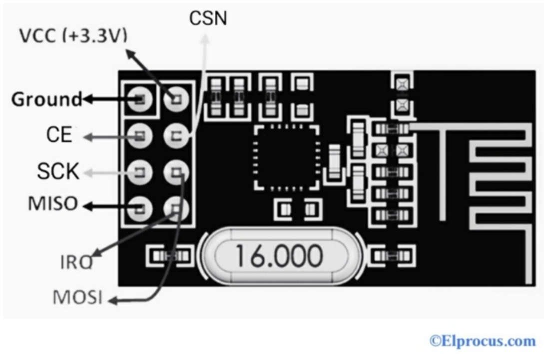


Figure 1.6: NRF24L01 pinout diagram[15]

- **Remote control system of the drone**

(a) **ESP32 microcontroller :**
The ESP32 microcontroller is the brain of the drone's remote control system. It converts the analog inputs from the joysticks into digital data and generates control signals for the flight controller by moving the drone's rotors.

(b) **NRF24L01 with antenna :**
In this control system, the NRF24L01 module is considered a transmitter.

(c) **Joystick Module Switch Sensor :**
The joystick module, or joystick module, is used in DIY projects

using microcontrollers, such as robot control. As we know, the module provides analog outputs, so it can be used to power analog inputs depending on direction or movement. It can also be connected to a moving camera to control its movement.

Figure 1.7: Joystick module[17]

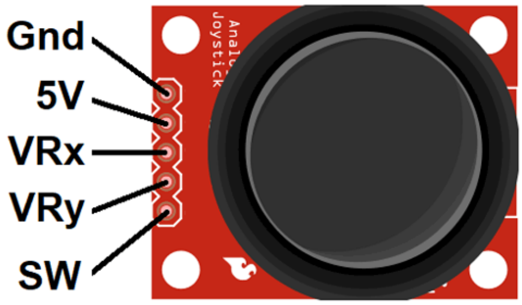


Figure 1.8: joystick module pins diagram[17]

(d) **Battery :**
A lithium-polymer (Li-Po) battery is a rechargeable lithium-ion battery using a polymer electrolyte instead of a liquid electrolyte. These batteries are used in applications where weight is a critical factor, such as mobile devices, remote-controlled aircraft, and some electric vehicles. They offer higher specific energy than other types of lithium batteries. And in the drone remote control, its role is to power the various components of this remote control.[18]

Figure 1.9: Li-Po batteries[19]

(e) **NRF24L01 Wireless Adapter Module :**
This module serves as a power intermediary to ensure a stabilized voltage for the NRF24L01, and it is required when increasing the radio transmission power.

It can be replaced by a parallel capacitor to avoid any malfunction of the NRF24L01 module, but it is considered that the intermediary power module is the best for stabilized voltage.[20]

Figure 1.10: The power capacitor with the NRF24L01+ module[20]

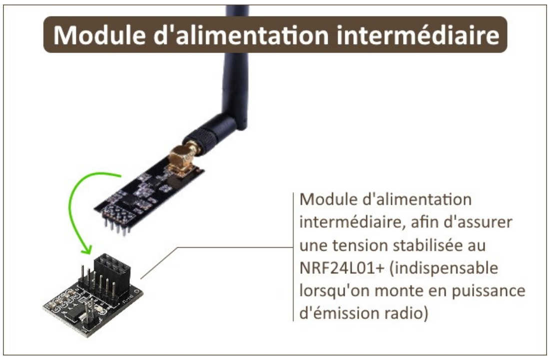


Figure 1.11: The intermediate power supply module with the NRF24L01+ module[20]

2. **Propulsion System :**

   (a) **Brushless motors :**
   The quadrotor drone is equipped with four BLDC (brushless direct current) motors. These are synchronous motors without a commutator and brushes. These motors have a rotor (with permanent magnets) in the form of a cage and a stator consisting of windings. The rotor and stator are made from ferromagnetic materials. An air gap separates the two.[19]

Figure 1.12: Brushless DC motors[19]

- **Principles of Operation of Brushless DC Motors :**

  Brushless DC motors consist of a set of coils, called the armature winding, inside another set of coils or a set of permanent magnets, called the stator. A voltage is applied to the coils, which creates torque in the armature and causes movement.[21]

- **Types of Brushless DC Motors :**

  There are two categories of brushless DC motors :

  – **Inrunner Motors:**
    Inrunner motors have the rotating part inside while the outer shell remains stationary. Remote-controlled cars often use Inrunner motors because they can spin much faster than Outrunner motors.[21]

  – **Outrunner Motors:**
    Outrunner motors have the rotor on the outer area of the motor. They can produce more torque, allowing them to drive larger propellers used on airplanes and multirotors.[21]

- **Parameters of Brushless DC Motors:**

  It is important to carefully consider all relevant parameters when selecting a brushless motor for a particular application, for example:

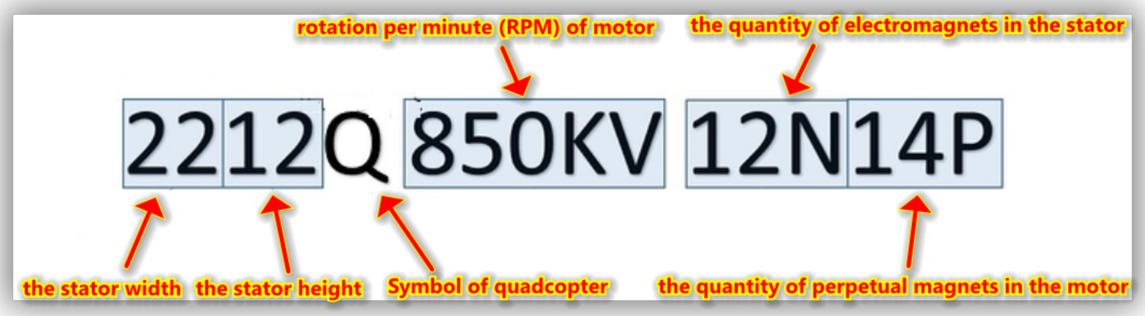


Figure 1.13: The parameters of a brushless DC motor[21]

- **Advantages and Disadvantages of Brushless Motors:**

  Brushless DC machines (BLDC motors) have several advantages over conventional DC motors, such as:

  – better thermal behavior due to no friction and the external rotor, resulting in efficient cooling.

– lower rotor inertia (no commutator).

– No vibrations, sparks, or friction.

– low noise and low voltage.

– higher speed range.

– long lifespan.

However, brushless DC motors also have several disadvantages, including:

– need for a good controller.

– risk of poor starts or cuts.

– more expensive than conventional DC motors.

(b) **Speed Controllers :**
The performance and stability of drones are strongly influenced by electronic speed controllers (ESCs). An ESC is a device capable of regulating the speed of an electric motor in a drone. It acts as an intermediary between the motor and the flight controller, allowing precise control of the drone's speed and performance.

Figure 1.14: The electronic speed variator (ESC)[22]

An electronic speed controller typically consists of the following components :[23]

- **Microcontroller :** responsible for processing signals received from the flight controller.

- **Power MOSFETs :** high-power transistors that regulate the current flow to the motor. They provide a switching mechanism to control the motor's speed and direction.

- **Voltage regulator :** ensures that the ESC receives a stable power supply from the drone's battery.
- **Connectors :** for interfacing with the flight controller, motor, and power source.

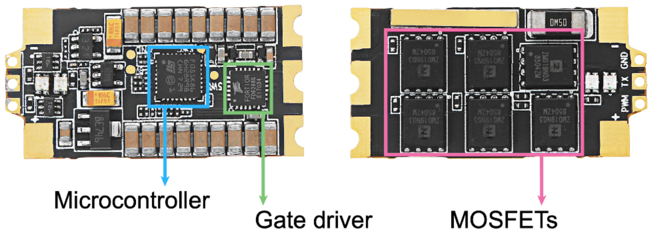


Figure 1.15: The components of an electronic speed variator[22]

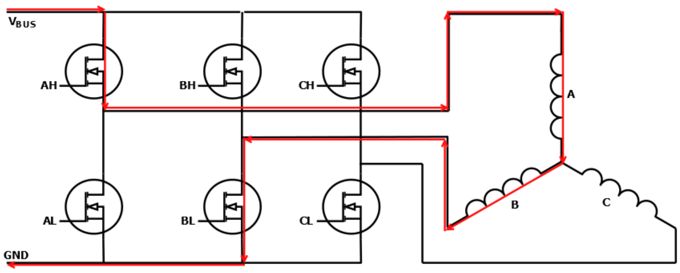


Figure 1.16: The main function of ESC[24]

(c) **Propellers :**

Drone propellers function by rotating with the force applied by the motor. The high atmospheric pressure at the bottom of the propellers provides lift to the entire drone. Additionally, their rotation keeps the drone stable and propels it forward.

- **What are drone propellers made of ?**

Drone propellers are typically made from two different types of materials. Most drone models use propellers made of nylon or high-quality flexible plastic. However, carbon fiber propellers are used in high-quality drones, such as the DJI Matrice 600, as they are quieter and offer better flight performance.[25]

A propeller is described using three parameters: its size, pitch, and number of blades. For example, a propeller might measure (5×4.8×2). The first number (5) primarily refers to the size in inches. The pitch is indicated by the next number, which is 4.8 inches. Finally, the number of blades is indicated by the third number (2).[26]

A larger propeller has more contact with the air, increasing the drone's stability during hover. However, a smaller diameter requires more effort to move safely through the air. Nonetheless, they can accelerate or decelerate easily, making them more responsive than their larger counterparts.

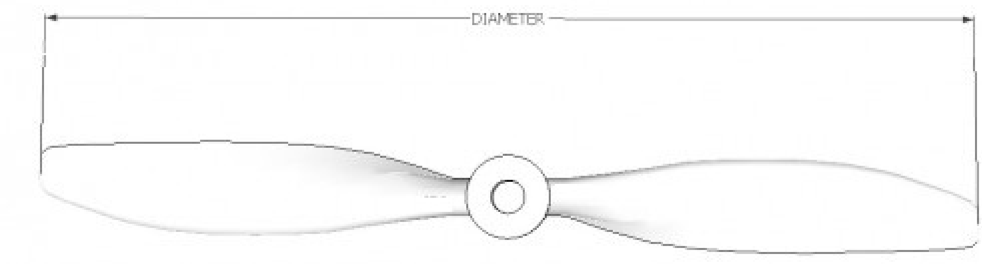


Figure 1.17: Drone propellers diameter

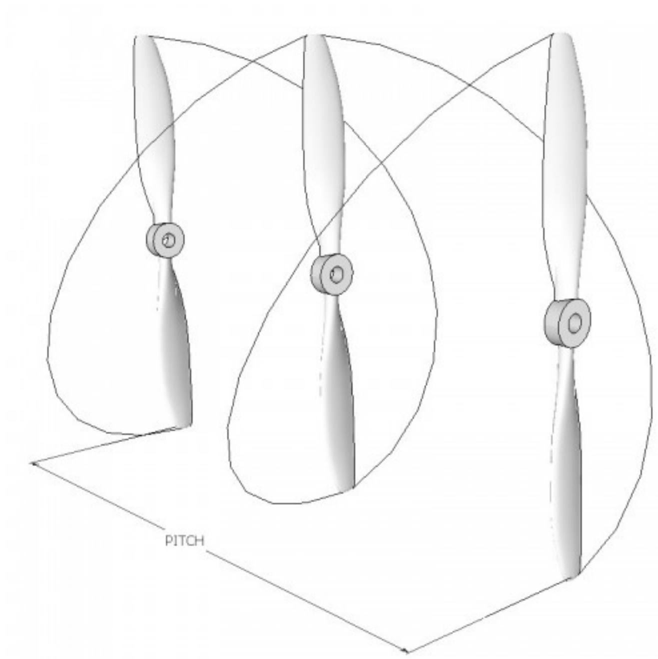


Figure 1.18: Drone propellers pitch

- **What is the principle of rotation for drone propellers ?**

  Drone propellers rotate in opposite directions so that the resulting torque acting on the drone is zero, preventing the main body from rotating. The force applied to the propellers is the kinetic force that moves the stator, and according to Newton's third law of motion (the law of action and reaction), there is an equal and opposite force applied to the motor stator. Since the stator is fixed to the drone's frame, this causes it to rotate in the opposite direction from the motor rotation. To avoid this scenario, two facing propellers must each rotate in the same direction. That is, two of them must rotate clockwise (CW), and the other two must rotate counterclockwise (CCW). Generally, a drone's propeller rotates at the necessary speed to facilitate flight. Researchers at Virginia Tech have stated that the majority of drone propellers rotate at 8000 revolutions per minute, resulting in a frequency of 133 times per second. Of course, the size of the drone determines the exact speed.

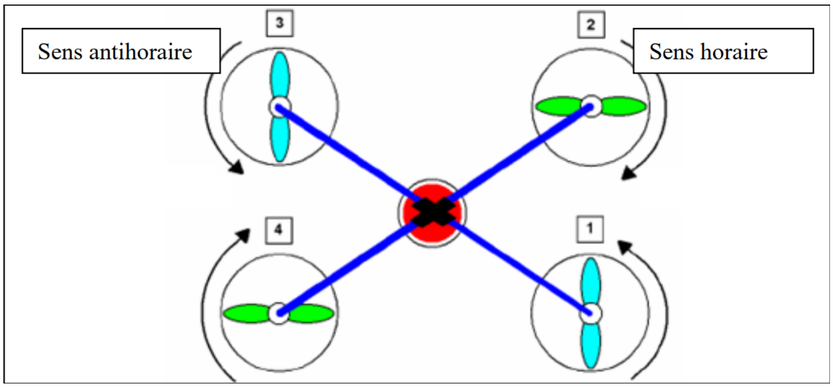


Figure 1.19: Directions of rotation of drone propellers

- **What are the advantages and disadvantages of drone propellers ?**

  In terms of mechanical efficiency, the fewer the blades, the more efficient the drone. Therefore, two-blade propellers are considered the best, and their most significant advantages and disadvantages are :

  - **The advantages :**

* Better foldable design.
* Higher speed.
* lightweight and portable.
* More efficient.
* More durable.

– **The disadvantages :**

* Affected by wind due to its lightweight.
* Low stability.
* Tedious.

3. **Navigation system :**

(a) **MPU6050 module :**
The MPU6050 module is a microelectromechanical system (MEMS) with a three-axis accelerometer and a three-axis gyroscope inside. This allows us to measure speed, acceleration, orientation, displacement, and many other motion-related parameters of a system or object.

- **Accelerometer sensors :**
An accelerometer is a device capable of measuring the linear acceleration of a drone. Therefore, to calculate the three linear accelerations of the drone along the three orthogonal axes of space (x, y, and z), three accelerometers are required.

- **Gyroscopic sensors :**
A gyroscope can be used to calculate the absolute rotation of an object around an axis. Therefore, three gyroscopes are required to determine the three rotations along the three orthogonal axes of space (x,y,z).

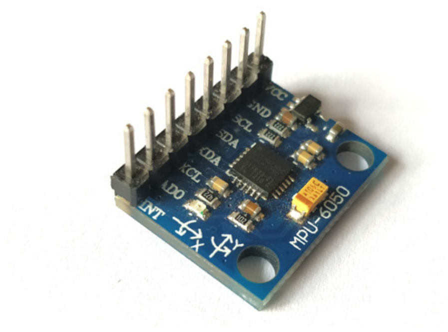


Figure 1.20: MPU6050 model[27]

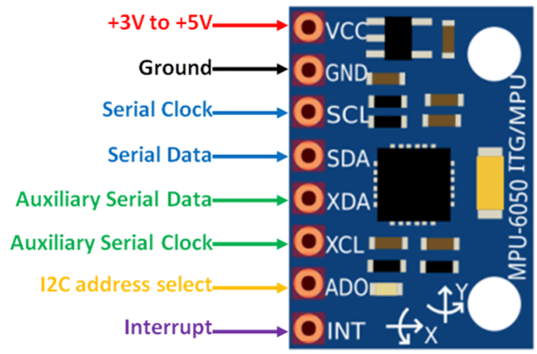


Figure 1.21: MPU6050 module pin assignment[27]

(b) **Global positioning system (GPS) sensors :**
A GPS positioning system uses signals emitted by a specially designed satellite constellation to locate the exact position of an object on Earth. The position can be physically represented on a map in terms of latitude, longitude, and altitude, with an accuracy of about ten meters for standard systems. That is why they are primarily used for locating drones.

Figure 1.22: GPS models[28]

4. **Imaging system :**

(a) **ESP32 cam :**
An ESP32-CAM is a low-cost, low-power system-on-chip (SoC) that integrates a camera and a microcontroller. It is based on the Espressif Systems ESP32 microcontroller and features a small form factor, low power consumption, and a camera interface. The ESP32-CAM can be used for various applications, such as security systems, remote monitoring, and machine vision. It has a built-in camera module that supports JPEG and MJPEG image formats

and can be easily integrated with a variety of sensors and devices using its I/O pins.[29]

Figure 1.23: ESP32-CAM[29]

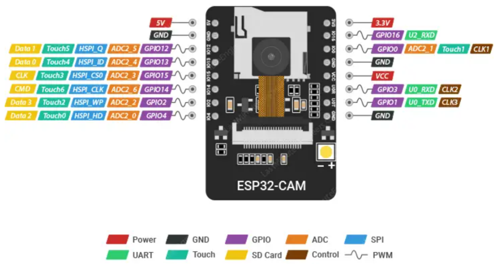


Figure 1.24: ESP32-CAM pinout[29]

Technical specifications of ESP-CAM	
Processors	Xtensa dual-core 32-bit LX6 microprocessor running at 240 MHz and operating at up to 600 DMIPS. Ultra low consumption co-processor.
Memories	SRAM of 520 KB. Internal flash memory of 4 MB. External PSRAM memory of 4 MB.
Camera	2 megapixel OV2640 sensor. Board size: UXGA 1622 × 1200. Image transfer rate from 15 to 60 fps.
Wireless connectivity	Wi-Fi : 802.11 b/g/n. Bluetooth : v4.2 BR/EDR and BLE
Power Management	Deep sleep current of 5 uA.

Table 1.2: drones progress table [1]

1.6 The advantages and disadvantages of drones :

The advantages of drones are numerous, including :[30]

- Non-polluting.
- Remote access to dangerous or inaccessible areas.

- Proximity to the subject to be photographed or filmed (a few meters or less).
- Collecting precise and reliable data to improve decision-making and operations.
- The ability to adapt to various environments and tasks is due to the diverse range of drones available.

