MISR UNIVERSITY
FOR SCIENCE & TECHNOLOGY

Misr University for Science and Technology Faculty of Engineering.

Department of Electronics and Communication Engineering.

B. Sc. Final Year Project

PROJECT TITLE

AI-Based Vision and Audio System for Drone and UAV Detection, Tracking and Blocking

Presented By

Ahmed Samir Zaki	95855
Nada Raafat Ali	95905
Hamza Farahat Mohamed	91713
Amr Ashraf Ibrahem	95593
Mariez Samy Eissa	97903

Supervisor/ Dr. Ashraf Mostafa Samy

Supervisor/ Dr. Mamdouh Gouda

2024 - 2025

MISR UNIVERSITY
FOR SCIENCE & TECHNOLOGY

Misr University for Science and Technology Faculty of Engineering.

Department of Electronics and Communication Engineering.

B. Sc. Final Year Project

PROJECT TITLE

AI-Based Vision and Audio System for Drone and UAV Detection, Tracking and Blocking

Presented By

Ahmed Samir Zaki	95855
Nada Raafat Ali	95905
Hamza Farahat Mohamed	91713
Amr Ashraf Ibrahem	95593
Mariez Samy Eissa	97903

Supervisor/ Dr. Ashraf Mostafa Samy

Supervisor/ Dr. Mamdouh Gouda

2024 - 2025

DECLARATION

I hereby certify that this material, which I now submit for assessment on the program of study leading to the award of Bachelor of Science in (insert title of degree for which registered) is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed

Registration No.:

Date: Day, xx Month Year.

ACKNOWLEDGMENT

We would like to express our sincere gratitude and appreciation to **Dr. Ashraf Mostafa Samy** and **Dr. Mamdouh Gouda** for their invaluable guidance and support throughout the graduation project. Their experience, insights, and dedication have been instrumental in shaping our ideas and guiding us toward successful completion.

We would also like to extend our deepest thanks to our family and friends for their unwavering encouragement and understanding. Their continued support has been a source of motivation and strength during difficult times.

Our team would like to acknowledge the contributions of every individual who has played a role in our project, whether big or small. Everyone's feedback and assistance have been critical to achieving our goals.

Finally, we would like to express our gratitude to our fellow team members for their hard work, cooperation, and commitment. Our collective efforts and shared vision have been the driving force behind our project achievements.

Thanks again to Dr. Mamdouh Gouda, Dr. Ashraf Mostafa Samy, our families, friends, and everyone who participated in the graduation project. Your support and belief in us have made a huge difference, and we are truly grateful for the opportunity to work alongside such exceptional individuals.

List of contents

List of figures

CHAPTER 2. DRONES & GNSS

CHAPTER 3. THE HARDWARE COMPONENTS OVERVIEW

CHAPTER 4. ARTIFICIAL INTELLIGENCE

CHAPTER 5. BLOCKING TECHNIQUES

CHAPTER 6. RESULTS

Chapter 1

Overview of the system

1.1 Background

Unmanned aerial vehicles (UAVs), commonly known as drones, have emerged as one of the most transformative technologies in recent years. Initially developed for military purposes, drones are now used across a wide spectrum of civil, commercial, and governmental applications. From package delivery and infrastructure inspection to precision agriculture and film making, the utility and accessibility of drones continue to expand rapidly.

Despite their benefits, the increased availability and affordability of drones have introduced new challenges, particularly in the context of unauthorized or malicious drone operations. These threats include illegal surveillance, illegal trade, disruption of public events, and even potential terrorist activities. For example, drones have been reported flying over restricted airspace near airports, causing flight delays and posing serious safety risks. In military or governmental contexts, drones can be used for espionage or unauthorized reconnaissance.

To mitigate such risks, it is essential to develop systems capable of detecting, identifying, and neutralizing unauthorized drones in real time. These systems must be accurate, reliable, and adaptable to various environments and threat levels. A complete counter-drone solution typically includes two major components: detection and jamming.

1.2 Drone Detection

Detection is the first and most critical step in any drone defense system. Accurate detection ensures that countermeasures are only activated when necessary, minimizing false alarms and ensuring public safety. Various detection methods exist, each with its advantages and limitations. This project focuses on two complementary approaches: vision-based and sound-based detection.

1.2.1 Vision-Based Detection

Vision-based detection involves the use of cameras (either RGB or thermal) combined with artificial intelligence (AI) and image processing algorithms to identify drones in the field of view. Modern object detection models, such as the YOLO (You Only Look Once) family of algorithms, can detect drones based on their shape, size, movement, and visual features.

This method is particularly effective in well-lit, open environments where visual line of sight is available. AI models trained on diverse drone datasets can achieve high detection accuracy and

real-time performance, even distinguishing between drones and similar-looking objects like birds or balloons.

However, vision-based detection can be limited by factors such as:

- Poor lighting or adverse weather conditions.
- Occlusion or cluttered backgrounds.
- Limited detection range due to camera resolution.

1.2.2 Sound-Based Detection

Sound-based detection systems use microphones and digital signal processing techniques to detect the unique acoustic signature of drones. Most drones produce a distinct sound generated by their rotors and motors, typically in the high-frequency range. By analyzing audio signals using spectral and temporal features, it is possible to recognize the presence of a drone and sometimes even estimate its location.

This method is particularly valuable in situations where the drone is not visible but still within auditory range (e.g., behind obstacles, at night, or during cloudy weather). It also serves as an early warning system in environments with low visual visibility.
Challenges associated with sound-based detection include:
Background noise (e.g., wind, vehicles, human activity).
Variations in drone motor types and noise levels.
Limited effectiveness in large or noisy environments.

By combining vision and sound-based techniques, the overall robustness and accuracy of detection can be significantly improved, with each method compensating for the limitations of the other.

1.3 Jamming

Once a drone has been detected and classified as a potential threat, jamming serves as the active strategy to reduce or neutralize that threat. Jamming involves the transmission of intentional radio frequency (RF) signals that interfere with the drone's control or navigation systems. The primary objective is to interrupt the drone's ability to receive commands from its remote operator or to follow GPS-based navigation routes.

Jamming can target several key subsystems of a drone:

- Navigation systems, such as GPS signals in the 1.5 GHz L1 band, disrupting the drone's ability to maintain its flight path.
- Control links, which typically operate in the 2.4 GHz or 5.8 GHz ISM bands, used by most commercial drones for manual control.
- Telemetry and video transmission, which may also use these same frequency bands to transmit real-time data and video back to the operator.

Depending on the intensity and nature of the jamming, drones may respond in several ways: hovering in place, initiating a return-to-home command, or performing an emergency landing. Some more advanced models may be pre-programmed with escape protocols or autonomous fallback behaviors to avoid jamming.

While jamming is effective, it also presents several technical and regulatory challenges:

- The risk of unintended interference with legitimate communication systems operating in the same bands.
- Legal restrictions on RF transmission power and frequency use, particularly in civilian and urban areas.
- The need for directional antennas or precise power control to contain jamming to the threat zone only.
- The possibility of countermeasures implemented in drones, such as frequency hopping or autonomous rerouting.

An essential component of the jamming subsystem is the antenna, which serves as the interface between the transmitter and the electromagnetic environment. The antenna's characteristics including gain, directivity, and operational frequency, directly affect the range, efficiency, and safety of the jamming process. In this project, a directional antenna is employed to focus the jamming energy on the target drone, enhancing effectiveness while minimizing side effects. The antenna is carefully selected or designed to match the specific frequency bands used in the jamming system.
Given these factors, jamming must be precisely controlled and selectively activated, ensuring that it is only deployed when a true threat is detected and confirmed by the detection subsystems.

1.4 Importance and Real-World Applications

The integration of multi-modal detection systems with targeted jamming is crucial for defending sensitive airspace. This is especially true in areas such as:

- Airports, where even a small drone can halt flights and cause economic disruption.
- Military installations, where security is paramount, and unauthorized surveillance can have serious consequences.
- Public events, where drone attacks or disturbances can threaten civilian safety.
- Critical infrastructure, such as power plants, data centers, or government buildings.

A robust, real-time counter-drone system helps ensure safety, security, and continuous operations in these contexts.

1.5 Project Scope and Objectives

This project focuses on the development of a hybrid drone detection and jamming system. The primary components of the system include:

- A vision-based detection subsystem utilizing AI-driven object detection models to identify drones in real time using camera feeds.
- A sound-based detection subsystem that captures and analyzes acoustic signals to detect drones based on the distinct rotor noise patterns they generate.
- A jamming subsystem that transmits interference signals within specific frequency bands commonly used by commercial drones, disrupting their control links, video transmission, or GPS-based navigation.

The proposed system is designed to be modular, scalable, and suitable for deployment in a wide range of environments. It aims to provide early detection, accurate classification, and safe mitigation of unauthorized or potentially dangerous drones.

In addition, the project incorporates the design and integration of a high-performance antenna tailored to the jamming subsystem. This antenna plays a critical role in effectively radiating the jamming signals, with a focus on covering specific frequency bands (1.1 GHz-1.7 GHz).

In the following chapters, we will explore the theoretical background, system architecture, component design, implementation procedures, and performance evaluation of each subsystem in detail.

Chapter 2

Drones & GNSS

Presented by/

A. AMR ASHRAF IBRAHEM ANANY 95593

HAMZA FARAHAT MOHAMED HAMZA 91713

2.1 Drone

Unmanned Aerial Vehicles (UAVs), commonly referred to as drones, have seen exponential growth in both civilian and military applications. Their versatility and accessibility have led to widespread use, but also to significant concerns regarding safety, privacy, and security. This chapter explores the various types of drones, their typical specifications, the potential dangers they pose, and the critical role of Global Navigation Satellite Systems (GNSS) in their operation.

2.1.1 Types of Drones

Drones are generally classified based on size, range, and application. Each type offers unique advantages and limitations that make it suitable for specific applications.

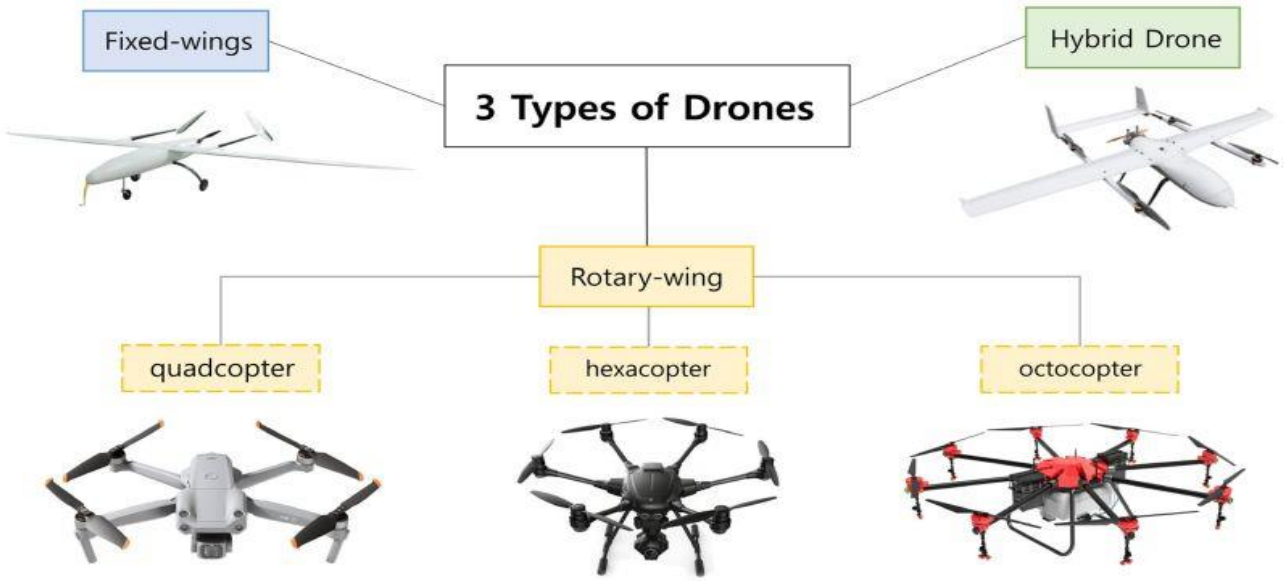


Figure 2. 1 Types of Drones

2.1.1.1 The multi-rotor drone

It is the most used type, particularly in commercial and recreational fields. These drones are typically configured with four (quadcopter), six (hexacopter), or eight (octocopter) rotors. Their key advantages include ease of control, vertical takeoff and landing (VTOL) capability, and stable hovering, which makes them highly effective for aerial photography, surveillance missions, and infrastructure inspection

Figure 2. 2 Quadcopter

Figure 2. 3 Hexacopter

Figure 2. 4 octocopter

Multi-rotor drones typically offer flight times between 10 to 30 minutes and have a control range of up to 5 kilometers. Their payload capacity is moderate, often supporting lightweight cameras or small sensors, and they can reach speeds of 30–60 km/h. Due to their hovering stability, they are equipped with GNSS receivers, high-definition cameras, and ultrasonic sensors for precise positioning and obstacle avoidance.

2.1.1.2 fixed-wing drones

Fixed-wing drones is also (UAV) resembling traditional aircraft and can cover larger areas over extended flight durations. These drones require a runway or launching mechanism for takeoff and cannot hover or land vertically. Their aerodynamic design allows for greater energy efficiency, longer operational range, and higher cruising speeds. Fixed-wing drones are predominantly used in applications such as topographic mapping, agricultural monitoring, and military reconnaissance.