Drones also have many disadvantages, including :[30]

- Weather conditions, particularly wind, must be within the drone's limits to ensure good results.
- Local restrictions and prior authorizations are required to fly in large cities.
- Increased risks of collisions with airplanes, birds, people, or buildings raise questions about public aviation safety.
- Exposure to hacking and malicious use leads to cybersecurity and data protection issues.
- Battery capacity is limited, limiting flight duration and requiring frequent recharges.
- Vulnerable to radio and GPS interference, which can disrupt drone operation and affect the accuracy of collected data.
- Noise nuisance caused by drones, which can disturb wildlife and vegetation and inconvenience residents.

1.7 Conclusion

In this chapter, we provide an overview of the various general concepts and definitions of drones. Additionally, we classified drones based on their sizes, number of propellers, and different shapes. Furthermore, we outlined the advantages and disadvantages of drones. Additionally, we discussed some work related to the design of quadcopter drones with various devices and their applications. Next, we will present a study on computer vision algorithms that allow drones to detect, identify, and track objects.

CHAPTER 2

Modeling and controlling a quadcopter

2.1 Introduction

Modeling is important to the study of all systems because understanding the behavior of the physical system and its surroundings is crucial. The quadcopter is thought to be one of the most challenging flying systems because of the many physical elements that affect its dynamics, such as aerodynamic effects, gravity, gyroscopic effects, friction, and moments of inertia. The main reason for this complexity is that the flying mode affects how these impacts manifest. Furthermore, the quadcopter is an aerodynamically unstable, nonlinear system that needs a suitable control system to stabilize it.
The dynamic modeling of the quadcopter based on the missions scheduled and the operator-defined navigation environment will be demonstrated in this chapter. It concentrates on piloting and managing a quadcopter equipped with a PID controller, which has proven to operate very well in a range of situations, such as drone racing, where dexterity and agility are essential. On the other hand, it has a number of drawbacks, including the differential instrument amplifying noise and the occasional high control signal caused by wind that exceeds the objective and climbs further with cumulative error correction.

2.2 Principle of flight

According to Newton's third law of motion, there is an equal and opposite reaction to every action. The quadcopter's mounted propellers force air downward. This shows action according to Newton's third law. There needs to be an equal and opposite reaction for Newton's law to be true. The quadcopter is pushed upward by this reaction. The quadcopter starts to move upward when this force surpasses the force of gravity dragging it down. The quadcopter's four propellers are located in the four corners of its frame. The speed and direction of spinning of each propeller are adjusted individually for the drone's balance and mobility. To keep the system balanced, the four rotors are separated by an equal distance. One set of rotors rotates clockwise, and the other pair rotates counterclockwise.

The movement of the quadcopter is controlled by changing the speed of each rotor to change the lift force and torque generated by each of them, and this is represented in the following :[31]

- **Vertical (throttle/Hover) :**
  Quadcopter can move up against gravity or come down in a controlled manner.
  - If all four propellers run at normal speed, then the drone will move down.
  - If all four propellers run at a higher speed, then the drone will move up.

- **Rotational (yaw) :**
  Clockwise or anti-clockwise turns about the vertical axis.
  - If two propellers on the right diagonal run at high speed, then the drone will rotate in an anti-clockwise direction.
  - If two propellers on a left diagonal run at high speed, then the drone will rotate in a clockwise direction.

- **Front-to-back Lateral (pitch) :**
  Tilting along the lateral axis, causes the quadcopter's nose to dip or rise. This in turn enables forward or backward motion.
  - If two rear propellers run at high speed, then the drone will move in a forward direction.
  - If two front propellers run at high speed, then the drone will move in a backward direction.

- **Left-to-right Lateral (roll) :**
  Tilting along this axis, enables movement towards left or right.
  - If two right propellers run at high speed, then the drone will move in the left direction.
  - If two left propellers run at high speed, then the drone will move in the right direction.

2.3 Dynamic model of a quadcopter

Quadcopters are difficult to model since their dynamics are very nonlinear and totally coupled. The following working hypotheses may help you better understand the dynamic model described below:

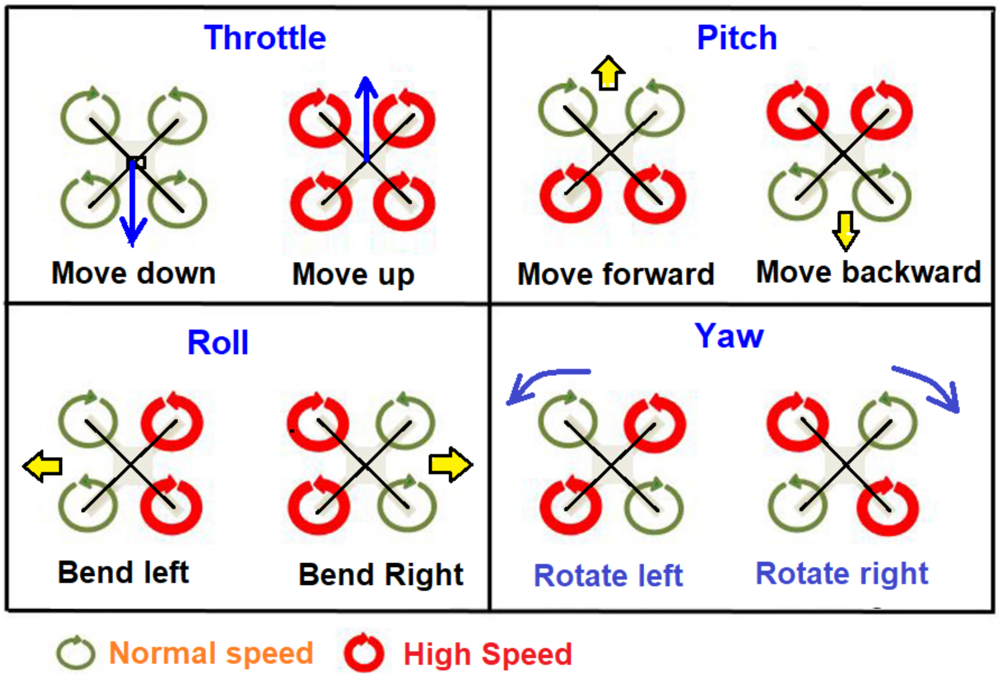


Figure 2.1: The movements of a quadcopter[32]

- The structure of the quadcopter is assumed to be rigid and symmetrical, hence the assumption that the inertia matrix is diagonal.
- The propellers are assumed to be rigid to be able to neglect the effect of their deformation during rotation.
- The landmark linked to this structure is generally assumed to coincide with its center of gravity. This leads us to consider the dynamics of the quadcopter as those of a rigid body in space.
- The lift and drag forces are proportional to the squares of the rotational speed of the rotors, which is a very close approximation of the aerodynamic behavior.
- Atmospheric conditions are the standard conditions of pressure and temperature.
- Sensors (e.g., gyroscopes, accelerometers) measure relative to body-fixed-axes.

2.3.1 Reference frame

To evaluate the mathematical model of the quadrotor, we use two references: a fixed reference linked to the earth, the inertial reference frame E (O, X, Y, Z), and another mobile called the body fixed reference frame B (o', x, y, z). To define the orientation over time of the quadcopter frame relative to the inertial frame, we use the Euler angles (ϕ, θ, ψ) which represent roll, pitch and yaw respectively.

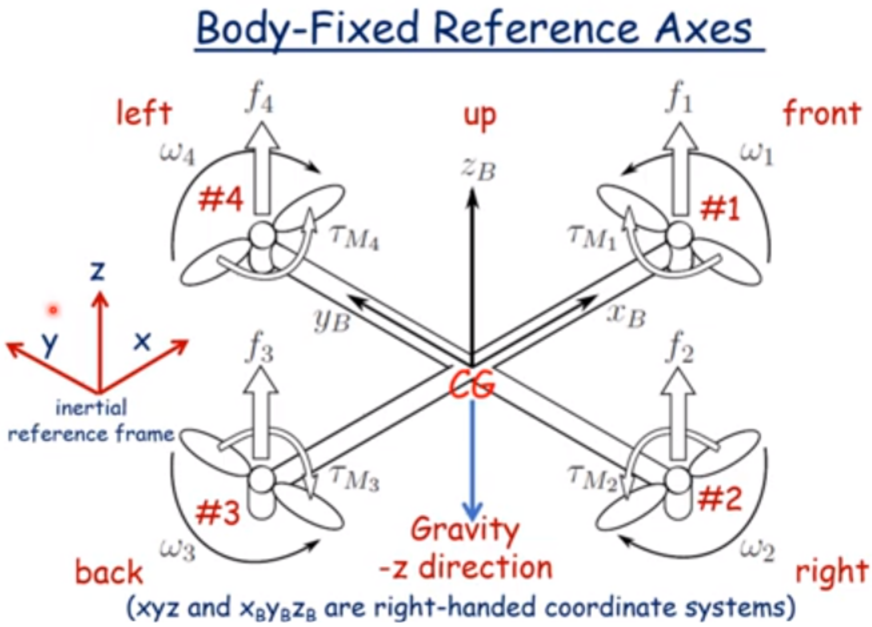


Figure 2.2: The references frames concerned to the drone dynamics

While :

- *f1, f2, f3, f4 :* Lift forces of each rotor.
- $\tau m1, \tau m2, \tau m3, \tau m4$: The moments (torques) of each rotor.
- $\omega 1, \omega 2, \omega 3, \omega 4$: Angular velocities of each rotor.
- *x, y, z :* The inertial reference frame coordinates.
- *xb, yb, zb :* The body fixed reference frame coordinates.

2.3.2 Euler's angles

Euler angles are a straightforward technique to express a three-dimensional rotation. They are widely employed in robotic applications. As we are in the presence of conservative forces (the weight is always in the Z direction of the fixed reference of the Earth E (O, X, Y, Z), we therefore need the matrix of 3D rotation to bring them back into the mobile reference frame B (o', x, y, z) Indeed, the weight forces will thus be expressed using the angles represented, respectively:

- Roll angle (ϕ) : $-\frac{\pi}{2} < \phi < \frac{\pi}{2}$
- Pitch angle (θ) : $-\frac{\pi}{2} < \theta < \frac{\pi}{2}$
- Yaw angle (ψ) : $-\pi < \psi < \pi$

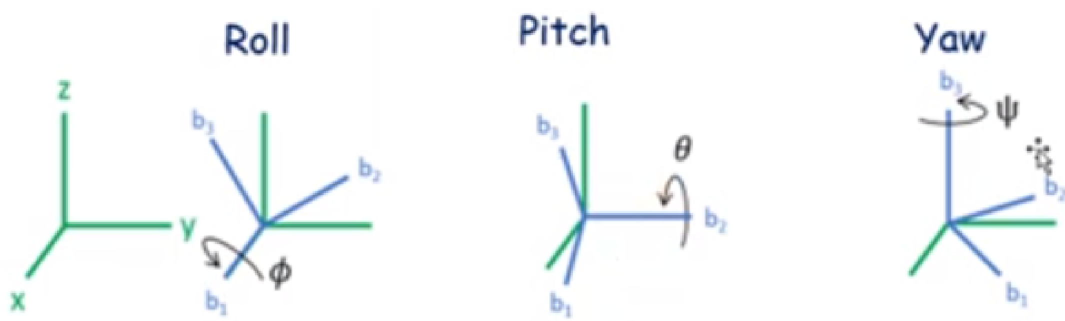
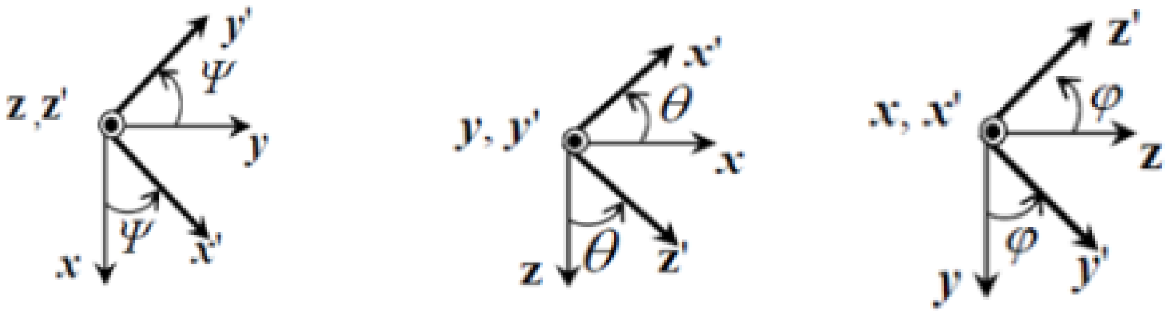


Figure 2.3: Euler's angles in the three rotational motion

And we can easily set up the three elementary submatrices and conduct the following multiplication :



(a) Yaw (b) Pitch (c) Roll

Figure 2.4: The rotational movements of drone[21]

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} b1 \\ b2 \\ b3 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2.3}$$

This matrix, which connects the body-fixed reference frame to the inertial reference frame, is classified as a rotation matrix.

$$\begin{bmatrix} \hat{b_1} \\ \hat{b_2} \\ \hat{b_3} \end{bmatrix} = \begin{bmatrix} R \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} ; \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} = \begin{bmatrix} R \end{bmatrix}^{-1} \begin{bmatrix} \hat{b_1} \\ \hat{b_2} \\ \hat{b_3} \end{bmatrix} ; \begin{bmatrix} R \end{bmatrix}^{-1} = \begin{bmatrix} R \end{bmatrix}^{T} \tag{2.4}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) \\ \cos(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \tag{2.5}$$

2.3.3 Angular velocities

Angular velocity is the time rate of change of direction of an object. Since the drone has three rotational movements (Roll, Pitch, Yaw), this means that it can be expressed as three angular speeds in the inertial reference frame. The total angular velocity can also be calculated according to the theory of summation of angular velocities, because it is complicated to calculate it directly from the body fixed reference frame relative to the inertial reference frame.[33]

- **Addition Theorem for Angular Velocities :**
  - Consider multiple reference frames :

$$R_1, R_2, ..., R_n$$

  - The following relation applies, whether the angular velocities are simple or not :

$$\vec{\omega}^{R \to R_n} = \vec{\omega}^{R \to R_1} + \vec{\omega}^{R_1 \to R_2} + ... + \vec{\omega}^{R_n - 1 \to R_n}$$

  - There exists at any one instant only one :

$$\vec{\omega}^{R \to R_n}$$

  - Also :

$$\vec{\omega}^{R \to R_n} = -\vec{\omega}^{R_n \to R}$$

  - And also, we assume the presence of two additional intermediate references to calculate the angular velocity of rotational movements relative to the inertial reference frame as shown in the figure below :

We conclude from the previous figure :

$R - R_1$, Yaw(ψ), $\hat{k} = \hat{k_1}$
$R_1 - R_2$, Pitch(θ), $\hat{j} = \hat{j_1}$
$R_2 - R_3$, Roll(ϕ), $\hat{i} = \hat{i_1}$

After applying the theorem of addition of angular velocities, we find :

$$\vec{\omega}^{R \to R_3} = \vec{\omega}^{R_2 \to R_3} + \vec{\omega}^{R_1 \to R_2} + \vec{\omega}^{R \to R_1} \tag{2.6}$$

$$\vec{\omega}^{R \to R_3} = \dot{\phi}\hat{i_3} + \dot{\theta}\hat{j_2} + \dot{\psi}\hat{k_1} \tag{2.7}$$

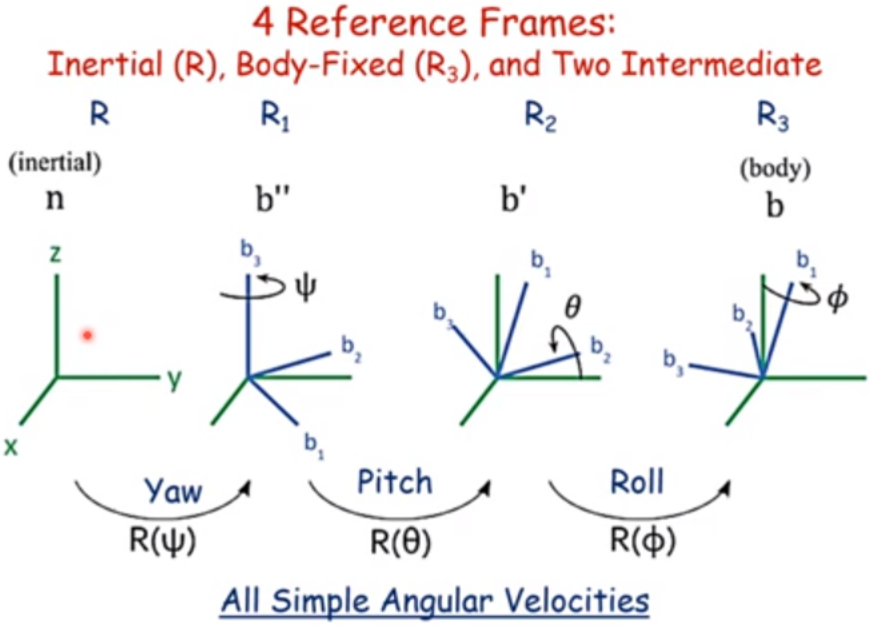


Figure 2.5: The 4 references frames[33]

Using the diagrams below, we can obtain the following matrices :

$$\begin{bmatrix} \hat{i} \\ \hat{j} \end{bmatrix} = \begin{bmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \hat{i}_1 \\ \hat{j}_1 \end{bmatrix} \tag{2.8}$$

$$\begin{bmatrix} \hat{k}_1 \\ \hat{i}_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} \hat{k}_2 \\ \hat{i}_2 \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} \hat{j}_2 \\ \hat{k}_2 \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \hat{j}_3 \\ \hat{k}_3 \end{bmatrix} \tag{2.10}$$

$$\begin{aligned} \vec{\omega}^{R\to R_3} &= \dot{\phi}\hat{i_3} + \dot{\theta}(\cos(\phi)\hat{j_3} - sin(\phi)\hat{k_3}) + \dot{\psi}(\cos(\theta)\hat{k_2} - sin(\theta)\hat{i_2}) \\ &= \dot{\phi}\hat{i_3} + \dot{\theta}(\cos(\phi)\hat{j_3} - \sin(\phi)\hat{k_3}) + \dot{\psi}\cos(\theta)(\sin(\phi)\hat{j_3} + \cos(\phi)\hat{k_3} - \dot{\psi}\sin(\theta)\hat{i_3}) \end{aligned}$$

$$\begin{aligned} \vec{\omega}^{R\to R_3} = \hat{i_3}(\dot{\phi} - \dot{\psi}\sin(\theta)) + \hat{j_3}(\dot{\theta}\cos(\phi) + \dot{\psi}\cos(\theta)\sin(\phi)) \\ + \hat{k_3}(-\dot{\theta}\sin(\phi) + \dot{\psi}\cos(\theta)\cos(\phi)) \end{aligned} \tag{2.11}$$