Figure 2. 3 Fixed wing drone

Fixed-wing drones are built for endurance, with flight times ranging from 1 to 3 hours and control ranges extending beyond 10 kilometers. They can carry larger payloads, including mapping equipment or agricultural sensors, and achieve higher speeds of 70–150 km/h. These drones commonly utilize GNSS, inertial navigation systems (INS), and advanced imaging sensors, making them suitable for large-scale surveying or reconnaissance.

2.1.1.3 Vertical Take-Off and Landing (VTOL) drone

A more advanced and versatile category is the hybrid Vertical Take-Off and Landing (VTOL) drone, which integrates the long-range and endurance features of fixed-wing platforms with the vertical flight capabilities of multi-rotors. These drones can take off and land vertically like helicopters while transitioning to fixed-wing flight for efficient long-distance travel. Hybrid VTOL drones are increasingly employed in logistics, delivery services, industrial inspections, and emergency response missions.

Figure 2. 4 Hybrid drone

Hybrid VTOL drones combine the features of multi-rotors and fixed-wing platforms, offering flight times of 45 minutes to 2 hours and ranges between 5 to 15 kilometers. Payload capacities vary depending on design but typically support both vertical lift systems and forward-flight gear. These drones integrate multiple navigation sensors including GNSS, barometers, and vision-based systems, along with HD or thermal cameras for inspection and search-and-rescue operations.

2.2 Dangers of Drones

Despite their numerous benefits, drones present several dangers and risks that must be carefully managed. One major concern is security threats, where drones can be used for unauthorized surveillance or spying, capturing sensitive or private information without consent. This raises significant privacy concerns, especially when drones operate over residential areas or critical infrastructure. Additionally, drones have been exploited to carry harmful payloads such as explosives, hazardous materials, or contraband, raising the risk of terrorist attacks, sabotage, or other criminal activities.

From an airspace safety perspective, drones can interfere with manned aircraft, especially near airports, helipads, or flight corridors, increasing the chance of mid-air collisions or accidents that could endanger lives and property. Unauthorized drone flights in restricted or no-fly zones, such

as near government buildings, military bases, or crowded events, pose serious security and safety challenges. Privacy invasion is another significant issue, as drones equipped with high-resolution cameras or thermal imaging can monitor individuals or private properties without permission, potentially leading to misuse or harassment.

Moreover, drones have been increasingly used for illegal activities such as smuggling drugs, weapons, or contraband across borders or into prisons. This complicates law enforcement efforts and demands more effective regulation and countermeasures. Additionally, malfunctioning or poorly controlled drones can cause property damage or personal injury, especially in crowded or urban environments. As drone technology becomes more accessible, the potential for misuse and accidental harm grows, emphasizing the need for strict operational guidelines, advanced detection systems, and public awareness.

2.3 Overview of the GNSS

The Global Navigation Satellite System (GNSS) is a collective term for satellite-based navigation systems that provide accurate positioning, navigation, and timing information worldwide. GNSS includes well-known systems such as the United States' GPS, Russia's GLONASS, Europe's Galileo, and China's BeiDou. These systems consist of a constellation of satellites orbiting the Earth, ground control stations, and user receivers.

GNSS satellites continuously transmit signals containing precise time and orbital data. GNSS receivers, like those installed in drones, process signals from multiple satellites to calculate their exact location in three-dimensional space using trilateration. This positioning information is critical for various applications, including navigation, mapping, timing synchronization, and autonomous control.

In drones, GNSS enables precise flight path planning, real-time location tracking, and autonomous navigation. It supports key functions such as waypoint following, altitude control, and the Return-to-Home feature. Despite its advantages, GNSS technology faces challenges including signal blockages in urban or dense environments, signal jamming, and spoofing attacks that can disrupt or manipulate positioning data. As such, GNSS reliability and security are ongoing areas of research and development to ensure safe and efficient drone operations.

2.3.1 What is GNSS?

The Global Navigation Satellite System (GNSS) is an advanced satellite-based positioning technology that provides users with global location, velocity, and time synchronization.
Unlike single-system navigation methods, GNSS integrates multiple satellite constellations, allowing for more accurate and reliable positioning across the world.
GNSS is widely used in numerous sectors, including aviation, maritime navigation, land surveying, geodesy, telecommunications, disaster response, precision agriculture, and autonomous vehicle navigation. Its ability to provide continuous, real-time positioning makes it an essential tool for both civilian and military applications.

2.3.2 How Does GNSS Work?

GNSS relies on a network of satellites orbiting the Earth that continuously transmit radio signals, these signals carry information about the satellite's position and time of transmission.
A GNSS receiver on the ground determines its location by measuring the time it takes for signals from multiple satellites to reach it. Using a process called trilateration, the receiver calculates its position based on the known locations of at least four satellites. The more satellites in view, the higher the accuracy.
Key elements of GNSS positioning include:

- Satellite Constellations: Groups of satellites that provide global coverage.
- Ground Control Stations: Monitor and control the satellite network.
- User Equipment (Receivers): Devices that process GNSS signals to compute position, velocity, and time.

The satellite broadcasts a signal that contains orbital data (from which the position of the satellite can be calculated) and the precise time the signal was transmitted.
Orbital data includes a rough almanac for all satellites to aid in finding them, and a precise ephemeris for this satellite. The orbital ephemeris is transmitted in a data message that is superimposed on a code that serves as a timing reference. The satellite uses an atomic clock to maintain synchronization of all the satellites in the constellation.
The receiver compares the time of broadcast encoded in the transmission of three (at sea level) or four (which allows an altitude calculation also) different satellites, measuring the time-of-flight to each satellite. Several such measurements can be made at the same time to different satellites, allowing a continual fix to be generated in real time using an adapted version of trilateration.

2.3.3 Major GNSS Constellations

Several countries operate independent GNSS systems to ensure global positioning capability:

- GPS (Global Positioning System) – Operated by the United States, the oldest and most widely used GNSS.
- GLONASS (Globalnaya Navigatsionnaya Sputnikovaya Sistema) – Russia's GNSS, known for using Frequency Division Multiple Access (FDMA).
- Galileo: The European Union's GNSS, designed for high accuracy and civilian independence.
- BeiDou (BDS): China's GNSS, offering global services with an emphasis on the Asia-Pacific region.
- In addition to these core systems, regional satellite navigation systems such as India's NavIC and Japan's QZSS provide additional coverage and augmentation.

2.3.4 GNSS Frequency Bands & Signal Modulation

GNSS operates in specific L-band frequency ranges, allocated by the International Telecommunication Union (ITU) under the Radio Navigation Satellite Service (RNSS).