According to the preceding equation, we can calculate the angular velocity matrix in the body fixed reference frame.

$$\Omega = \begin{bmatrix} \omega_{roll} \\ \omega_{pitch} \\ \omega_{yaw} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{2.12}$$

To obtain the angular velocity in the inertial reference frame, this matrix must be inverted, but its inverse is not equivalent to the transpose matrix since it is not a rotational matrix.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan(\theta)\sin(\phi) & \tan(\theta)\cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \frac{\sin(\phi)}{\cos(\theta)} & \frac{\cos(\phi)}{\cos(\theta)} \end{bmatrix} \begin{bmatrix} \omega_{roll} \\ \omega_{pitch} \\ \omega_{yaw} \end{bmatrix} \tag{2.13}$$

In the case where the quadcopter makes angular movements of small amplitudes :

$$\begin{cases} \cos(\phi) = \cos(\theta) = \cos(\psi) = 1 \\ \sin(\phi) = \sin(\theta) = \sin(\psi) = 0 \end{cases}$$

So the angular velocity will be :

$$\Omega = \begin{bmatrix} \phi & \theta & \psi \end{bmatrix}^T \tag{2.14}$$

2.3.4 Linear velocities

The linear velocities in the body-fixed reference frame as a function of the linear velocities in the inertial reference frame are given by :

$$V = \begin{bmatrix} {\upsilon_x}^b \\ {\upsilon_y}^b \\ {\upsilon_z}^b \end{bmatrix} = R \times \begin{bmatrix} {\upsilon_x}^m \\ {\upsilon_y}^m \\ {\upsilon_z}^m \end{bmatrix} \tag{2.15}$$

While :

- R = The rotation matrix.

$$V = \begin{bmatrix} {\upsilon_x}^b \\ {\upsilon_y}^b \\ {\upsilon_z}^b \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} {\upsilon_x}^m \\ {\upsilon_y}^m \\ {\upsilon_z}^m \end{bmatrix} \tag{2.16}$$

2.3.5 The physical effects acting on the quadcopter

The forces

The forces affecting the system are :

- **Weight :** Weight is the force of attraction that the earth exerts on a body, and it is given by :

$$\begin{aligned} \vec{W} &= -W\hat{k} \\ W &= mg \end{aligned} \tag{2.17}$$

  While :
  - W : weight force.
  - m : the total masse.
  - g : the acceleration of gravity.

- **Thrust :** These are the forces generated by the motor's spinning. They are perpendicular to the propellers' plane. These forces are related to the square of the motor's rotational speed :

$$\begin{aligned} \vec{T_i} &= T_i\hat{b_3} \\ T_i &= k_p \times \omega_i^2 \\ T_i &= k_p \times (\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \end{aligned} \tag{2.18}$$

  While :
  - T_i : thrust forces.
  - ω_i : propellers speeds.
  - kp : lift factor.

The moments

There are various acting moments on the quadcopter. These moments are the result of thrust and drag forces, as well as gyroscopic effects.

- **Roll moment :** The rotation around the axis (x) is due to the moment created by the difference between the lift forces of rotors (2) and (4), which is described by the following equation :

$$\begin{aligned} \vec{\tau_\phi} &= dk_p(\omega_4^2 - \omega_2^2)\hat{b_1} \\ \tau_\phi &= dk_p(\omega_4^2 - \omega_2^2) \\ \tau_\phi &= d(F_4 - F_2) \end{aligned} \tag{2.19}$$

  While :

  - τ_ϕ : the moment applied in the roll motions.
  - d : the distance between rotor end the center of gravity.

- **Pitch moment :** The rotation around the axis (y) is due to the moment created by the difference between the lift forces of rotors (1) and (3), which is described by the following equation :

$$\begin{aligned} \vec{\tau_\theta} &= dk_p(\omega_3^2 - \omega_1^2)\hat{b_2} \\ \tau_\theta &= dk_p(\omega_3^2 - \omega_1^2) \\ \tau_\theta &= d(F_3 - F_1) \end{aligned} \tag{2.20}$$

  While :

  - τ_θ : the moment applied in the pitch motions.

- **Yaw moment :** Rotation around the axis (z) is due to a reactive torque induced by the drag torques in each propeller, which is described by the following relationship :

$$\begin{aligned} \vec{\tau_\psi} &= dk_p(\omega_4^2 + \omega_2^2 - \omega_1^2 - \omega_3^2)\hat{b_3} \\ \tau_\psi &= k_d(\omega_4^2 + \omega_2^2 - \omega_1^2 - \omega_3^2) \end{aligned} \tag{2.21}$$

  While :

  - τ_ψ : the moment applied in the yaw motions.
  - b : drag factor.

- **Aerodynamic friction moment :** It is given by :

$$\tau_a = k_{fa}\Omega^2 \tag{2.22}$$

While :

- k_{fa} : Coefficient of aerodynamic friction.
- Ω : Angular velocity.

Gyroscopic effect

The gyroscopic effect is described as the difficulty in changing a rotating mass's location or orientation in its plane of revolution.

- **Gyroscopic moment of propellers :** It is given by the following relation :

$$M_{gp} = \sum_{i=1}^{4} \Omega \wedge J_r \begin{bmatrix} 0 & 0 & (-1)^2\omega_i \end{bmatrix}^T \tag{2.23}$$

While :

- M_{gp} : the gyroscopic moment of propellers.
- J_r : the inertia of the rotors.

- **Gyroscopic moment due to quadcopter movements :** It is given by the following relation :

$$M_{gm} = \Omega \wedge J\Omega \tag{2.24}$$

While :

- M_{gm} : the gyroscopic moment due to quadcopter movements.
- J : the inertia of the system.

2.4 Development of the mathematical model according to Newton-Euler

After describing the various equations, we may construct the mathematical model using the Newton-Euler formulation. The equations are written in the following format :

$$\begin{cases} \sum \vec{F} = (F_f + F_t + F_g) = \dfrac{d}{dt}\vec{L} = m\dfrac{d\vec{v}^{R\to C}}{dt} \\ \sum \overrightarrow{\vec{M}} = (M_f - M_{gm} - M_{gh} - M_a) = \dfrac{d\overrightarrow{\vec{H}}}{dt} \end{cases} \tag{2.25}$$

While :

- ξ : the position vector of the quadcopter, such as :

$$\xi = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{2.26}$$

- m : the total mass of the quadcopter.
- J : symmetrical inertia matrix of dimension (3x3), it is given by :

$$J = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \tag{2.27}$$

- Ω : the angular velocity in the inertial reference frame.
- R : the rotation matrix.
- $\wedge$: the vector product.
- F_f : the total force generated by the four rotors is given by :

$$\begin{aligned} F_f &= R \times \begin{bmatrix} 0 & 0 & \sum_{i=1}^{4} F_i \end{bmatrix}^T \\ F_i &= k_p \omega_i^2 \end{aligned} \tag{2.28}$$

- F_t : the drag force along the axes (x, y, z), it is given by :

$$F_t = \begin{bmatrix} -k_{ftx} & 0 & 0 \\ 0 & -k_{fty} & 0 \\ 0 & 0 & -k_{ftz} \end{bmatrix} \times \dot{\xi} \tag{2.29}$$

- k_{ftx}, k_{fty}, k_{ftz}: Translation drag coefficients.
- F_g: force of gravity, it is given by:

$$F_g = \begin{bmatrix} 0 & 0 & -mg \end{bmatrix} \tag{2.30}$$

- M_f: a moment caused by thrust and drag forces.

$$M_f = \begin{bmatrix} d(F_4 - F_2) & d(F_3 - F_1) & k_d(\omega_4^2 + \omega_2^2 - \omega_1^2 - \omega_3^2) \end{bmatrix} \tag{2.31}$$

- M_a: a moment resulting from aerodynamic friction, it is given by:

$$M_a = \begin{bmatrix} k_{fax}\dot{\phi}^2 & k_{fay}\dot{\theta}^2 & k_{faz}\dot{\psi}^2 \end{bmatrix} \tag{2.32}$$

- k_{fax}, k_{fay}, k_{faz}: aerodynamic friction coefficients.
- M_{gp}: gyroscopic moment of propellers, and it is given by:

$$M_{gp} = \sum_{i=1}^{4} \Omega \wedge J_r \begin{bmatrix} 0 & 0 & (-1)^2\omega_i \end{bmatrix}^T \tag{2.33}$$

- J_r : the rotor inertia.
- M_{gm}: gyroscopic moment due to quadcopter movements.

$$M_{gm} : \Omega \wedge J\Omega \tag{2.34}$$

2.4.1 Translational equations of motion

We have :

$$m\ddot{\xi} = F_f + F_t + F_g \tag{2.35}$$

And :

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} \sum_{i=1}^{4} F_i - \begin{bmatrix} k_{ftx}\dot{x} \\ k_{fty}\dot{y} \\ k_{ftz}\dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.36}$$

We then get the differential equations that describe the translation movement :

$$\begin{cases} \ddot{x} = \dfrac{1}{m} + (\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\psi)\sin(\phi))(\sum\limits_{i=1}^{4} F_i) - \dfrac{k_{ftx}}{m}\dot{x} \\ \ddot{y} = \dfrac{1}{m} + (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\sin(\psi))(\sum\limits_{i=1}^{4} F_i) - \dfrac{k_{fty}}{m}\dot{y} \\ \ddot{z} = \dfrac{1}{m} + (\cos(\phi)\cos(\theta))(\sum\limits_{i=1}^{4} F_i) - \dfrac{k_{ftz}}{m}\dot{z} - g \end{cases} \tag{2.37}$$

2.4.2 Rotational Equations of Motion

We have :

$$J\dot{\Omega} = M_{gm} + M_f - M_a - M_{gh} \tag{2.38}$$

When we replace every single moment with its matching formula, we get :

$$\begin{aligned}\begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}\begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} &= -\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \wedge \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}\right) \\ &+ \begin{bmatrix} db(\omega_4^2 - \omega_2^2) \\ db(\omega_3^2 - \omega_1^2) \\ k_d(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{bmatrix} - \begin{bmatrix} k_{fax}\dot{\phi}^2 \\ k_{fay}\dot{\theta}^2 \\ k_{faz}\dot{\psi}^2 \end{bmatrix} - \begin{bmatrix} J_r\Omega_r\dot{\theta} \\ -J_r\Omega_r\dot{\phi} \\ 0 \end{bmatrix}\end{aligned} \tag{2.39}$$

We then get the differential equations that define the rotating movement :

$$\begin{cases} I_x\ddot{\phi} = -\dot{\theta}\dot{\psi}(I_z - I_y) - k_{fax}\dot{\phi}^2 - J_r\Omega_r\dot{\theta} + db(\omega_4^2 - \omega_2^2) \\ I_y\ddot{\theta} = -\dot{\phi}\dot{\psi}(I_z - I_x) - k_{fay}\dot{\theta}^2 - J_r\Omega_r\dot{\phi} + db(\omega_3^2 - \omega_1^2) \\ I_z\ddot{\psi} = -\dot{\phi}\dot{\theta}(I_y - I_x) - k_{faz}\dot{\psi}^2 - db(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2) \end{cases} \tag{2.40}$$

The control inputs are u1, u2, u3, and u4 of the system and are represented as a function of the angular speeds of the four rotors, as follows :

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} k_p & k_p & k_p & k_p \\ 0 & -dk_p & 0 & dk_p \\ -dk_p & 0 & dk_p & 0 \\ k_d & -k_d & k_d & -k_d \end{bmatrix}\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \tag{2.41}$$

While the translation movement is along the "x, y, z" axis, the rotation movement is along the yaw.
And :

$$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4 \tag{2.42}$$

As a result, the quadcopter's overall dynamic model is as follows :

$$\begin{cases} \ddot{\phi} = \dfrac{(I_y - I_z)}{I_x}\dot{\theta}\dot{\psi} - \dfrac{J_r}{I_x}\Omega_r\dot{\theta} - \dfrac{k_{fax}}{I_x}\dot{\phi}^2 + \dfrac{d}{I_x}u_2 \\ \ddot{\theta} = \dfrac{(I_z - I_x)}{I_y}\dot{\phi}\dot{\psi} - \dfrac{J_r}{I_y}\Omega_r\dot{\phi} - \dfrac{k_{fay}}{I_y}\dot{\theta}^2 + \dfrac{d}{I_y}u_3 \\ \ddot{\psi} = \dfrac{(I_x - I_y)}{I_z}\dot{\theta}\dot{\phi} - \dfrac{k_{faz}}{I_z}\dot{\psi}^2 + \dfrac{1}{I_z}u_4 \\ \ddot{x} = -\dfrac{k_{ftx}}{m}\dot{x} + \dfrac{1}{m}u_x u_1 \\ \ddot{y} = -\dfrac{k_{fty}}{m}\dot{y} + \dfrac{1}{m}u_y u_1 \\ \ddot{z} = -\dfrac{k_{ftz}}{m}\dot{z} - g + \dfrac{\cos(\phi)\cos(\psi)}{m}u_1 \end{cases} \tag{2.43}$$

While :

$$\begin{cases} u_x = (\cos(\phi)\cos(\psi)\sin(\theta) + \sin(\phi)\sin(\psi)) \\ u_y = (\cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi)) \end{cases} \tag{2.44}$$

2.4.3 Rotor dynamics

The dynamic equations for the rotor of a brushless DC motor are as follows :

$$\begin{cases} V = ri + L\dfrac{di}{dt} + k_e\omega \\ k_m i = J_r\dfrac{d\omega}{dt} + C_s + k_r\omega^2 \end{cases} \tag{2.45}$$

While :

- V : The motor input voltage.
- ω_i : The rotor angular speed.
- k_e, k_m : The constants of electrical and mechanical torques.
- k_r : The load torque constant.
- r, L : The resistance and inductance of the motor.
- J_r : The rotor inertia.
- C_s : The dry friction.

$$\dot{\omega_1} = bV_1 - \beta_0 - \beta_1\omega_i - \beta_2\omega_i^2,\ i\epsilon[1,4] \tag{2.46}$$

And :

$$\beta_0 = \frac{C_s}{J_r},\ \beta_1 = \frac{k_e k_m}{rJ_r},\ \beta_2 = \frac{k_r}{J_r},\ b = \frac{k_m}{rJ_r} \tag{2.47}$$

2.5 System state representation

To represent the state of our system, we consider X as the state vector of the system :

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} & z & \dot{z} \end{bmatrix} \tag{2.48}$$

The output vector :

$$Y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{bmatrix}^T = \begin{bmatrix} \phi & \theta & \psi & x & y & z \end{bmatrix}^T \tag{2.49}$$

The input vector :

$$U = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T \tag{2.50}$$

The dynamic equations provide us with the representation in the following state space :

$$\begin{cases} \dot{X} = f(X, Y) \\ Y = h(X) \end{cases} \tag{2.51}$$

$$\begin{pmatrix}
\dot{\phi} & = \dot{x_1} = x_2 \\
\ddot{\phi} & = \dot{x_2} = a_1 x_4 x_6 + a_2 x_2^2 + a_3 \Omega_r x_4 + b_1 u_2 \\
\dot{\theta} & = \dot{x_3} = x_4 \\
\ddot{\theta} & = \dot{x_4} = a_4 x_2 x_6 + a_2 x_4^2 + a_6 \Omega_r x_2 + b_2 u_3 \\
\dot{\psi} & = \dot{x_5} = x_6 \\
\ddot{\psi} & = \dot{x_6} = a_7 x_2 x_4 + a_8 x_6^2 + b_3 u_4 \\
\dot{x} & = \dot{x_7} = x_8 \\
\ddot{x} & = \dot{x_8} = a_9 x_8 + \frac{u_x}{m} u_1 \\
\dot{y} & = \dot{x_9} = x_{10} \\
\ddot{y} & = \dot{x_{10}} = a_{10} x_{10} + \frac{u_y}{m} u_1 \\
\dot{z} & = \dot{x_{11}} = x_1 2 \\
\ddot{z} & = \dot{x_{12}} = a_{11} x_{12} - g + \frac{\cos(x_1)\cos(x_3)}{m} u_1
\end{pmatrix} \tag{2.52}$$

$$\begin{pmatrix} u_x &= (\cos(x_1)\sin(x_3)\cos(x_5) + \sin(x_1)\sin(x_5)) \\ u_y &= (\cos(x_1)\sin(x_3)\sin(x_5) - \sin(x_1)\cos(x_5)) \\ a_1 &= \frac{(I_y - I_z)}{I_x} \\ a_2 &= -\frac{k_{fax}}{I_x} \\ a_3 &= -\frac{J_r}{I_x} \\ a_4 &= \frac{(I_z - I_x)}{I_y} \\ a_5 &= -\frac{k_{fay}}{I_y} \\ a_6 &= -\frac{J_r}{I_y} \\ a_7 &= \frac{(I_x - I_y)}{I_z} \\ a_8 &= -\frac{k_{faz}}{I_z} \\ a_9 &= -\frac{k_{ftx}}{m} \\ a_{10} &= -\frac{k_{fty}}{m} \\ a_{11} &= -\frac{k_{ftz}}{m} \\ b_1 &= \frac{d}{I_x} \\ b_2 &= \frac{d}{I_y} \\ b_3 &= \frac{1}{I_z} \end{pmatrix} \tag{2.53}$$

2.6 The PID controller

2.6.1 Definition

The PID controller continually monitors the error value and uses it to calculate the proportional, integral, and derivative values. The controller then sums these three numbers to generate the output. To construct the output, we will combine the proportional, integral, and derivative equations. We may state that the PID comprises:

1. **Gains :**
   The term "gain" refers to the "multiplication factor". By modifying the gain settings (proportional gain "K_P", integral gain "K_i" and derivative gain "K_d"), the user may regulate how much influence the PID controller has on the output and how the controller responds to differ-

ent changes in the process value.