The main GNSS frequency bands include:

- GPS, the center frequencies are 1575.42 MHz (L1), 1227.6 MHz (L2) and 1176.45 MHz (L5).
- GLONASS operates as frequency divisional multiple access (FDMA) and there are two operational center frequencies 1602 MHz (L1) and 1246 MHz (L2) and at 1207.14MHz (L3).
- GLONASS over this decade will also introduce Code Divisional Multiple Access (CDMA) like GPS.
- GALILEO has a range of frequencies assigned in the L-band as follows:
- E2 – L1 – E1 – Centre frequency 1575.42 MHz (band from 1559MHz – 1591MHz).
- E5A – Centre frequency 1176.45 MHz (band from 1164 MHz – 1188 MHz).
- E5B - Centre frequency 1207.14 MHz (band from 1188 MHz – 1215 MHz).
- E6 – Centre frequency 1278,75 MHz (band from 1260 MHz – 1300 MHz)

To allow multiple GNSS systems to coexist within these bands, modulation techniques like Code Division Multiple Access (CDMA) and Binary Offset Carrier (BOC) are used to reduce interference and enhance performance.

2.3.5 GNSS Augmentation Systems

To improve the accuracy, integrity, and reliability of GNSS signals, augmentation systems provide corrections to reduce errors caused by atmospheric disturbances, clock drifts, and signal obstructions.

Key augmentation systems include:

- SBAS (Satellite-Based Augmentation System): Uses geostationary satellites to provide correction signals (e.g., WAAS in the U.S., EGNOS in Europe, MSAS in Japan, and GAGAN in India).
- GBAS (Ground-Based Augmentation System): Provides high-accuracy corrections via ground stations, primarily used in aviation.
- RTK (Real-Time Kinematic) & PPP (Precise Point Positioning): Techniques used for centimeter-level precision, especially in surveying and autonomous systems.

2.3.6 Applications

GNSS technology has become an integral part of modern life, enabling a wide range of applications across multiple sectors. From everyday navigation to precision-driven scientific research, GNSS supports various industries that rely on accurate, real-time positioning, timing, and geospatial information.

Below are some of the most significant GNSS applications:

Navigation and Transportation.

Automotive Navigation

GNSS is widely used in vehicles for route planning, navigation, and real-time traffic information.

1) **Aviation**
   - GNSS is essential for modern aviation, offering precision navigation in both commercial and military aircraft.
   - Example: Unmanned aerial systems (UAVs) use GNSS for autonomous flight planning and navigation.
2) **Maritime Navigation**
   - GNSS plays a crucial role in ensuring the safe and efficient movement of ships and vessels.
3) **Military and Defense**
   - GNSS provides highly accurate positioning for military operations, enhancing situational awareness and coordination.

Applications:

- Guided missile systems, artillery targeting, and weaponry are guided by GNSS for precision strikes.
- Military personnel and vehicle tracking use GNSS to ensure efficient navigation in terrain and on battlefields.

4) **Earth Observation**
   - GNSS supports scientific research in monitoring Earth processes, space weather, and environmental changes.

2.3.7 The performance of GNSS is assessed using four criteria:

- **Accuracy**: Accuracy refers to the degree of closeness of the estimated position to the true position (real-world location).
- **Integrity**: Integrity is the ability of the GNSS system to provide timely warnings if the system's positioning service is not reliable.
- **Continuity**: Continuity refers to the ability of the GNSS system to maintain service without interruption over time
- **Availability**: Availability indicates the ability of the GNSS to provide service in a specific location, at a given time, and under certain conditions.

2.4 Modulation

Modulation is the process of conveying a message signal, for example a digital bit stream, into a radio frequency signal that can be physically transmitted.

In Global Navigation Satellite Systems (GNSS), Binary Phase Shift Keying (BPSK) and Binary Offset Carrier (BOC) are two important modulation schemes used for transmitting signals from satellites to receivers.

1) **BPSK (Binary Phase Shift Keying)**

   Definition: Phase shift keying is a digital modulation scheme that conveys data by changing, or modulating, the phase of the carrier wave. BPSK uses two phases which are separated by a half cycle.

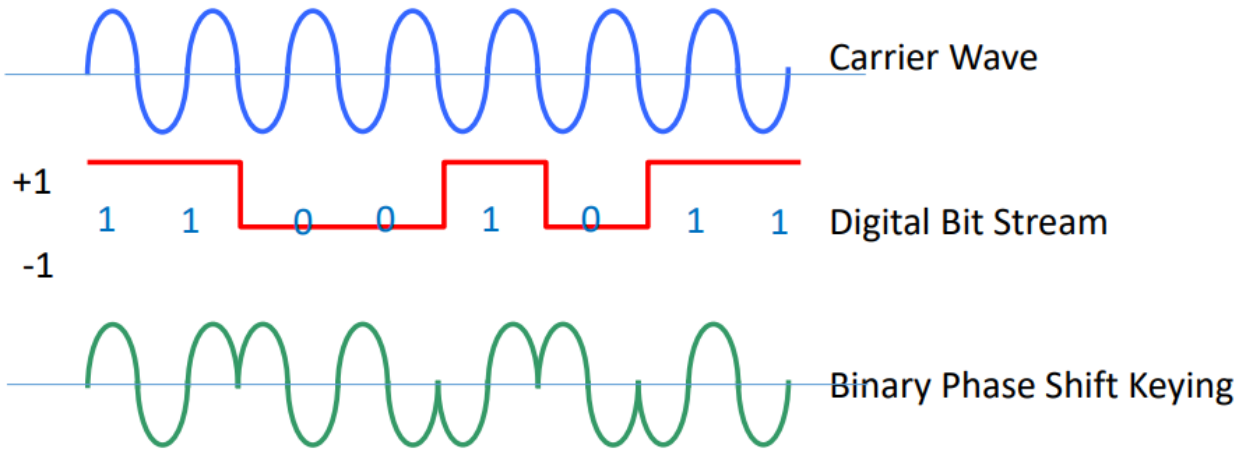


Figure 2. 5 Binary Phase Shift Keying (BPSK) Modulation Process

Characteristics:

- **Robustness:** BPSK is resistant to noise and interference, making it suitable for reliable communication.
- **Bandwidth Efficiency**: It has a narrow bandwidth, which can be an advantage in certain applications.

2) **Binary Offset Carrier (BOC):**

   Definition: BOC is a modulation scheme that combines BPSK with an offset frequency, effectively creating a more complex signal. It transmits binary data with two or more frequencies, resulting in improved spectral properties.

Characteristics:

- **Increased Bandwidth:** BOC signals have a wider bandwidth, which allows for better resistance to interference and multipath effects.
- **Multipath Mitigation:** The modulation structure helps to reduce the impact of multipath interference, enhancing accuracy in positioning

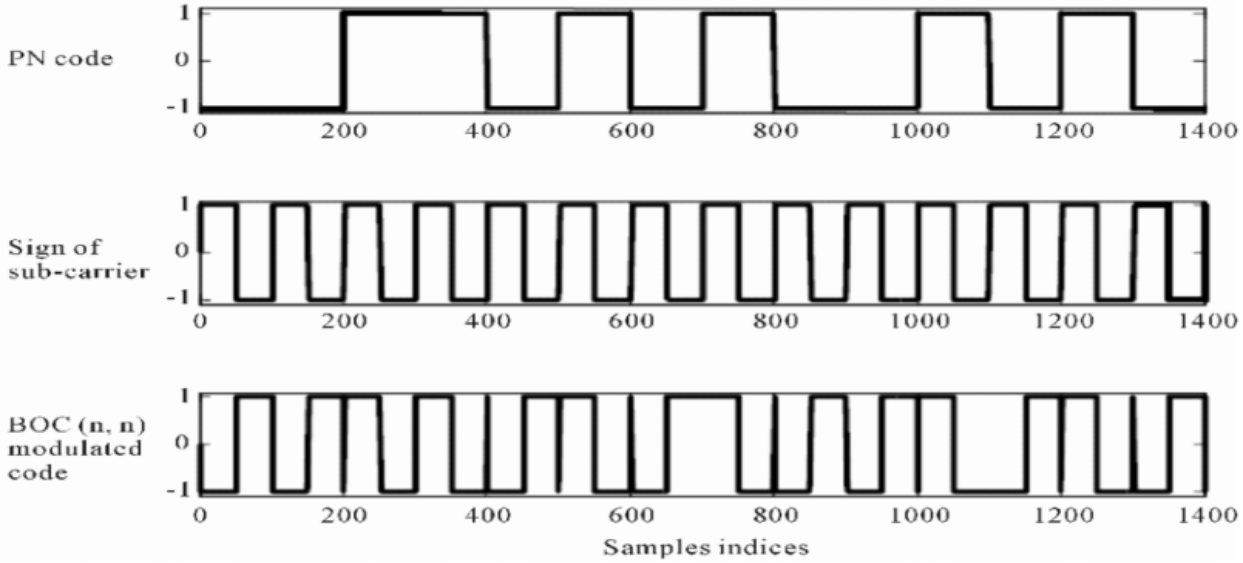


Figure 2. 6 Binary Offset Carrier (BOC) Modulation Components

3) Alternate Binary Offset Carrier (AltBOC):

Definition: is an advanced version of BOC that alternates between different BOC configurations. This modulation technique further optimizes signal performance for tracking and acquisition.

Comparison

- **Complexity:** BPSK is simpler and easier to implement than BOC, which requires more sophisticated signal processing.
- **Performance:** BOC typically provides better performance in challenging environments (like urban areas) due to its enhanced multipath resistance and spectral efficiency.
- **Application:** While BPSK is widely used in established GNSS systems, BOC represents an evolution that is increasingly being adopted in modern GNSS designs for improved signal quality.

2.5 Multiple Access techniques

Multiple Access Techniques refer to the methods used to allow multiple users or devices to share the same communication channel (like radio frequency, time, or space) without interference. These techniques are used to efficiently manage the available bandwidth and ensure that each user gets access to the channel when needed, especially in wireless communication systems.

Frequency Division Multiple Access (FDMA) and Code Division Multiple Access (CDMA) are from the major access techniques used to share the available bandwidth in a wireless communication system.

1) Frequency Division Multiple Access (FDMA)

Each user is assigned a unique frequency band within the available spectrum. Multiple users transmit simultaneously on different frequencies.

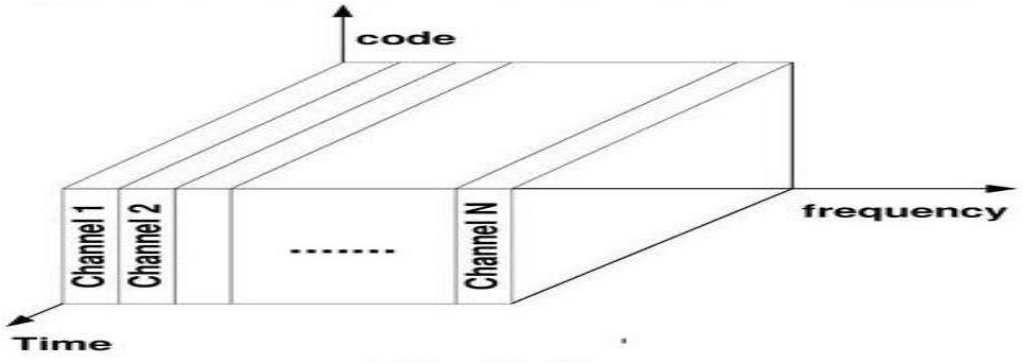


Figure 2. 7 Frequency Division Multiple Access (FDMA)

2) Code Division Multiple Access (CDMA)

Each user is assigned a unique code (a sequence of bits). All users transmit over the same frequency at the same time, but their signals are distinguished based on the codes used.

- All GNSS Signals except GLONASS are based on CDMA .
- Only GLONASS uses FDMA .
- Future Signals of GLONASS will also use CDMA.

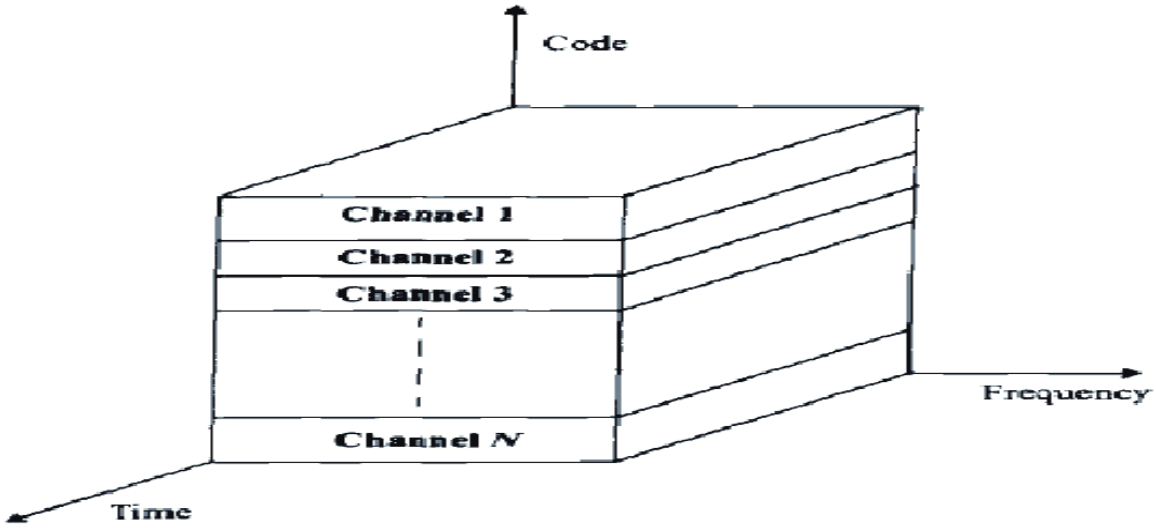


Figure 2. 8 Code Division Multiple Access (CDMA)