2. **PID terms :**
   - **P - Proportional :**
     The proportional is determined by multiplying the P-Gain by the error. The proportional function's objective is to produce a large instantaneous reaction on the output in order to get the process value closer to the desired value. The proportionate value has less of an impact on the outcome as the mistake decreases, such us :

$$\varepsilon = D_v - P_v \tag{2.54}$$

$$P = K_p \times \varepsilon \tag{2.55}$$

     While :
     – P : The proportional term.
     – K_p : The proportional gain.
     – ε : The error.
     – D_v : The desired value.
     – P_v : The process value.

   - **I - Integral :**
     The integral is calculated by multiplying the I-Gain by the error, then multiplying by the controller's cycle time (the frequency with which the PID calculation is performed), and continually collecting this result as the "total integral".

     The newly calculated integral value is added to the integral total. The integral will often have less immediate influence on the output than the proportional, but because the integral accumulates over time, the longer it takes for the process value to reach the set point, the greater the effect of the integral on the output.

     This helps to reduce the system's steady-state inaccuracy. The integral factor integrates the error term until it equals '0', and the integral math is:

$$\varepsilon = D_v - P_v \tag{2.56}$$

$$I = K_i \times \int_0^t \varepsilon dt \tag{2.57}$$

$$I_t = I_t + I \tag{2.58}$$

     While :

– I : The integral term.
– I_t : The total integral term.
– K_i : The integral gain.
– ε : The error.
– D_v : The desired value.
– P_v : The process value.
– dt : The cycle time of the controller.

- **D - Derivative :**
To determine the derivative, multiply the D-Gain by the process value's ramp rate. The derivative's objective is to "predict" where the process value will go, if the process value approaches the desired value too quickly, the derivative will restrict the output to keep the process value from exceeding the set point, and the derivative math is:

$$\varepsilon = D_v - P_v \tag{2.59}$$

$$D = K_d \times \left(\frac{d\varepsilon(t)}{dt}\right) = K_d \times \left(\frac{\varepsilon - \varepsilon_{prev}}{dt}\right) \tag{2.60}$$

While :
– D : The derivative term.
– K_d : The derivative gain.
– ε : The error.
– ε_{prev} : The previous error.
– D_v : The desired value.
– P_v : The process value.
– dt : The cycle time of the controller.

3. **Output :**
The PID controller output is calculated by adding the proportional, integral, and derivative. The amount of influence these three parameters have on the output is determined by the gain setting for each, this means that the output is:

$$U(t) = K_p + I_t + K_d \tag{2.61}$$

While :

- U(t) : The output.
- K_p : The proportional gain.
- I_t : The total integral.
- K_d : The derivative gain.

2.6.2 Work principle

In manual PID controller operation, the operator sometimes analyzes process factors and adjusts control variables. This adjustment results in the control variables being managed to define limitations, such as motor inputs, heating components, flow valves, and others. In contrast, PID devices operate automatically with continual analysis and modifications. Typically, automated PID controllers are classified as closed-loop systems, resulting in one or more controlling actions that consist of: "proportional, integral, and derivative".

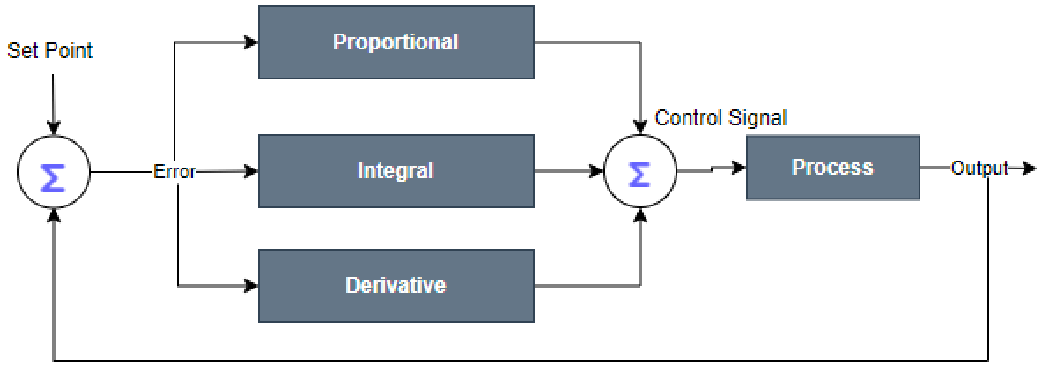


Figure 2.6: PID controller block diagram

2.7 Conclusion

This chapter provides the reader with an overview of the drone and its operational principles. We used the Newton-Euler mathematical modeling of the quadcopter, after making certain simplifying assumptions. This designed model takes into account all of the phenomena affecting the quadcopter. And how to use a PID controller to improve the stability of a quadcopter.

CHAPTER 3

Computer vision on drone

3.1 Introduction

Due to the increasing use of unmanned aerial vehicles (UAVs), the need to develop drones in the skies, especially in urban environments, is greater than ever. One of the most important developments that have been applied to the drone is computer vision, such as recognizing and classifying objects, tracking targets, controlling them through hand gestures, and others. There are many computer vision algorithms for object detection, whether they are machine learning algorithms such as the Haar-cascade algorithm or deep learning algorithms such as the YOLO (You Only Look Once) algorithm, the SSD (Single Short Detection) algorithm, and RCNN (Region-based Convolutional Neural Network) algorithm, etc...

In this chapter, we will focus on the YOLO algorithm for detecting objects on our quadcopter. Since this method processes images more quickly and accurately, it is often regarded as the best for object detection.

3.2 Related Works

Computer vision has various uses, and below are just a few examples of the works where it has been used with drones:

- **Design and Development of an Interactive Autonomous Monitoring and Surveillance Drone :** The purpose of this drone project was to make it capable of comprehending gesture-based human commands. An object recognition system that is based on vision is created and integrated into the on-board computer. It primarily accomplishes the purpose of identifying and recognising license plates.[34]

- **Person Identification using Autonomous Drone through Resource Constraint Devices :** This drone was created with the ability to distinguish a particular individual from a crowd by fusing sophisticated hardware components with cutting-edge algorithms.[35]

- **Face recognition drone :** The primary objective of the project is to use the drone's camera to identify a person's face and upload all of that person's personal data to a server room. It is helpful for military operations in far-off places, biometric presence, and surveillance.[6]

3.3 Object detection on drones

3.3.1 Definition of object detection :

The process of recognizing an item that is visible in images and videos is called object recognition. It is among the most significant uses of deep learning and machine learning. Teaching robots to comprehend (recognize) picture material in the same way that humans do is the aim of this field.

The tasks of item localization and image classification are combined in object detection algorithms. It takes an image as input and outputs one or more bounding boxes with the class label attached to each. These techniques are robust enough to handle multi-class classification, localization, and objects with multiple occurrences.

A basic illustration of this can be seen in CAPTCHA images. In contrast, object detection distinguishes specific objects within an image based on predefined labels. For example, it finds every road sign in the image and classifies it along with other objects such as people and cars.

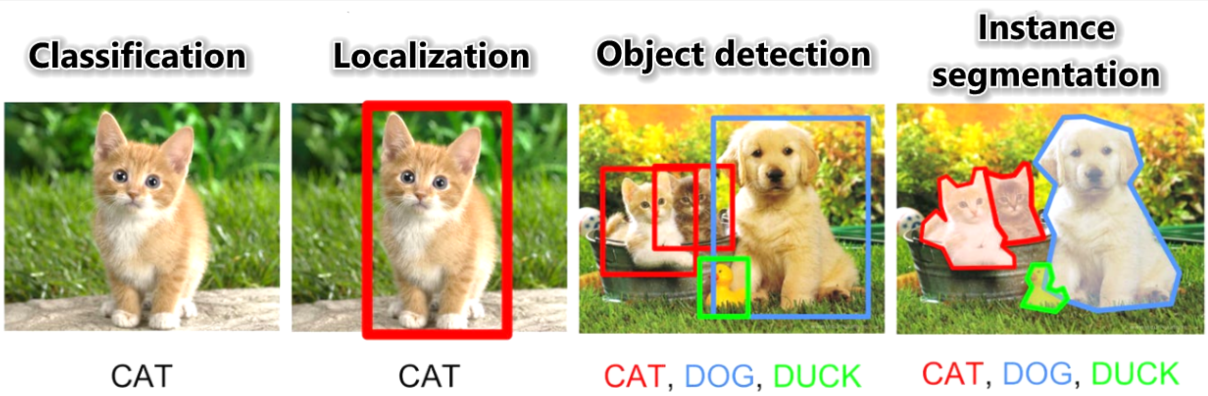


Figure 3.1: Difference between classification. Localization and Detection[36]

3.3.2 Algorithms for object detection :

A computer vision approach called object detection helps to recognize and categorize certain objects in a given environment. Identifying instances of each item, separating them, and analyzing their features—which are essential for making real-time predictions—is the primary objective of object detection, which involves scanning digital photos or real-world settings. In this case, artificial intelligence (AI) is frequently used, which is used to categorize robots that simulate human intellect and cognitive processes including learning, problem-solving, speech and face recognition, and decision-making, among others. Machine learning and deep learning are the most popular methods for detecting objects.[37]

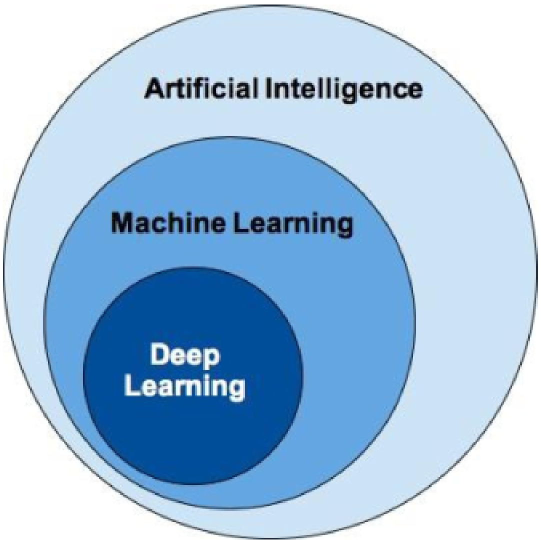


Figure 3.2: Relationship of the concepts of artificial intelligence, learning and deep learning machine.[38]

Support vector machines (SVMs) are used in combination with both of these techniques to extract features, train algorithms, and classify objects. The degree of efficacy and industry applicability of various object detecting techniques varies.

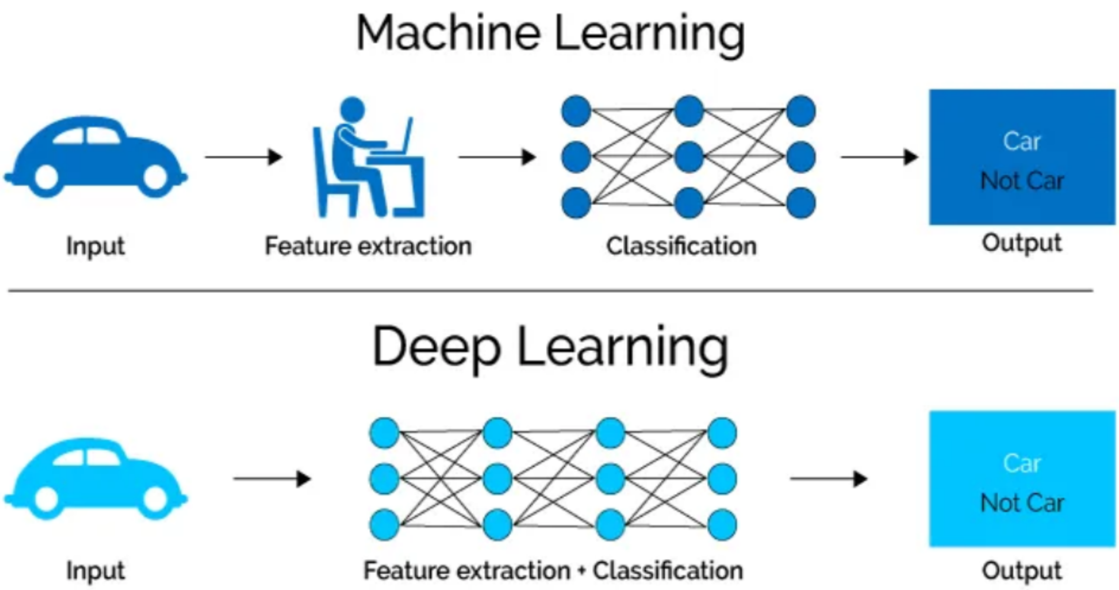


Figure 3.3: Difference between machine learning and deep learning[39]

Let's examine this in more detail :

- **Machine learning :**

  The advantage of utilizing a machine learning method for object detection is that, unlike automatic training data, it uses user entered data for categorization. As a result, the method is more stable and less prone to errors overall. Deep learning procedures include autonomous feature selection tailored to your tech stack, whereas machine learning models are based on human feature selection. Such as :

  – Aggregate channel features (ACF).
  – Deformable parts model (DPM).
  – Bag of features model.
  – Viola-Jones algorithm.
  – and others...

- **Deep learning :**

  The convolutional neural networks models, one type of deep learning technique, generate object predictions more quickly and accurately. Naturally, for it to occur, you'll need a more powerful GPU and more datasets!. There are many different object detection tasks that employ deep learning. Neural networks enable modern video security cameras and monitoring systems to correctly identify things or faces that are unfamiliar to them. Deep learning-based object recognition models are separated into two stages:

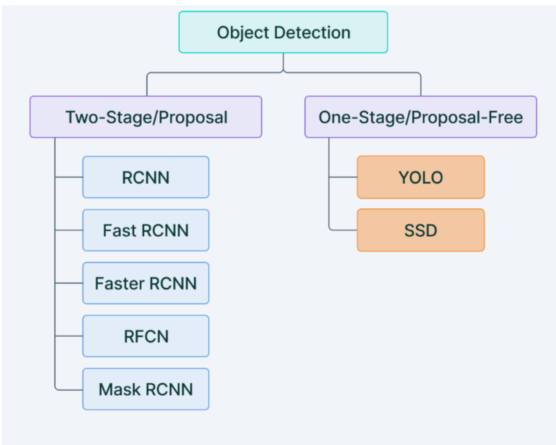


Figure 3.4: Object detection algorithms categories[40]

3.3.3 Convolutional neural network architecture :

Definition of convolutional neural network :

A computational model called an convolutional Neural Network (CNN) is based on the neural architecture of the human brain. It is made up of layers of linked nodes, or artificial neurons. These nodes process information, and during training, the network modifies the connection strengths (weights) to learn from the input. This allows the network to identify patterns, anticipate outcomes, and perform a variety of machine learning and artificial intelligence tasks. It might consist of one layer or several layers of linked nodes (neurons) arranged in layers.[41]

The term ‘Convolution” in CNN denotes the mathematical function of convolution which is a special kind of linear operation wherein two functions are multiplied to produce a third function which expresses how the shape of one function is modified by the other. In simple terms, two images which can be represented as matrices are multiplied to give an output that is used to extract features from the image.

An input layer, one or more hidden layers, and an output layer are the node layers that make up neural networks. Every node has a weight and a threshold value, which is referred to as the neuron’s activation. More precisely, a value in the range between (0.0) and (1.0), and the image you might have in your mind is that each neuron is lit up when its activation is a high number. and they are all connected artificial neurons. One node is activated and transfers data to the next layer of the network when its output exceeds the threshold value. If it is less than the cutoff, no information is sent.[42]

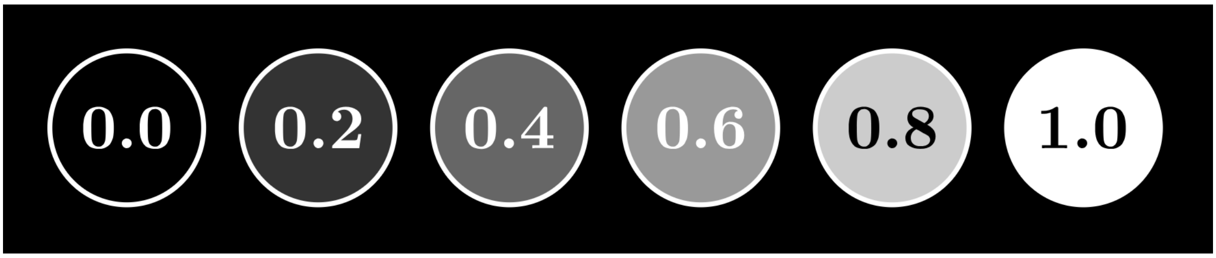


Figure 3.5: The numbers of activation of each neuron

For example, we have a dedicated neural network to identify handwritten numbers on images with ($28 \times 28 = 784$ pixels).[42]

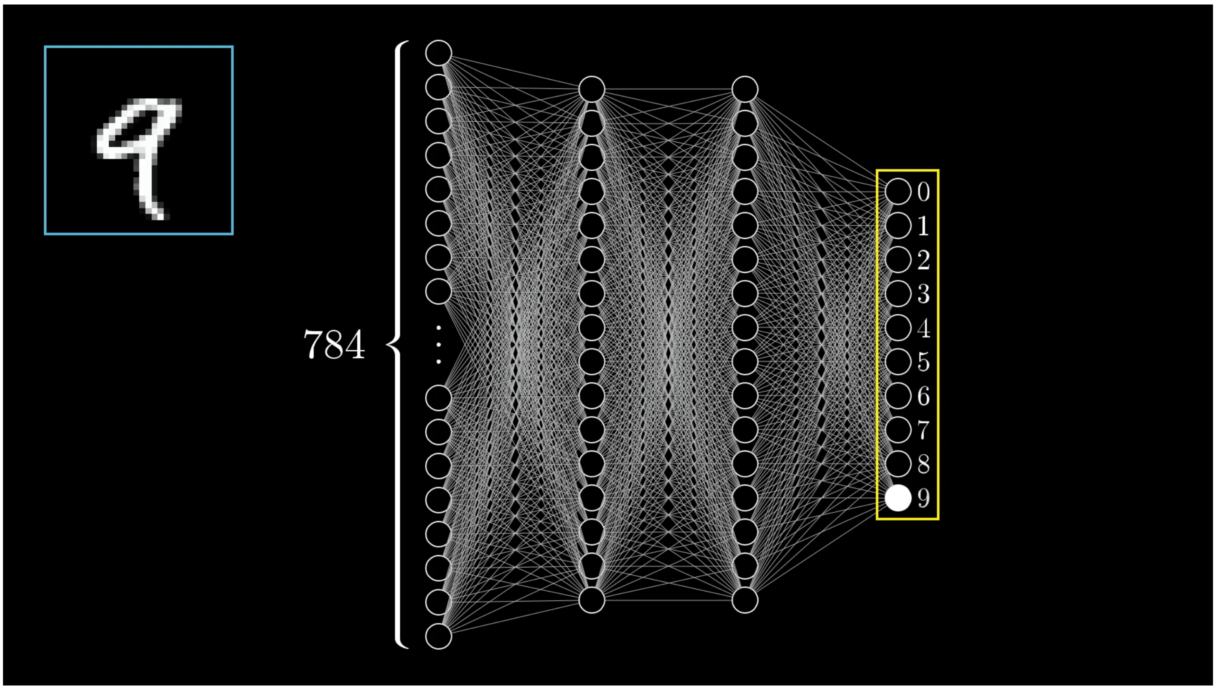


Figure 3.6: Convolutional neural network example for detect digits

Convolutional neural network layers :