CDMA vs. FDMA

	CDMA [GPS, QZSS, Galileo, BeiDou, IRNSS, Future GLONASS Satellites]	FDMA [GLONASS]
PRN Code	Different PRN Code for each satellite Satellites are identified by PRN Code	One PRN Code for all satellites Satellites are identified by center frequency
Frequency	One Frequency for all satellites	Different frequency for each satellite
Merits & Demerits	Receiver design is simpler No Inter-Channel Bias More susceptible to Jamming	Receiver design is complex Inter-channel bias problem Less susceptible to Jamming

Figure 2. 9 CDMA vs FDMA

GPS (Global Positioning System)

- **Modulation Scheme**: CDMA with Binary Phase Shift Keying (BPSK)
- **Details:** Each satellite transmits a unique pseudo-random noise (PRN) code using BPSK modulation. The L1 frequency (1575.42 MHz) carries the C/A (Coarse/Acquisition) code.

GLONASS (Global Navigation Satellite System)

- **Modulation Scheme:** FDMA with Frequency Division Multiple Access
- **Details:** Each satellite operates on a different frequency, which minimizes interference. GLONASS originally used analog signals, but newer signals have incorporated digital modulation techniques.

Galileo

- **Modulation Scheme:** CDMA with BPSK and AltBOC (Alternate Binary Offset Carrier)
- **Details:** Galileo uses a combination of BPSK for the E1 signal and AltBOC for higher frequency signals, which helps improve signal robustness and accuracy.

BeiDou

- **Modulation Scheme:** CDMA
- **Details:** Like GPS and Galileo, BeiDou employs CDMA modulation with different frequencies and codes for its various signals, enhancing capacity and performance.

Constellation	Signal	Frequency [GHz]	Modulation	Multiplexing
GPS	L1 C/A	1575.42	BPSK	CDMA
	L1 C	1575.42	TMBOC	CDMA
	L5	1176.45	BPSK	CDMA
Galileo	E1	1575.42	CBOC	CDMA
	E5a	1176.45	AltBOC	CDMA
	E5b	1207.14	AltBOC	CDMA
Compass	B1	1561.098	QPSK	CDMA
	B2	1207.14	BPSK	CDMA
Glonass	L1OF	1602 + n×0.5625	BPSK	FDMA
	L1OC	1575.42	BOC	CDMA

Figure 2. 10 GNSS Signal Characteristics by configuration

2.6 GPS

The Global Positioning System (GPS) is a satellite-based navigation system developed and maintained by the United States government, providing global coverage for positioning, navigation, and timing services. It consists of a constellation of at least 24 satellites orbiting Earth, transmitting signals on multiple L-band frequencies, including L1 (1575.42 MHz) for civilian use, L2 (1227.60 MHz) for military use and L5 (1176.45 MHz) for safety-critical applications like aviation. GPS receivers calculate their position by measuring the time delay of signals from at least four satellites, achieving accuracy within 1–3 meters for civilian use and even higher precision with augmentation systems like WAAS. Widely used in navigation, mapping, timing, agriculture, and emergency services, GPS is a foundational technology that has revolutionized modern life, with ongoing advancements improving its accuracy, reliability, and resistance to interference.

Characteristics of GPS Signal

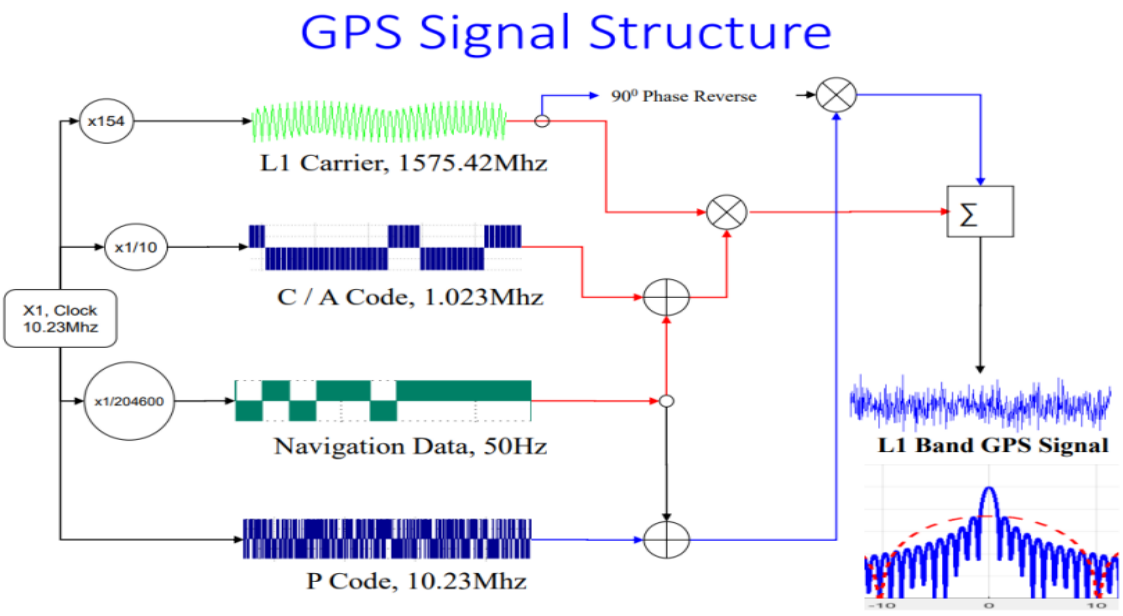


Figure 2. 11 GPS signal structure

GPS Signals have basically three types of signals

1) Carrier Signal

A carrier signal is a waveform (usually a sinusoidal wave) that is used to carry information in communication systems.
Here's how it works:

- **Modulation:** Modulation is the process of conveying a message signal, for example a digital bit stream, into a radio frequency signal that can be physically transmitted.
- **Transmission**: The modulated carrier signal is then transmitted over a medium (like air for radio waves or cables for wired communication). The carrier frequency is typically much higher than the frequency of the information being sent.
- **Demodulation:** At the receiving end, the carrier signal is demodulated to retrieve the original information.

2) PRN Code

A PRN code, or Pseudo-Random Noise code, is used in satellite navigation systems like GPS to uniquely identify each satellite and help with signal processing.
It can be categorized primarily into two types:

1) **C/A Code (Coarse/Acquisition Code):** This is a pseudo-random noise code used for civilian applications. Each satellite has a unique C/A code, allowing receivers to identify and acquire signals quickly.
2) **P(Y) Code (Precision Code):** This code is used primarily for military applications and offers higher accuracy. It is encrypted for security and provides more resistance to jamming compared to the C/A code.

- The C/A (Coarse/Acquisition) code is a specific type of PRN code that serves several key functions:

1. Identification: Each GPS satellite transmits its own unique C/A code, allowing receivers to identify which satellite they are receiving signals from.

2. Signal Acquisition: The C/A code helps receivers quickly acquire and lock onto satellite signals, facilitating the process of determining the satellite's location.

3. Time Synchronization: The code also allows for precise timing measurements, which are essential for calculating distances to the satellites and determining the receiver's position.

4. Frequency Hopping: The C/A code is designed to appear randomly, which helps minimize interference and improves the robustness of the signal against jamming.

- The C/A code is a sequence of bits that repeats every 1,023 millisecond and is a crucial component of the GPS system, primarily used for civilian applications.

PRN Code is a sequence of randomly distributed zeros and ones that is one millisecond long.

- This random distribution follows a specific code generation pattern called Gold Code.
- There are 1023 zeros or ones in one millisecond.
- In case of GPS, PRN code is 1023 bits long.
- Each GPS satellite transmits a unique PRN Code.
- GPS receiver identifies satellites by its unique PRN code or ID
- It is continually repeated every millisecond and serves for signal transit time measurement.
- The receiver can measure where the PRN code is terminated or repeated.

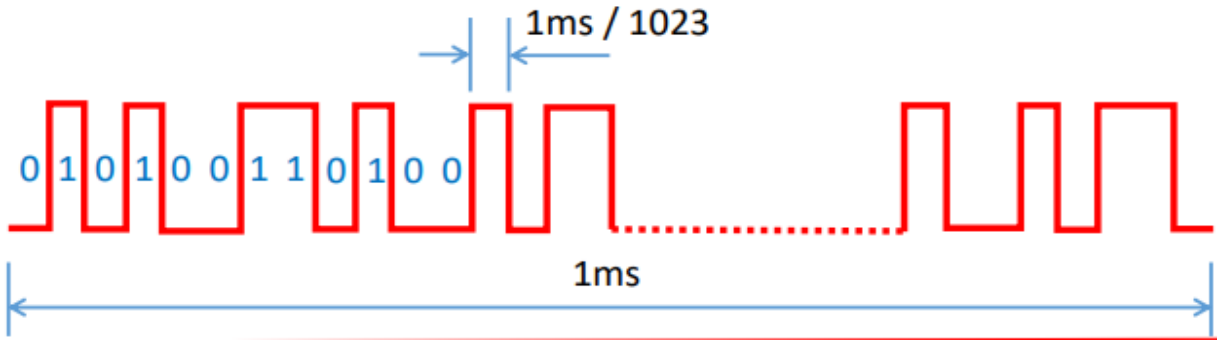


Figure 2. 12 GPS C/A Code Structure Over 1 ms

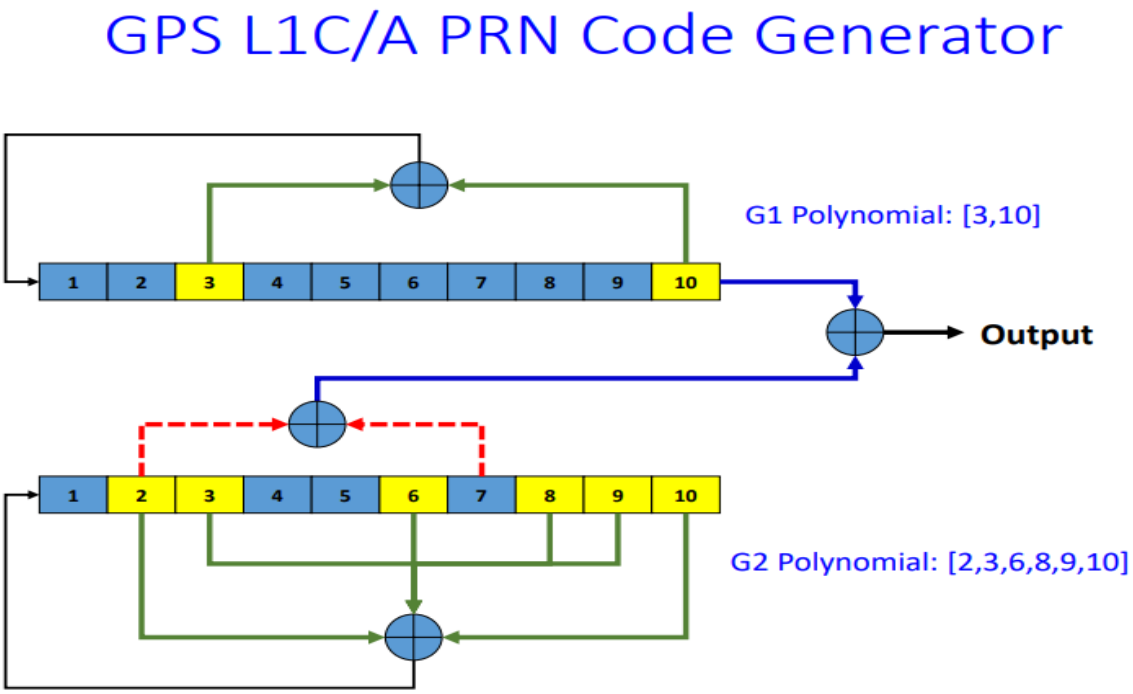


Figure 2. 13 GPS L1 C/A PRN Code Generator Using G1 and G2 Shift Registers

Chapter 3

The Hardware Components Overview

3.1 Introduction

This chapter outlines the requirements for each component in the order they are designed and implemented. The proposed system is a standalone UAV jammer that relies on a vision-based drone detection approach and helical antennas for directional jamming. To operate autonomously, the system integrates multiple hardware modules and software programs that work together seamlessly.