A CNN architecture consists of two primary components, which are:[43]

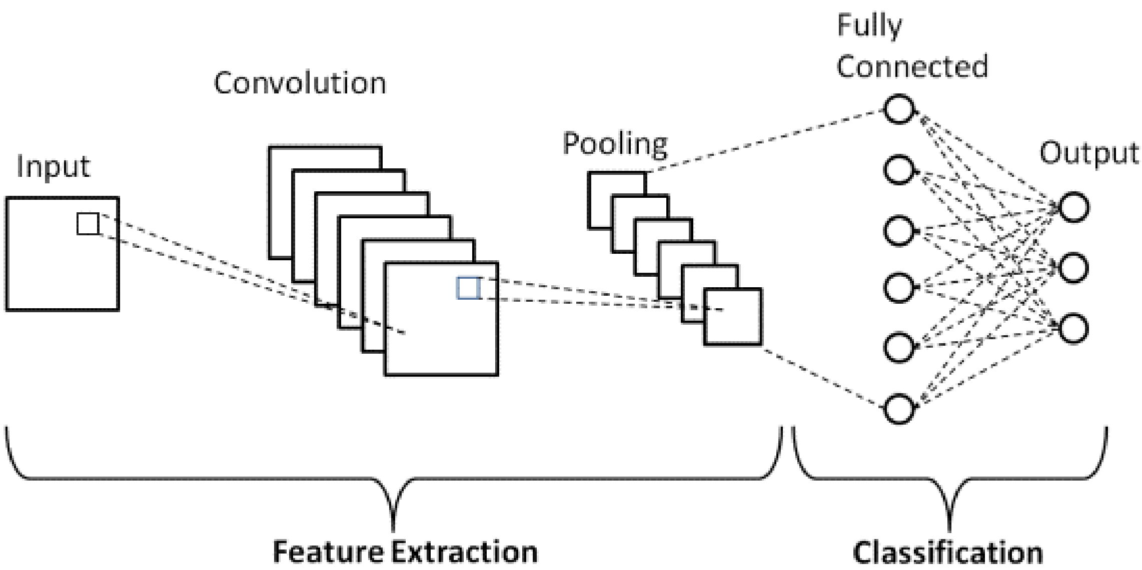


Figure 3.7: The CNN architecture diagram

The goal of this CNN feature extraction model is to minimize the amount of features in a dataset. It generates new features that are a summary of the preexisting features found in the first feature collection. The CNN architecture diagram shown below illustrates the several layers that make up CNN. We will explain each layer in detail.

1. **The convolutional layers :**
   The different characteristics from the input images are initially extracted using this layer. This layer performs the convolutional mathematical process between an input image and a filter of a specified size (MxM). The dot product between the filter and the portions of the input image with regard to the filter's size (MxM) is calculated by sliding the filter over the image.
   The output, known as the feature map, provides us with details about the image, including its borders and corners. This feature map is later supplied to other layers so that they may identify more features from the input image.
   After applying the convolution operation to the input, the CNN convolution layer transfers the output to the following layer (pooling layer). CNN's convolutional layers are very advantageous since they guarantee that the pixels' spatial link remains preserved.

2. **The pooling layers :**
   A Pooling Layer usually comes after a Convolutional Layer. This layer's main goal is to minimize the convolved feature map's size in order to save computing expenses. Reducing the connections between layers and performing separate operations on every feature map is how this is accomplished. There are several kinds of pooling processes, depending on the technique employed. It essentially provides an overview of the characteristics produced by a convolution layer.
   The biggest element in a feature map is used in max pooling. The average of the components in an Image segment of a certain size is determined by average pooling. Sum Pooling computes the total sum of the components in the designated section.
   Typically, the Pooling Layer acts as a link between the FC Layer and the Convolutional Layer.

3. **The fully connected layers :**
   The neurons are connected between two separate layers by the Fully Connected (FC) layer, which also includes the weights and biases. These layers make up the final few levels of a CNN architecture and are often positioned before the output layer.

This involves flattening and feeding the input image to the FC layer from the earlier layers. The mathematical function operations often occur in a few additional FC levels after the vector has been flattened. At this point, the process of categorization starts. Two completely linked layers will outperform a single connected layer, which is why two layers are connected. In CNN, these layers reduce the need for human supervision.

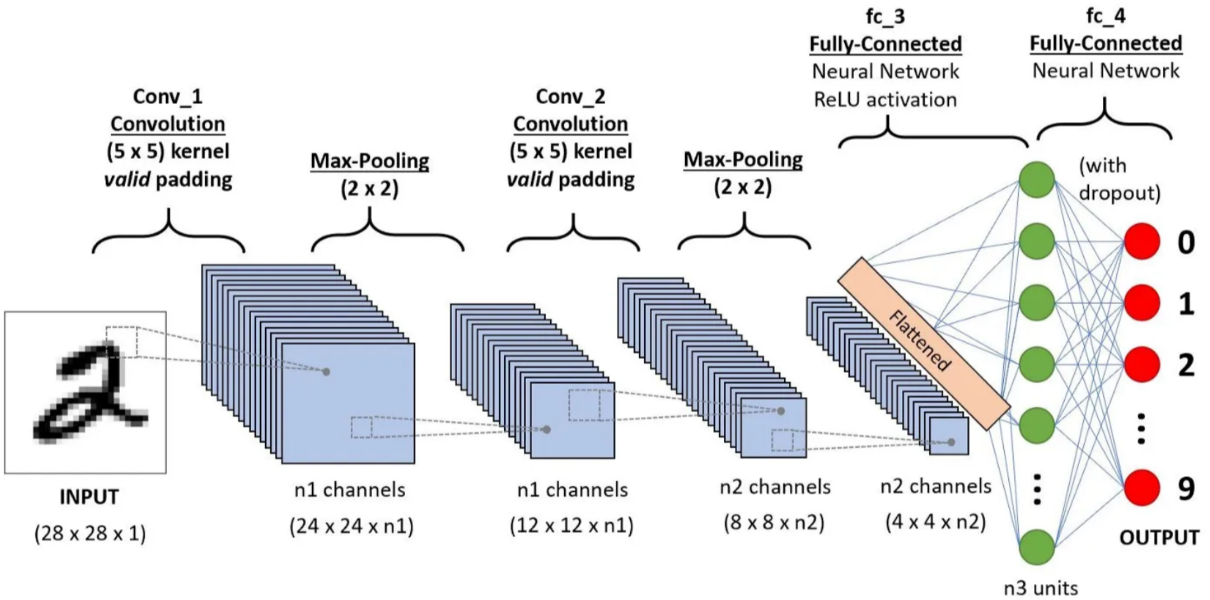


Figure 3.8: Convolutional neural network layers[44]

Weights and biases in neural networks :

Each layer's neurons have the following weights and biases :[45]

- **Weights :**
  Weights are numerical values associated with the connections between neurons. They determine the strength of these connections. The impact that the output of one neuron has on the input of another. Consider weights to be the coefficients that modify the effect of incoming information. They have the power to elevate or decrease the significance of particular data.
  A positive weight indicates that the neuron in the second layer should be on if the neuron in the first layer is on, while a negative weight indicates that the neuron in the second layer should be off.

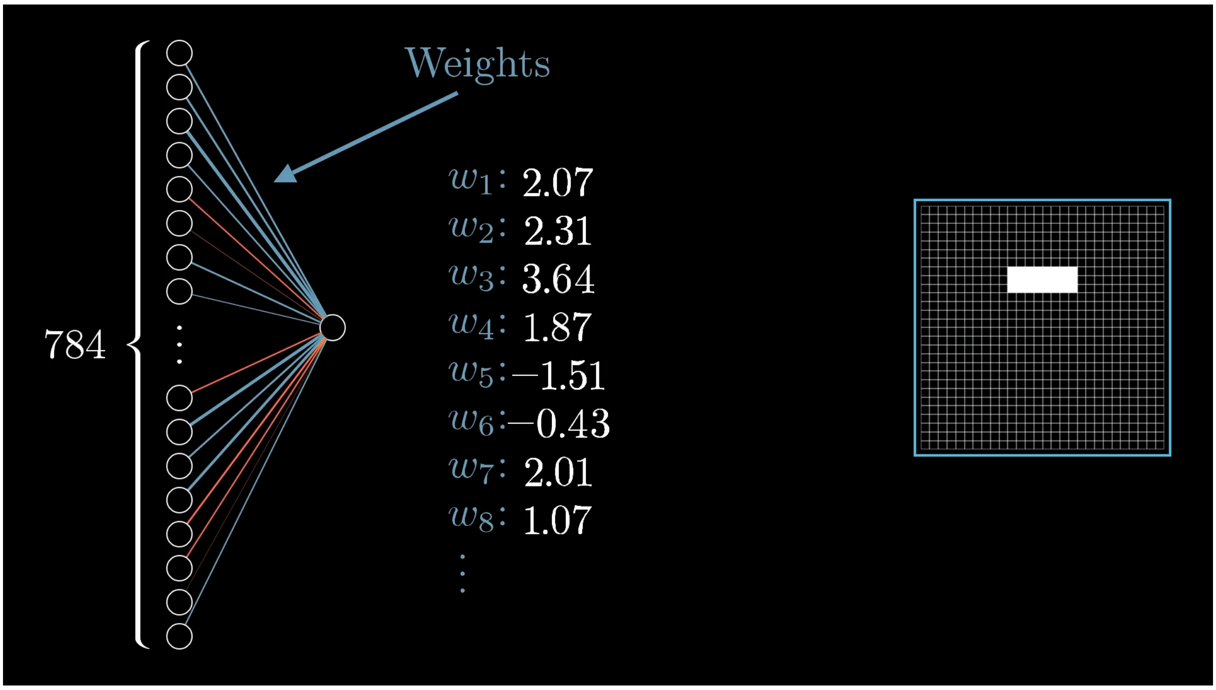


Figure 3.9: Weights of one neuron

Therefore, you take all of the activations from the neurons in the first layer and compute their weighted total to really determine the value of this second-layer neuron.

$$\omega_1 a_1 + \omega_2 a_2 + \omega_2 a_2 + \ldots + \omega_n a_n \tag{3.1}$$

- **biases :**
  Biases provide neural networks an essential extra degree of adaptability. In essence, biases are constants connected to every neuron. Biases, in contrast to weights, are added to the neuron's output rather than being linked to any particular input. Biases serve as a form of offset or threshold, allowing neurons to activate even when the weighted sum of their inputs is not sufficient on its own. They introduce a level of adaptability that ensures the network can learn and make predictions effectively.

The activation functions in CNN :

An activation function is introduced to the neural network to enable non-linear expression, which improves accuracy and helps the network better fit the results. However, the performance of different activation functions varies across neural networks. Among the most important functions of activation in neural networks are:[46]

- **ReLU function :**
  ReLU is the activation function that eliminates negative values; an equivalent function max(0, X), can be used. Anything on the x-axis' negative side gets trimmed to zero when using this function.[41]

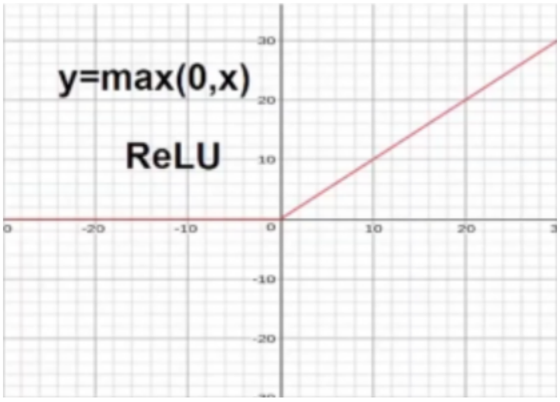


Figure 3.10: The ReLU function

- **Sigmoid function :**
  The range of the combined inputs is converted by this function to a range between 0 and 1. The sigmoid function, for instance, will limit an infinite range to a number between 0 and 1, assuming the output is from minus infinity to infinity, as depicted by the x-axis.[41]

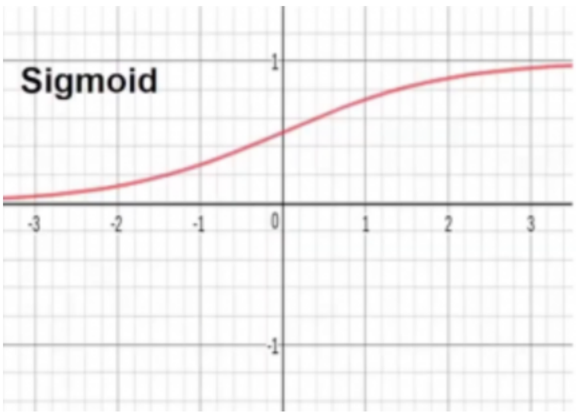


Figure 3.11: The Sigmoid function

- **tanh function :**
  By using this function, the range of the combined inputs is changed to fall between (-1) and (1). Tanh's form is very similar to that of the sigmoid, but its range is limited between (-1) and (1).[41]

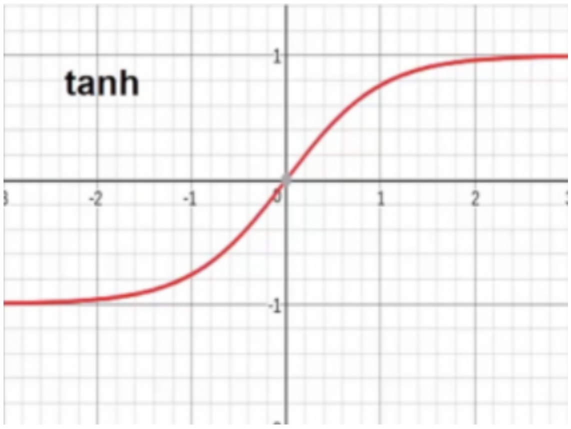


Figure 3.12: The tanh function

The learning process in neural network :

In neural networks, learning processing could propagate in two different ways, which are :

- **Forward propagation :**
  The first stage of turning incoming data into an output or prediction using a neural network is called forward propagation or feed forward. This is how it operates:[45]

  – **Input layer :** The neural network's input layer receives the input data.

  – **Weighted sum :** Each neuron in the subsequent layers calculates a weighted sum of the inputs it receives,with the weights being the changeable parameters.

  – **Adding biases :** The bias connected to every neuron is added to this weighted total. This presents an activation offset or threshold.

  – **Activation function :** An activation function is applied to the outcome of the weighted sum with bias. Based on the computed value, this function decides whether the neuron should fire or stay inactive.

  - **Propagation :** The output of one layer becomes the input for the next layer, and the process repeats until the final layer produces the network's prediction.

- **Backward propagation :**
  After the network has produced a prediction, it is critical to evaluate its accuracy and make any necessary modifications to enhance subsequent forecasts. Backward propagation is used in this situation:

  - **Error computation :** The network's prediction is contrasted with the real goal. The resultant error assesses the difference between the prediction and reality and is sometimes expressed as a loss or cost.

  - **Gradient descent :** Reducing this error is necessary for backward propagation. The network computes the gradient of the error with respect to the weights and biases in order to do this. The sharpest decline in error is indicated by this gradient.

  - **Weight and bias updates :** The network uses this gradient information to update the weights and biases throughout the network. The goal is to find the values that minimize the error.

  - **Iterative procedure :** On batches of training data, this forward and backward propagation procedure is repeated repeatedly. The weights and biases of the network approach values that minimize the error with each iteration.

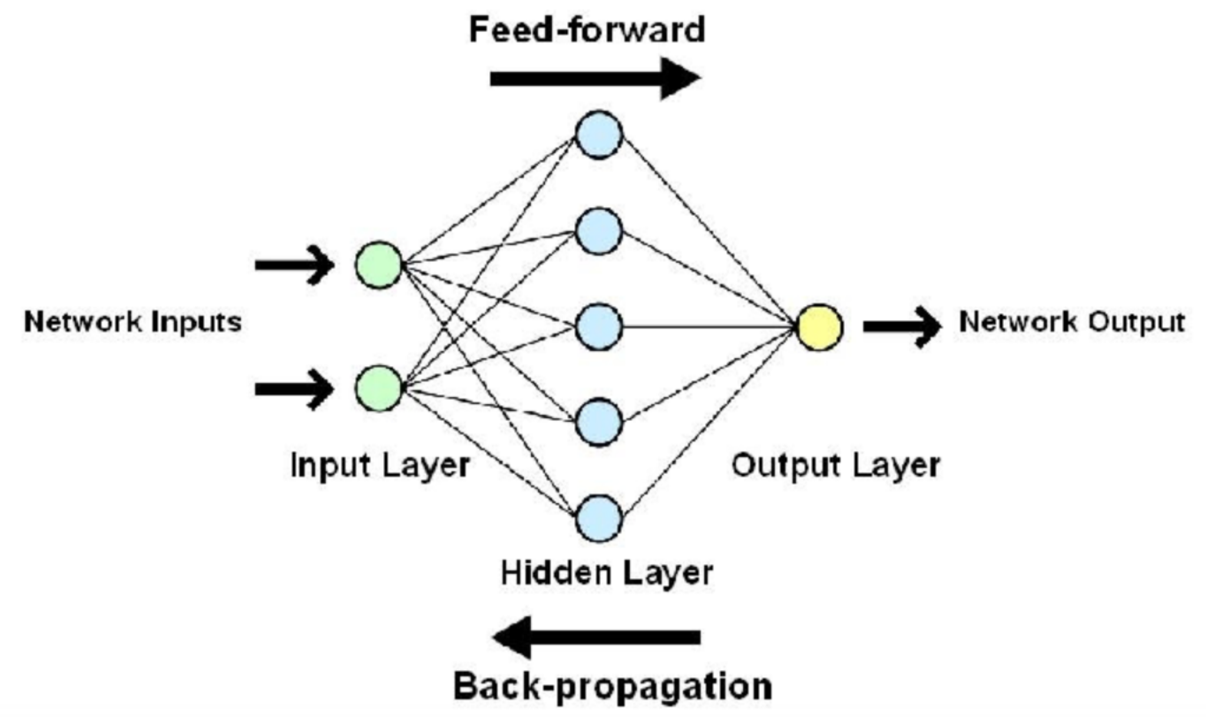


Figure 3.13: The learning process in a simple neural network.[41]

3.3.4 YOLO algorithm :

YOLO definition :

You Only Look Once (YOLO) is a state-of-the-art, real-time object detection algorithm introduced in 2015 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi in their famous research paper "You Only Look Once: Unified, Real-Time Object Detection".[47]
By utilizing a single convolutional neural network (CNN) to associate probabilities with each detected image and spatially separated bounding boxes, the authors reframe the object identification problem as a regression problem rather than a classification task.

YOLO characteristics :

- **Speed :**
  Yolo is really quick, as it doesn't deal with complex pipelines. It has a 45 frames per second (FPS) processing speed for images. Furthermore, when compared to other real-time systems, YOLO achieves more than double the mean average precision (mAP), making it an excellent choice for real-time processing.[48]

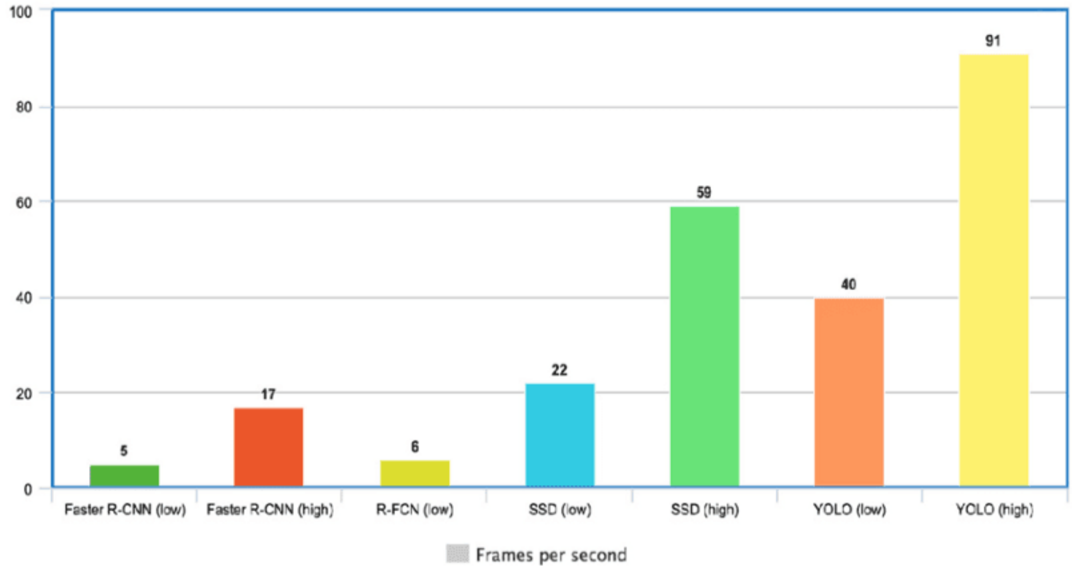


Figure 3.14: YOLO speed compared to other state-of-the-art object detectors

- **Detection accuracy :**
  YOLO has extremely few background mistakes and a significantly higher accuracy than other state-of-the-art models.[38]

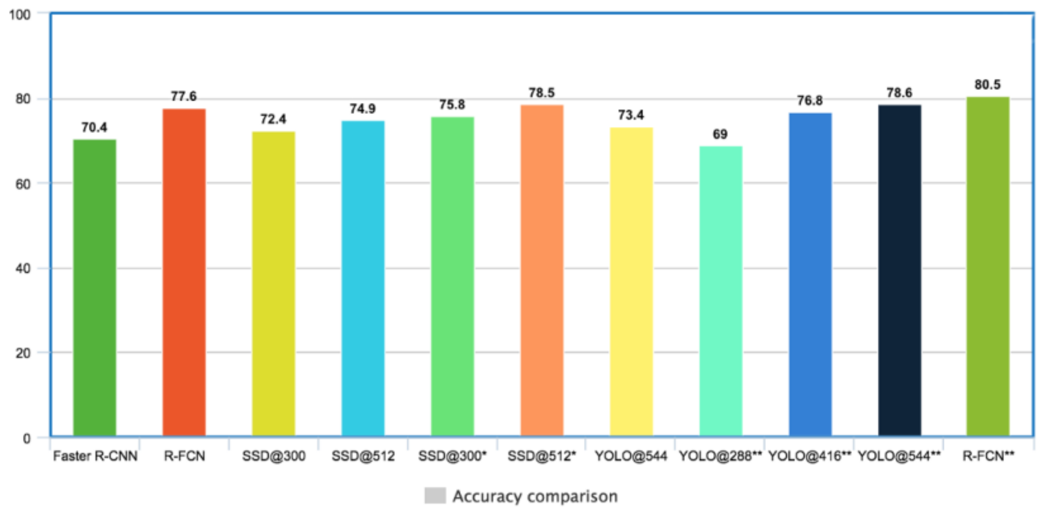


Figure 3.15: YOLO accuracy compared to other state-of-the-art object detectors

- **Good generalization :**
  This is particularly valid for the updated iterations of YOLO, which will be discussed later. With those improvements, YOLO went a step further and offered improved generalization for new domains, making it ideal for applications that require fast and robust object detection.[48]

- **Open-source :**

By making YOLO open-source, the model has been continuously improved by the community. This explains some of the reasons YOLO has advanced so far in such a short period of time.[48]

YOLO architecture :

After receiving an image as input, the YOLO method employs a basic deep convolutional neural network to identify objects in the image. YOLO architecture is as illustrated below, it has overall 24 convolutional layers, 4 max-pooling layers, and two fully connected layers.[49]

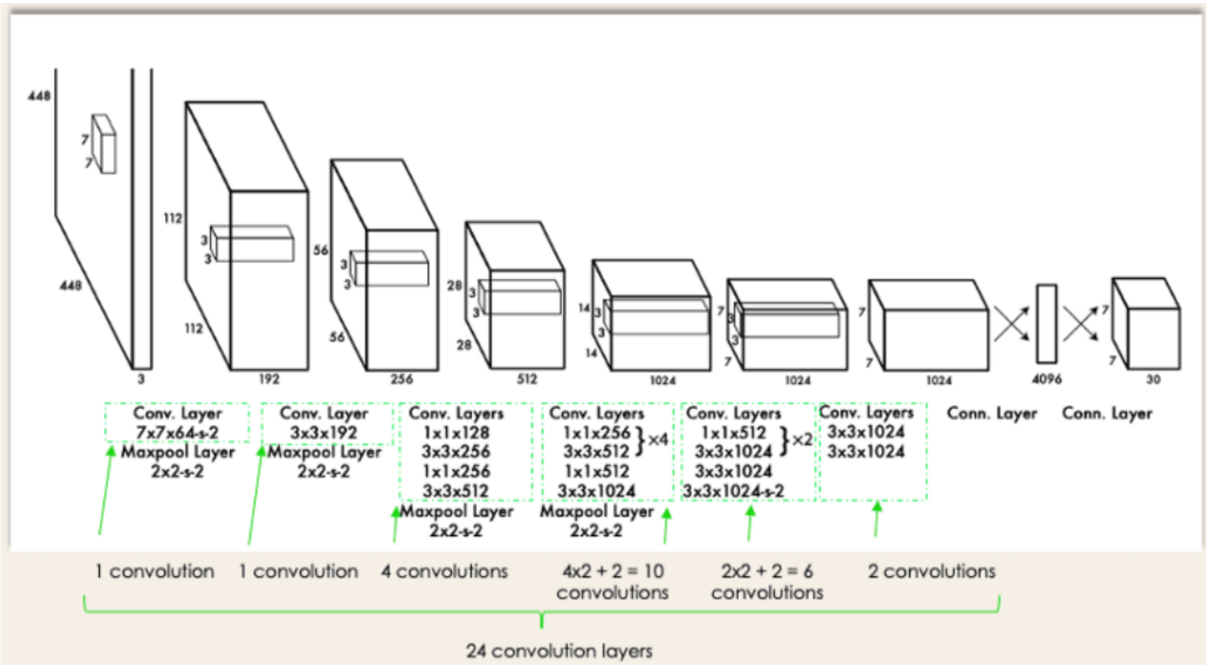


Figure 3.16: The architecture of YOLO algorithm

YOLO operating concept :

You Only Look Once (YOLO) proposes using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection.[49]

An input image is divided into a S × S grid by YOLO. An object's center falls into a grid cell, and that grid cell is in charge of detecting it. Bounding boxes and confidence scores for those boxes are predicted for each grid cell. The model's level of confidence that the box includes an item and the accuracy of the anticipated box are both indicated by these confidence scores.

For each grid cell, YOLO predicts numerous bounding boxes. We only want one bounding box predictor to be in charge of each item during training. Depending on whose prediction has the greatest current IOU with the ground truth, YOLO assigns one predictor as "responsible" for making an object prediction. As a result, the bounding box predictors become specialized. By enhancing its ability to predict certain item sizes, aspect ratios, or classifications, each predictor raises the recall score as a whole.

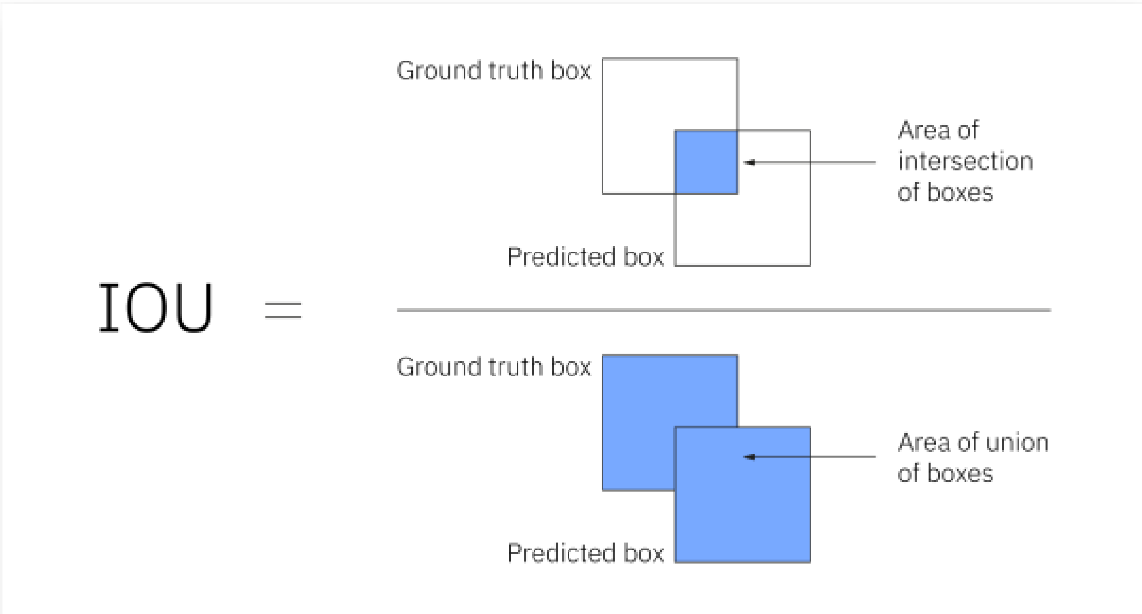


Figure 3.17: Intersection over Union (IoU)

Non-maximum suppression is a crucial method in the YOLO models (NMS). NMS is a post-processing procedure that increases object detection's precision and effectiveness. It is typical practice in object detection to produce numerous bounding boxes for a single item in an image. All of these bounding boxes show the same item, even if they could overlap or be in various locations. NMS is used to extract a single bounding box for each item in the image and to find and eliminate unnecessary or inaccurate bounding boxes.

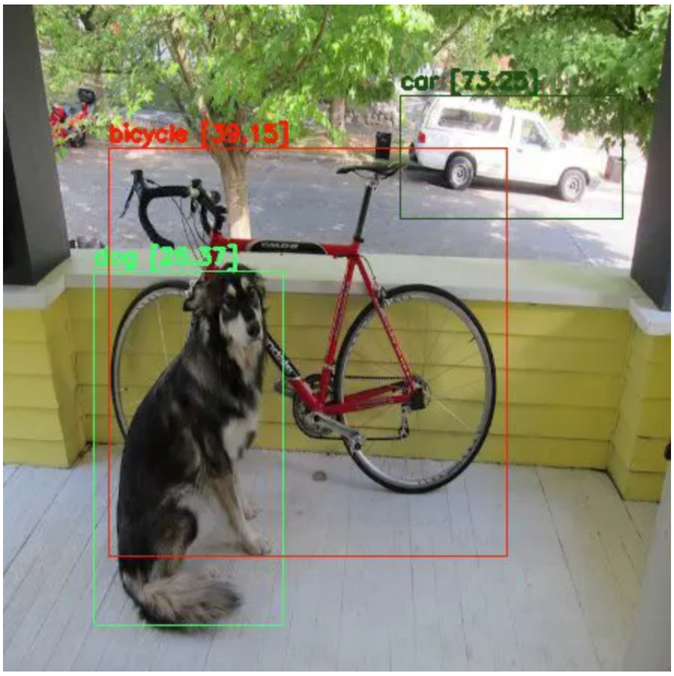


Figure 3.18: Model predicting a dog, car, and bicycle[50]

YOLO applications :

There are several uses for YOLO object detection in daily life. We shall discuss a few of them in the following areas in this section: self-driving automobiles, healthcare, agriculture, and security surveillance. Let's understand each one with specific examples:

- **Healthcare :**
  Because patients differ biologically from one another, it might be difficult to locate organs in real time during surgery. YOLOv3 was utilized in Kidney Recognition in CT to help localize the kidneys in 2D and 3D from CT (computerized tomography) images.[51]

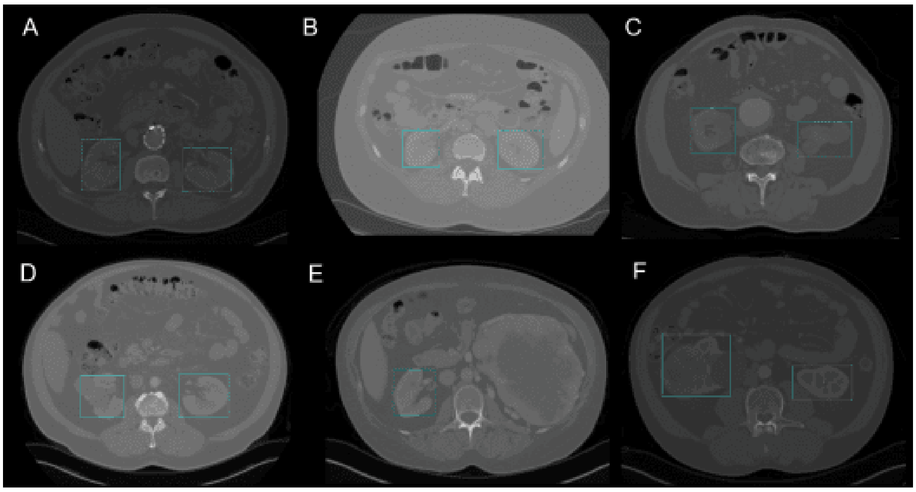


Figure 3.19: 2D Kidney detection by YOLOv3

- **Agriculture :**
  Today's agriculture, robots and artificial intelligence are crucial. Robots with vision systems called harvesting robots were developed to take the position of human fruit and vegetable pickers. YOLO is used by one of the greatest models in this sector to determine the best fruit and vegetable varieties for harvesting.[52]

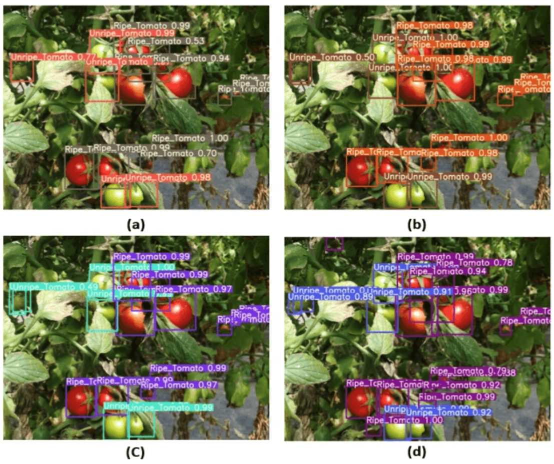


Figure 3.20: Image from Tomato detection based on modified YOLOv3 framework

- **Security surveillance :**
  Although security monitoring is the primary use for object detection, it is not the only one. YOLOv3 was utilized to calculate social distance violations between individuals during the COVID-19 epidemic.[53]

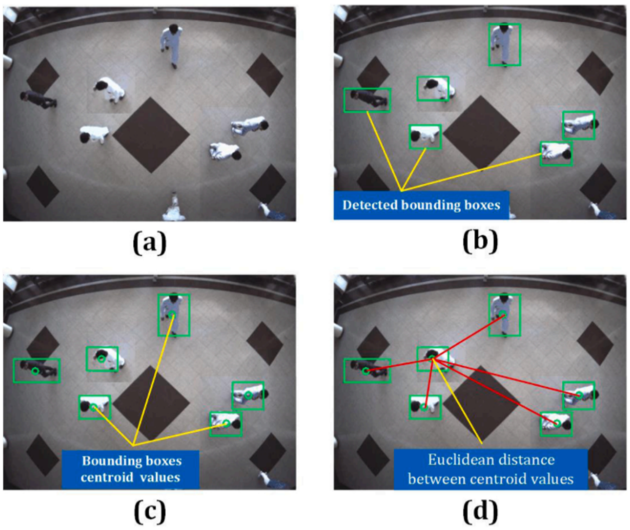


Figure 3.21: The image of calculating the centroid of each detected bounding box and the distance between each pair of centroids.

YOLO versions :

With several iterations since its first release in 2015, YOLO has seen significant development. In this part, we will learn about the variations among each of these variants.[48]

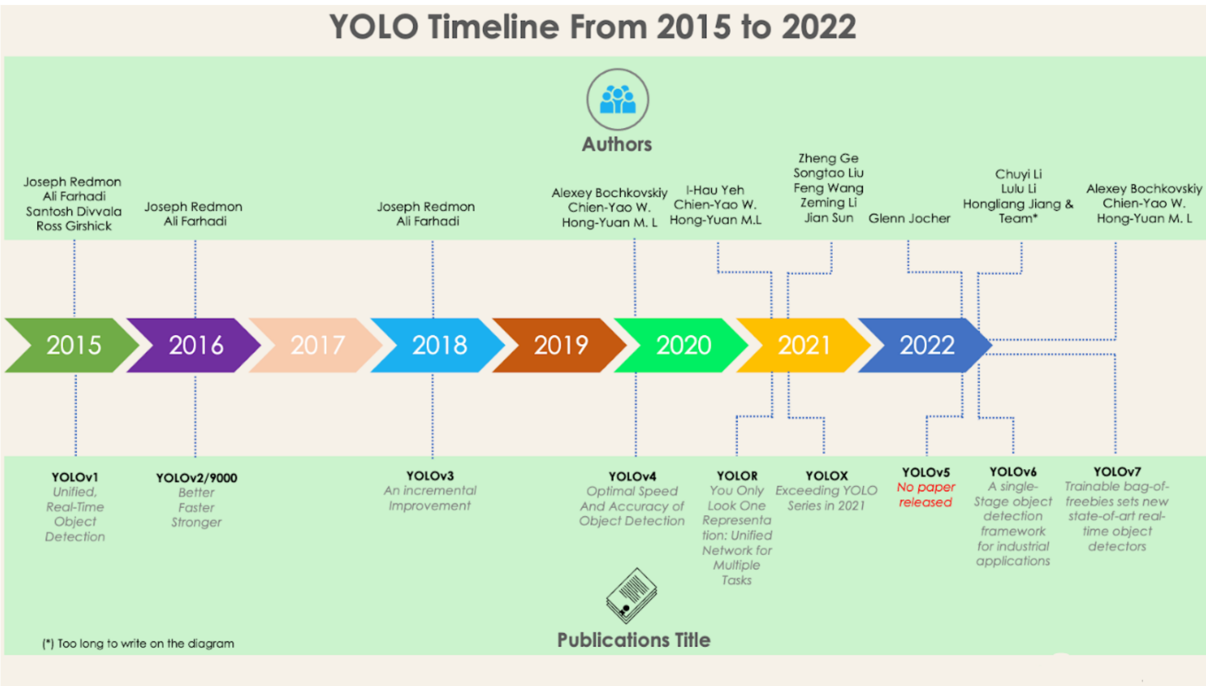


Figure 3.22: YOLO timeline from 2015 to 2022

3.4 Object tracking on drones

An algorithm is used in object tracking, a deep learning method, to follow an item's movements. To put it another way, it's the process of figuring out where moving objects are in a video and other pertinent details. Typically, object tracking involves the object detection procedure. Here is a brief rundown of the steps:

- Item detection involves the creation of a bounding box around an item for the algorithm to classify and detect it.
- Giving everything a distinct identity (ID).
- Keeping track of the identified item as it travels between frames and logging the necessary data.

3.5 Conclusion

The popular object identification technique YOLO (You Only Look Once) has completely changed the area of computer vision. It is a great option for applications requiring real-time object identification because of its speed and effectiveness. It has been enthusiastically embraced in several real-world applications and has demonstrated state-of-the-art performance on multiple benchmarks.

CHAPTER 4

Realisation of project

4.1 Introduction

This section will be devoted to the real-time simulation and implementation of our quadcopter. However, some other challenges and issues emerged during the implementation process. One of the big challenges we faced, which led to a long waiting process, was the unavailability of equipment on the market, including motors, batteries, and chargers. Once we had the equipment, we made a model of our quadcopter, although it is not considered the final version because it still needs a lot of modification and updating and it still has not reached our goal yet.

4.2 The choice of component

The parts required to build a quadcopter are selected based on several factors, including weight, size, cost, design, and intended purpose. The selection of drone parts is proportional as each part is chosen in relation to the others. The figure below shows the selection of brushless motor size and propeller reference based on the size of the quadcopter frame. Note that the size of the structure shown below corresponds to the distance between two motors at opposite corners (diagonally).[54]

Frame size	Propeller diameter	Motor size	KV
150mm	3Inch	1105, 1306	2000kv-3000KV
180mm	4Inch	1806	2600KV-3000KV
210mm	5Inch	2204, 2208, 2306	2300KV-2600KV
250mm	6Inch	2204, 2208, 2306	2000KV-2300KV
350mm	7Inch	2208	1600KV
450mm	8Inch, 9Inch, 10Inch	2212	1000KV

Table 4.1: Table shows the choice of brushless motor and propellers based on quadcopter frame size

From there, we know the size of the motors to target as well as the recommended KV range. As a reminder, a motor is described according to its size and its rotation speed for 1V. That is to say in our choices, a motor labeled “2208 - 2000KV” can rotate at 2000 rpm per volt (i.e. for a 3S lipo, at approximately 11.1V: 11.1 x 2000 = 22200 rpm). Brushless motors generally cannot exceed 50,000 or 60,000 rpm, with some exceptions. The higher the KV, the faster the motor will be able to run. On the contrary, the lower the KV, the more torque the motor will have.

The same for the choice of electronic speed controller, it depends on the size of the propellers and the size of the motor diameter. The table below shows the choice of ESCs.

Propellers size	Stator of motor diameter	Recommended KV	ESC capacity
2Inch	11mm	4000KV-8000KV	6A-12A
3Inch	13mm-14mm	3000KV-4000KV	12A-20A
4Inch	13mm-22mm	2400KV-2900KV	20A
5Inch	22mm-23mm	2200KV-2800KV	20A-35A
6Inch	22mm-23mm	2200KV-2800KV	30A-40A
7Inch	30mm	1800KV-2300KV	30A and more

Table 4.2: Table shows the choice of ESCs based on propellers and stator of brushless motor size

4.3 Characteristics of our drone

This section will cover every component of our quadcopter, including a description of each one's pricing, specifications, and references. The following table indicates that in particular:

Components	Reference	Characteristics	Price
4 × Brushless motors	SKYRC/2208	2000kv	8800DA
4 × Propellers	5045	12.7cm/11.43cm	1000DA
1 × LiPo battery	HJ POWER	3S/11.1v-1000mah-35C	5000DA
1 × LiPO battery charger	IMAX-B3	3S/2S-11.1v/7.4v	2100DA
4 × ESC	HW	30A	6800DA
1 × Quadcopter frame	QAV250	250mm/155g	6000DA
1 × Esp32	Dev Module	WIFI/ULP/BLE	1500DA
1 × Arduino	NANO	16 MHz clock speed	1200DA
1 × Accelerometer/gyroscope sensor	MPU6050	MEMS/3-axis accelerometer/3-axis gyroscope	550DA
1 × Ultrasound sensor	HC-SR04	2cm to 80cm	400DA
1 × Barometric sensor	BMP280	Pressure of 300hPa to 1100hPa/Temperature of -40°C to 85°C	350DA
2 × RF circuit	NRF24L01	Range of 15,24m to 60,96m	1500DA
2 × RF circuit drivers	HW-200	Voltage of4.8V to 12V	500DA
2 × Joysticks modules	Compatible with all boards	2 independent Potentiometer: one for each axis (X and Y)	600DA
Total price			36300DA

Table 4.3: Table showing drone equipment along with references, specifications, and costs

4.4 Simulation and realisation

4.4.1 The pre-final prototype :

In this model we attached the BLDC pins to the outputs of the Arduino NANO board, note that:

- Front right motor ⟷ pin A3 of Arduino NANO board ⟷ pin 13 of ESP32 board.
- Front left motor ⟷ pin A2 of Arduino NANO board ⟷ pin 12 of ESP32 board.
- Back right motor ⟷ pin A1 of Arduino NANO board ⟷ pin 26 of ESP32 board.
- Back left motor ⟷ pin A0 of Arduino NANO board ⟷ pin 25 of ESP32 board.

The PWM signal generation program is inserted into the ESP32 board, but the ESP32 generates signals of 3.3v level voltage in its GPIO, and the inputs of the ESC are forced 5v signals, so we use the Arduino NANO board as a booster of 3.3v to 5v.
Keep in mind that the voltage of a LiPo battery 11.1 V powers the BLDC motor. The ESC served as an adaptor circuit, converting 11.1 volts to 5 volts so that various boards (ESP32, Arduino NANO, etc.) could be powered.

The MPU6050 sensor is necessary for the accelerometer to measure linear acceleration or the gyroscope to identify orientation degrees. Additionally, the MPU6050 module may be utilized with either the SPI or I2C protocols; in our case we choose to use the I2C protocol. Therefore, the MPU6050 pin wiring is as follows:

- VCC pin ⟷ pin 5v of Arduino NANO board.
- SDA pin ⟷ pin 21 of ESP32 board.
- SCL pin ⟷ pin 22 of ESP32 board.
- GND pin ⟷ pin GND of ESP32 board.

The ultrasonic sensor measures an object's distance using sonar. It goes like this:

1. A high-frequency sound (40 KHz) is emitted by the ultrasonic transmitter (trigger pin).

2. The sound waves go through the atmosphere. Should it discover an item, it returns to the module.
3. The reflected sound (echo) is picked up by the ultrasonic receiver (echo pin).

Therefore, the ultrasonic sensor pins wiring is as follows:

- VCC pin ⟷ pin 5v of Arduino NANO board.
- Trig pin ⟷ pin 15 of ESP32 board.
- Echo pin ⟷ pin 14 of ESP32 board.
- GND pin ⟷ pin GND of ESP32 board.

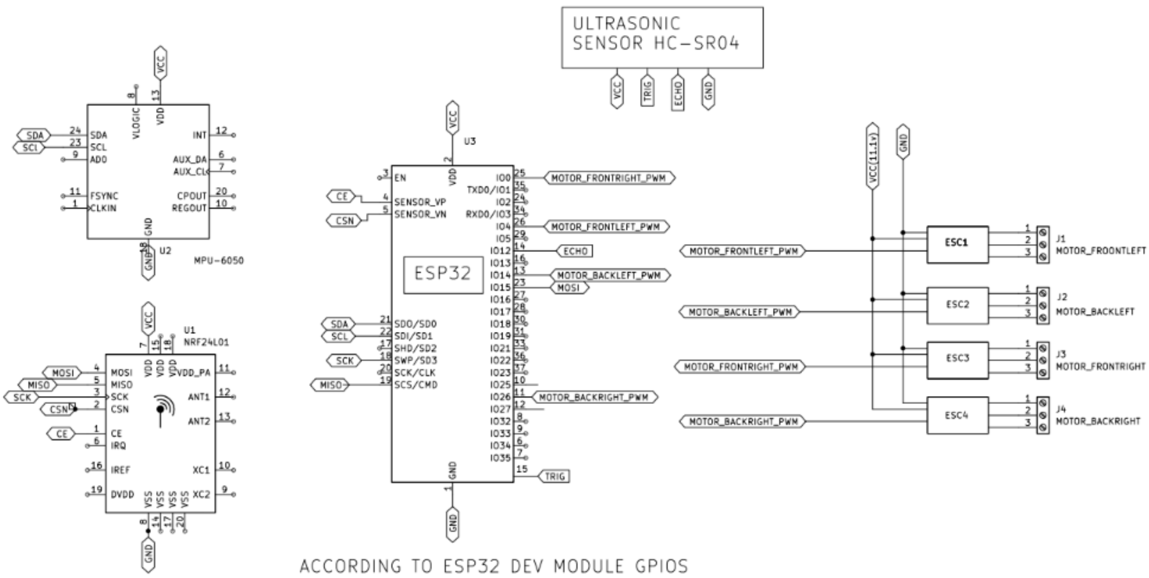


Figure 4.1: The quadcopter command circuit according to Kicad

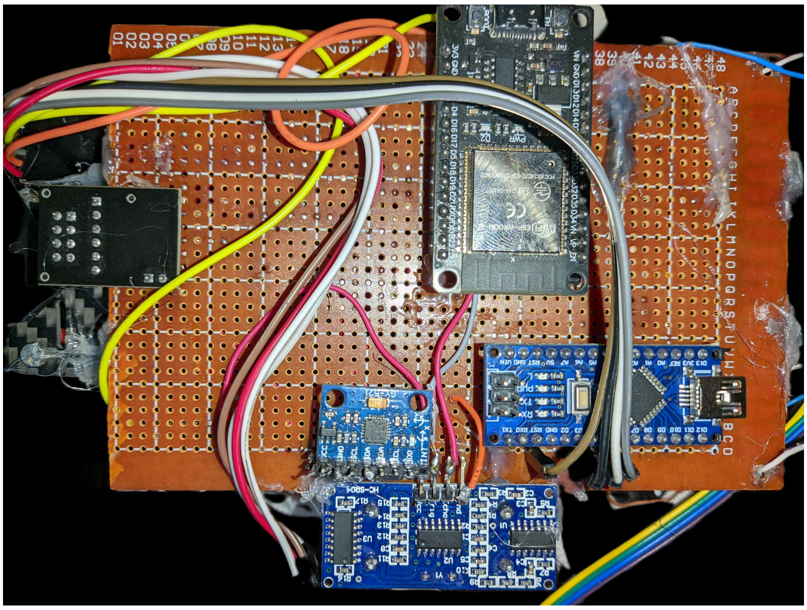


Figure 4.2: The quadcopter command circuit

The pre-final prototype version of our quadcopter is:

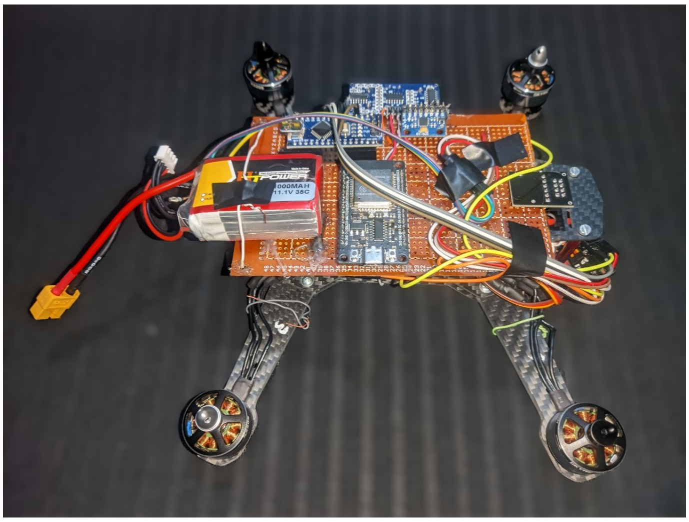


Figure 4.3: Quadcopter prototype without the propellers

4.4.2 The remote controller :

The remote control we made is based on an NRF24L01 module with an Arduino NANO controller and two joystick modules to control the drone at a range of 15m to 60m. The remote control we made is based on an NRF24L01 module with an Arduino NANO controller and two joystick modules to control the drone at a distance of 15m to 60m. We also added their drivers to protect the signal. However, we encountered a communication problem between the controller and the drone. So to avoid the risk of losing connection to the controller, we replaced it with an application called RemoteXY.

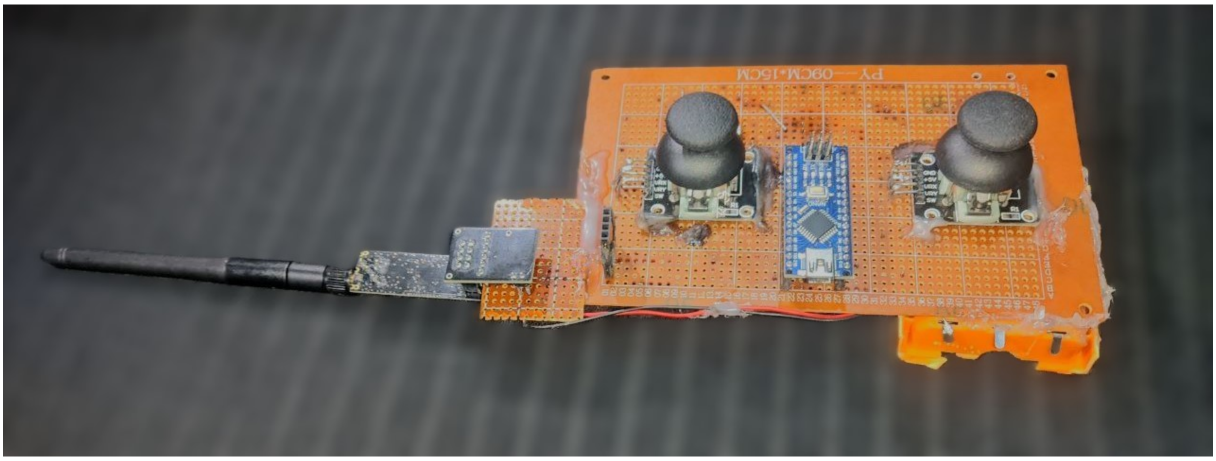


Figure 4.4: The remote controller of our quadcopter based on NRF24L01 module

RemoteXY is an application that creates a graphical interface to control the board. It provides the source code for your GUI, and you can communicate with your flight controller via Bluetooth.

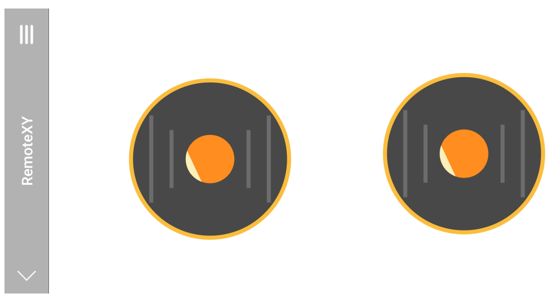


Figure 4.5: The remote controller (RemoteXY) based on Bluetooth

4.4.3 The test of PID :

Initial state test :

In the initial state, the drone's altitude, orientation degrees, and PWM are all equal to zero, or the hovering constant.

```
distance: 0.05 m
mfrontleft 99  mbackright 100  mbackleft 96  mfrontright 103
Rotation towards x: -2.76 degree
Rotation towards y: -3.65 degree
Rotation towards z: 0.40 degree
```

Figure 4.6: The initial state of quadcopter test results

Pitch forward test :

The two back motors will revolve faster than the two front motors in pitch forward motion. However, to calculate the error between them and make adjustments to minimize the error difference between them, the PID regulator takes the pitch angle's degree of orientation and compares it to the reference value, which equals 0 degrees in the no command signal state and 45 degrees in the pitch motion command.

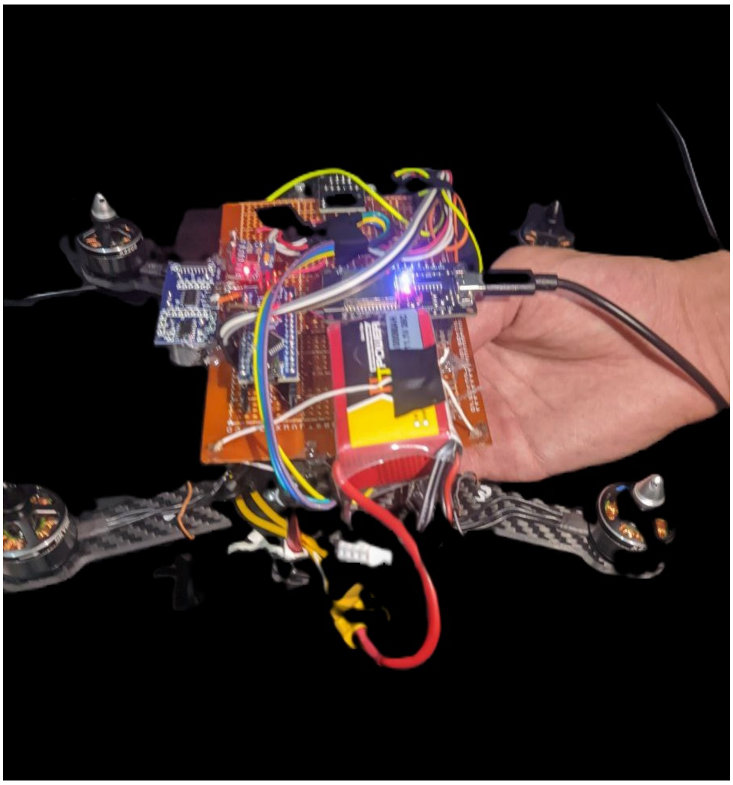


Figure 4.7: Pitch forward motion

```
distance: 0.50 m
mfrontleft 122  mbackright 77  mbackleft 81  mfrontright 118
Rotation towards x: -35.50 degree
Rotation towards y: -3.28 degree
Rotation towards z: -22.72 degree
```

Figure 4.8: Pitch forward motion test result

Pitch backward test :

The two front motors will revolve faster than the two back motors in pitch backward motion. However, to calculate the error between them and make adjustments to minimize the error difference between them, the PID regulator takes the pitch angle's degree of orientation and compares it to the reference value, which equals 0 degrees in the no command signal state and 45 degrees in the pitch motion command.

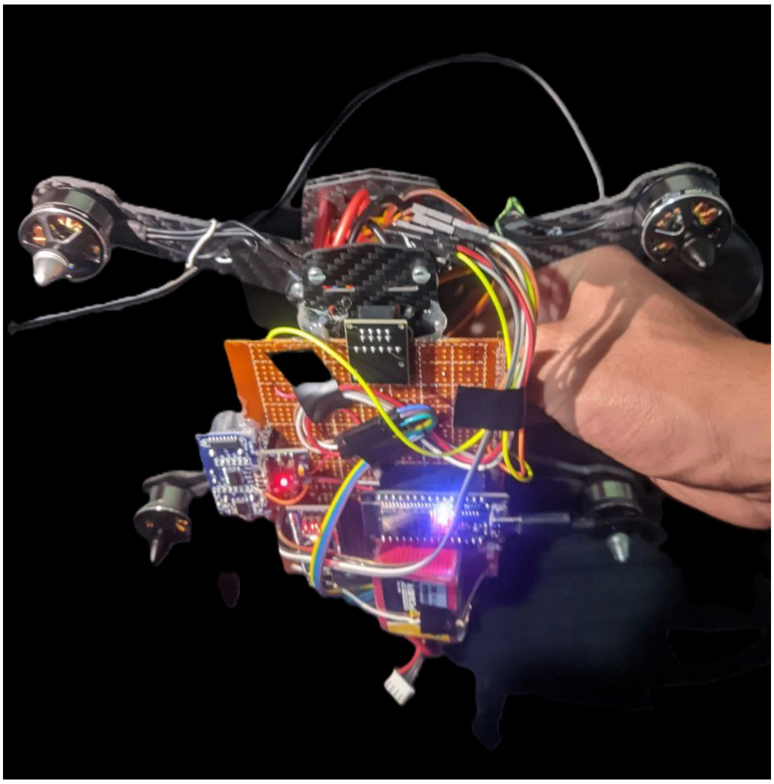


Figure 4.9: Pitch backward motion

```
distance: 0.04 m
mfrontleft 79  mbackright 120  mbackleft 120  mfrontright 79
Rotation towards x: 40.78 degree
Rotation towards y: -1.94 degree
Rotation towards z: -41.84 degree
```

Figure 4.10: Pitch backward motion test result

Roll right test :

The two left motors will revolve faster than the two right motors in roll right motion. However, to calculate the error between them and make adjustments to minimize the error difference between them, the PID regulator takes the roll angle's degree of orientation and compares it to the reference value, which equals 0 degrees in the no command signal state and 45 degrees in the roll motion command.
[htbp]

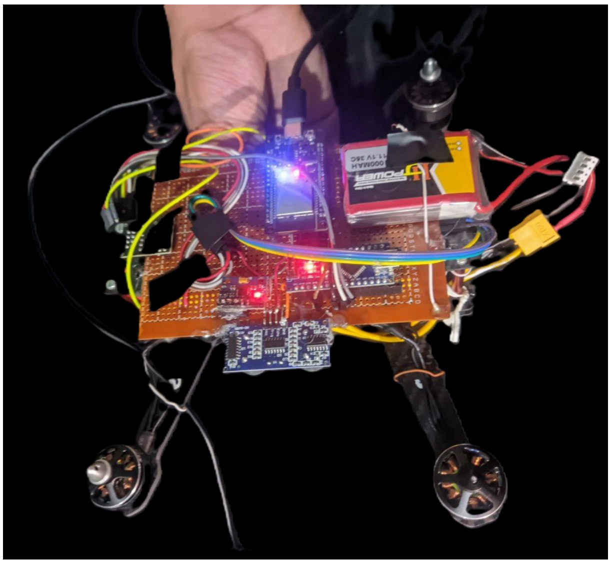


Figure 4.11: Roll right motion

```
distance: 0.05 m
mfrontleft 89  mbackright 110  mbackleft 84  mfrontright 115
Rotation towards x: -5.96 degree
Rotation towards y: -25.74 degree
Rotation towards z: -19.76 degree
```

Figure 4.12: Roll right motion test result

Roll left test :

The two right motors will revolve faster than the two left motors in roll left motion. However, to calculate the error between them and make adjustments to minimize the error difference between them, the PID regulator takes the roll angle's degree of orientation and compares it to the reference value, which equals 0 degrees in the no command signal state and 45 degrees in the roll motion command.

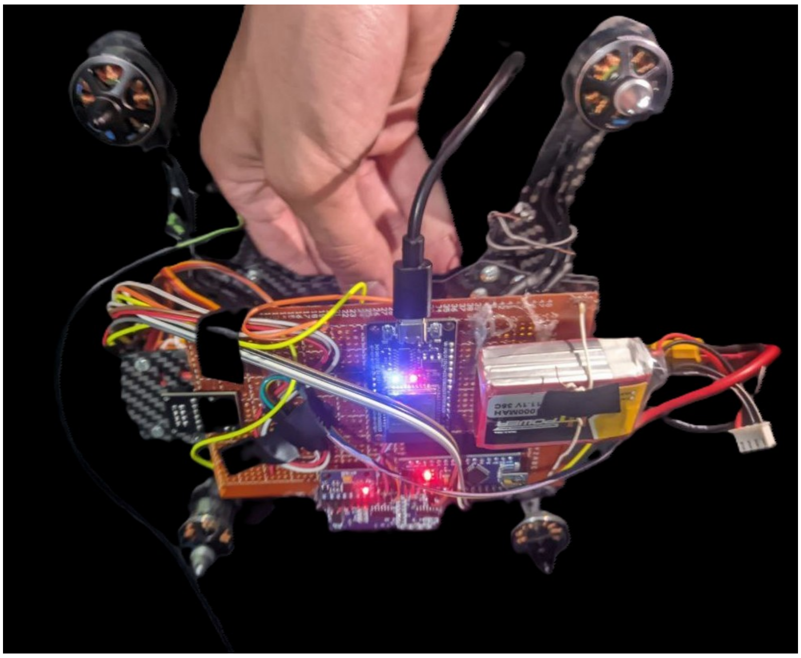


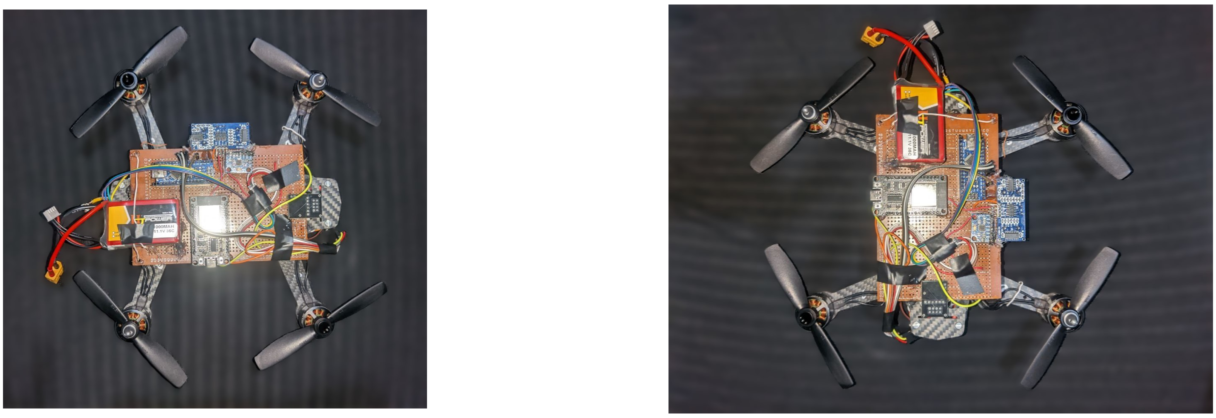
Figure 4.13: Roll left motion

```
distance: 0.09 m
mfrontleft 129  mbackright 70  mbackleft 118  mfrontright 81
Rotation towards x: -10.25 degree
Rotation towards y: 48.53 degree
Rotation towards z: -42.72 degree
```

Figure 4.14: Roll left motion test result

4.4.4 The final prototype :

TTo get the desired full version, our quadcopter's final model still needs upgrades. We don't even have enough time to use computer vision algorithms for object tracking and identification. To follow the quadcopter's obviously, one of the changes our drone requires is a GPS sensor to identify the location coordinates X and Y.



a b

Figure 4.15: The final prototype of our quadcopter

4.5 Conclusion

Much information regarding UAVs and their hardware-assisted integrated systems design is provided by this project. It displays the components and necessary equipment to fly a drone that is completely operational. Companies still gather data and conduct experiments to create the essential hardware and theory for drone use, even though they do not share many information about the advanced design of the drone.

General Conclusion

The quadrotor has seen a tremendous rise in popularity recently because to the growing interest in unmanned aerial vehicles (UAVs), vertical landing and takeoff aircraft, and the necessity for discrete and, most importantly, lightweight instrumentation. This serves as the primary driving force for the research. This work's primary goal was to build a small, remotely operated, quadrotor-based ESP32 drone that is stable against both endogenous and external disruptions.

The first chapter's objective was to give a comprehensive overview of the drones that are now in use and available so that the reader would have a better understanding before moving on to the quadrotor's theoretical and practical aspects.

The aim of the second chapter was to model and create a dynamic model that, with the help of the acting effects and Newton-Euler equations, characterizes the translation and rotation dynamics of the quadrotor system and deduces the equations of motion. And the PID command was used to control and stabilize the drone during any possible movement.

The third chapter describes how convolutional neural networks work to detect objects in images and then track them, but unfortunately, we did not have time to complete this part and apply it to the quadcopter.

Ultimately, our project's simulation and realization are the focus of Chapter fourth. The future quadrotor we intend to construct will enable us to:

- Without human assistance, follow the path provided by an Android application or voice command.
- For self-learning, store trajectory and external environment detection data in a large database.
- The flight lasted more than forty minutes.

Bibliography

[1] Richard J. Gross. L'évolution et l'histoire complète des drones : des années 1800 à 2022, mai 11, 2023.

[2] Thomas Van Hare. The first drone, August 29, 2013.

[3] MD FAHIM HOSSAIN, SHUVRO SANKAR SEN, ASHRAF CHOWDHURY ASHIK, and MOHAMMAD ZOHURUL ISLAM. *DESIGN AND IMPLEMENTATION OF AN IOT BASED FIRE AND SURVIVOR DETECTION DRONE*. PhD thesis, Faculty of Engineering, American International University–Bangladesh, 2023.

[4] ELMAHARAT Anis and LAKHDARI Raouf. *Conception et réalisation d'un mini drone*. PhD thesis, Faculté des Sciences et Technologies, 2021.

[5] Saifeddine Benhadhria, Mohamed Mansouri, Ameni Benkhlifa, Imed Gharbi, and Nadhem Jlili. Vagadrone: Intelligent and fully automatic drone based on raspberry pi and android. *Applied Sciences*, 11(7):3153, 2021.

[6] Apurv Saha, Akash Kumar, and Aishwary Kumar Sahu. Face recognition drone. In *2018 3rd International Conference for Convergence in Technology (I2CT)*, pages 1–5. IEEE, 2018.

[7] NOC. Impact of drone technology and use cases, September 29, 2020.

[8] htpratique. Quels sont les différents types de drones?, avril 11, 2020.

[9] Sumanth Pola. Classification of drones, Wednesday, November 11, 2015.

[10] YAAKOUB KHEDDAR and MOHAMED BELGHOUL. *Modélisation et commande d'un drone Quadrirotor*. PhD thesis, Université Ibn Khaldoun, 2017.

[11] Zakaria Bellahcene. Synthèse de lois de commande robuste pour un hélicoptère à quatre hélices, Mar 2013.

[12] Max Skyler. Understanding drone specifications.

[13] WIKIPEDIA The Free Encyclopedia. Esp32, 25 janvier 2024 at 14:56.

[14] RendyIrawan. Esphome crashes and gets stuck on reboot loop only after adding a few certain lines of code, Feb '21.

[15] ElProCus. What is nrf24l01 : Pin configuration and its working.

[16] Shawn. Getting started with nrf24l01 transceiver: Arduino guide, 4 years ago.

[17] COMPONENTS101. Joystick module, 2 April 2018.

[18] WIKIPEDIA The Free Encyclopedia. Lithium polymer battery, 20 November 2023, at 03:05.

[19] Mr.Khettab.K. Robotique aérienne. *E-learning University M'SILA*, 2021-2022.

[20] Jérôme. Module nrf24l01 arduino : caractéristiques, librairies, et explications (tutorial avec exemples de code arduino). *Passion Electronique*, 26 janvier 2024 à 13:35.

[21] Chris Johnson. How do drones work and fly, 5 years ago.

[22] Lauren Nagel. What is an electronic speed controller and how does an esc work, August 15, 2023.

[23] Yangjulia. Electronic speed controllers (escs) for drones: How do they work?, 13 Jun 2023.

[24] Microship. Bipolar switching.

[25] Paul Posea. How do drone propellers work? (complete beginner's guide), 2024.

[26] Laure Martin. Hélices de drone (guide du débutant), 20 janvier 2022.

[27] Components101. Mpu6050 accelerometer and gyroscope module, 17 March 2021.

[28] RAMASY Heritier Arantes. Modelisation, conception et realisation d'un vehicule aerien sans pilote quadrotor. *Academia.edu*, 06 Avril 2017.

[29] LastMinuteEngineers. Getting started with esp32-cam.

[30] Altitude-Drones. Avantages et inconvÉnients du drone.

[31] Dr. Sharad Pachpute. How do drones fly in air? which drone is more popular?

[32] Devopedia. Quadcopter.

[33] Kevin Craig. Angular velocity: Misunderstood and misstated, March 21, 2013.

[34] Yunze Li. *Design and development of an interactive autonomous monitoring and surveillance drone.* PhD thesis, University of Ontario Institute of Technology (Ontario Tech University), 2020.

[35] Bhavya Pareek, Priyanka Gupta, Gaurav Singal, and Riti Kushwaha. Person identification using autonomous drone through resource constraint devices. In *2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, pages 124–129. IEEE, 2019.

[36] Augmented AI. Object detection vs. classification in computer vision: Explained.

[37] Shreya Mattoo. What is object detection? importance, models and types, July 28, 2022.

[38] Sergio Sánchez Hernández, H Romero, and A Morales. A review: Comparison of performance metrics of pretrained models for object detection using the tensorflow framework. *IOP Conference Series: Materials Science and Engineering*, 844:012024, 06 2020.

[39] Sai. Traditional and representational machine learning, May 18, 2020.

[40] Alberto Rizzoli. The ultimate guide to object detection, June 10, 2021.

[41] Gourav Singh. Introduction to artificial neural networks, 24 May, 2024.

[42] Grant Sanderson. But what is a neural network?

[43] MK Gurucharan. Basic cnn architecture: Explaining 5 layers of convolutional neural network.

[44] Dharmaraj. Convolutional neural networks (cnn) — architecture explained.

[45] aimanasif2799. Weights and bias in neural networks.

[46] Wang Hao, Wang Yizhou, Lou Yaqin, and Song Zhili. The role of activation function in cnn. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pages 429–432, 2020.

[47] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[48] Zoumana Keita. Yolo object detection explained.

[49] Rohit Kundu. Yolo: Algorithm for object detection explained [+examples].

[50] Aditya Sharma. Understanding a real-time object detection network: You only look once (yolov1).

[51] Andréanne Lemay. Kidney recognition in ct using yolov3. *arXiv preprint arXiv:1910.01268*, 2019.

[52] Mubashiru Olarewaju Lawal. Tomato detection based on modified yolov3 framework. *Scientific Reports*, 11(1):1–11, 2021.

[53] Imran Ahmed, Misbah Ahmad, Joel JPC Rodrigues, Gwanggil Jeon, and Sadia Din. A deep learning-based social distance monitoring framework for covid-19. *Sustainable cities and society*, 65:102571, 2021.

[54] LordGG. Concevoir son drone : Du choix des composants au premier vol.