3.2 System Block Diagram

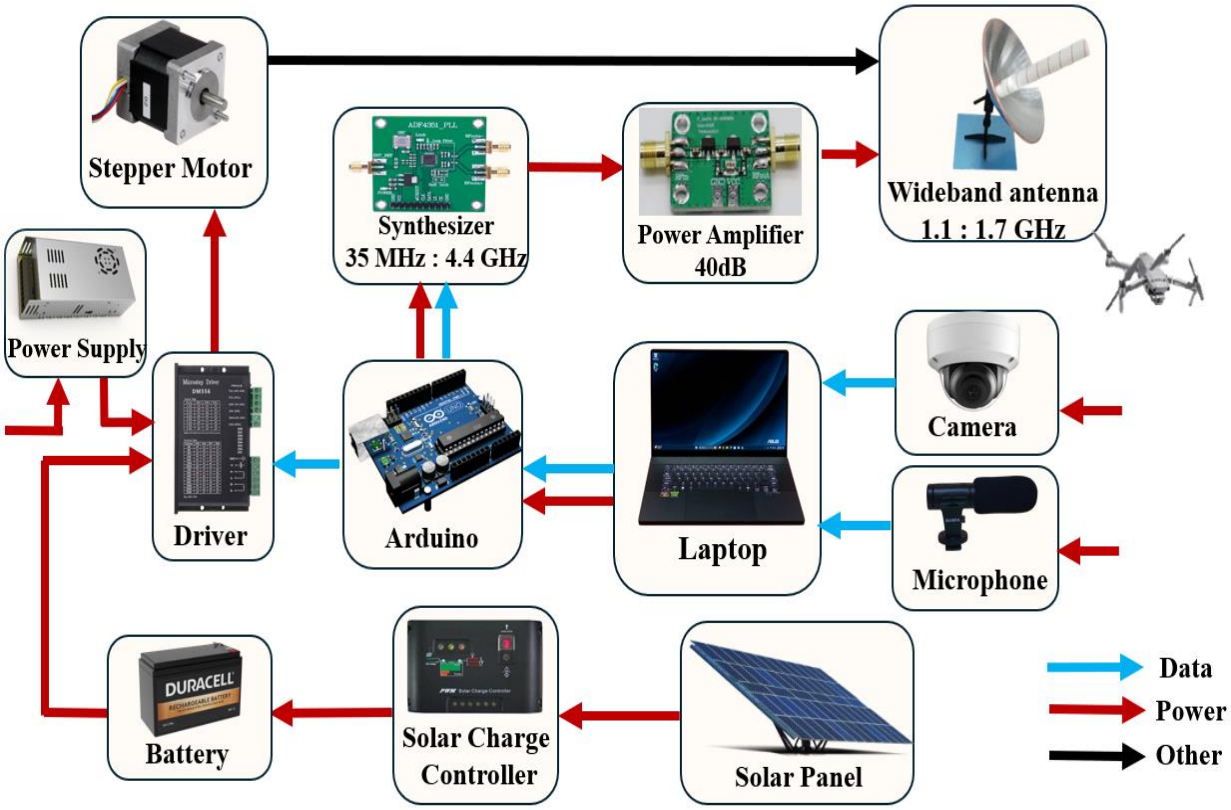


Figure 3. 1 System Block Diagram

3.3 Hardware requirements

To design a project that meets the required standards for a sufficient Embedded system, some requirements such as cost, time saving, security, mobility, and space must be taken into consideration. Therefore, we narrowed down our choice of hardware devices into the following:

3.3.1 Mamen MIC-07 Microphone Specifications

In this project, we used the Mamen MIC-07 mini shotgun microphone due to its compact design, high directionality, and advanced noise reduction features. Below are the detailed specifications and key features of the microphone:

3.3.1.1 Key Features

Figure 3. 2 MAMEN MIC-07

- Built-in 100mAh rechargeable lithium battery.
- Super-cardioid pickup pattern for highly directional audio capture.
- NCR (Noise Control Reduction) technology for effective background noise suppression.
- +10dB gain boost switch to enhance sensitivity and pickup range.
- 3.5mm headphone jack for real-time audio monitoring.
- 3.5mm TRS output interface to connect directly to cameras, smartphones, etc.
- Micro-USB charging port (compatible with power banks or USB sources).
- Cold shoe mount with a 1/4″ screw hole for mounting on
- cameras, tripods, or ball heads.
- Compatible with iOS, Android, Windows, and standalone audio recorders.
- Integrated power + gain switch and LED indicators for power and charging status.

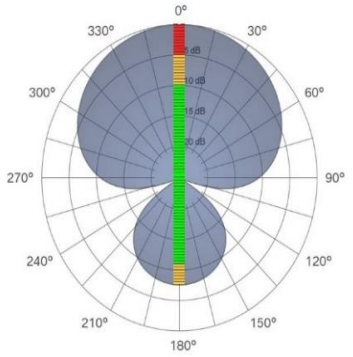


Figure 3. 3 Super Cardioid Pattern

3.3.1.2 Technical Specifications

Pickup Pattern	Super-cardioid - Shotgun	**Charging Port**	Micro-USB
Frequency Response	50Hz – 20kHz	**Battery Built-in**	100mAh lithium, rechargeable
Sensitivity	-38dB ± 3dB (0dB = 1V/Pa at 1kHz)	**Mounting Options**	Cold shoe mount and 1/4″ threaded hole
Output Impedance	2000Ω ± 30% (at 1kHz)	**Gain Boost**	+10 dB gain mode
Monitoring Output	3.5mm headphone jack	**Indicators**	LED for power and charging

Audio Output	3.5mm TRS output jack	**Compatibility**	iOS, Android, Windows, sound recorders, cameras

3.3.2 Professional Camera

4MP Fixed Dome Network Camera

Figure 3. 4 Hikvision IP Camera

- Camera Is used to detect Drones, UAVs and Birds within the dataset used in AI model.
- High quality imaging with 4 MP resolution Efficient H.265+ compression technology
- Clear imaging even with strong back lighting due to 120 dB WDR
- Water and dust resistant (IP67) and vandal resistant (IK10)
- EXIR 2.0: advanced infrared technology with long IR range
- Light Range up to 30 meters
- The field of view is presented in three measurements: horizontal, vertical, and diagonal: horizontal viewing angle in degrees, approximately 98.0°, vertical viewing angle, approximately 53.1° and viewing angle across the diagonal, approximately 114.7°.

3.3.2.1 Camera Features

• High quality imaging with 4 MP resolution

• Efficient H.265+ compression technology

• Clear imaging even with strong back lighting due to 120 dB WDR

3.3.2.2 Camera Specifications

Specification	Value
Max. Resolution	2560×1440
Min. Illumination	Color: 0.01 Lux (F2.0, AGC ON), B/W: 0 Lux with IR
Shutter Time	1/3 s to 1/100,000 s
Day & Night	IR cut filter
Angle Adjustment	Pan: 0° to 355°, Tilt: 0° to 75°
Supplement Light Range	Up to 30 m
Supplement Light Type	IR
Power	12 VDC ± 25%, 0.4 A, max. 5 W, Ø 5.5 mm coaxial power plug PoE: IEEE 802.3af, Class 3, 36 V to 57 V, 0.2 A to 0.15 A, max. 6.5 W
Ethernet Interface	1 RJ45 10 M/100 M self-adaptive Ethernet port

3.3.3 Arduino uno

3.3.3.1 Features

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst-case scenario you can replace the chip for a few dollars and start over again.

3.3.3.2 Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

• VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

• 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator and can damage your board. We don't advise it.

• 3V3. A 3.3-volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

• GND. Ground pins.

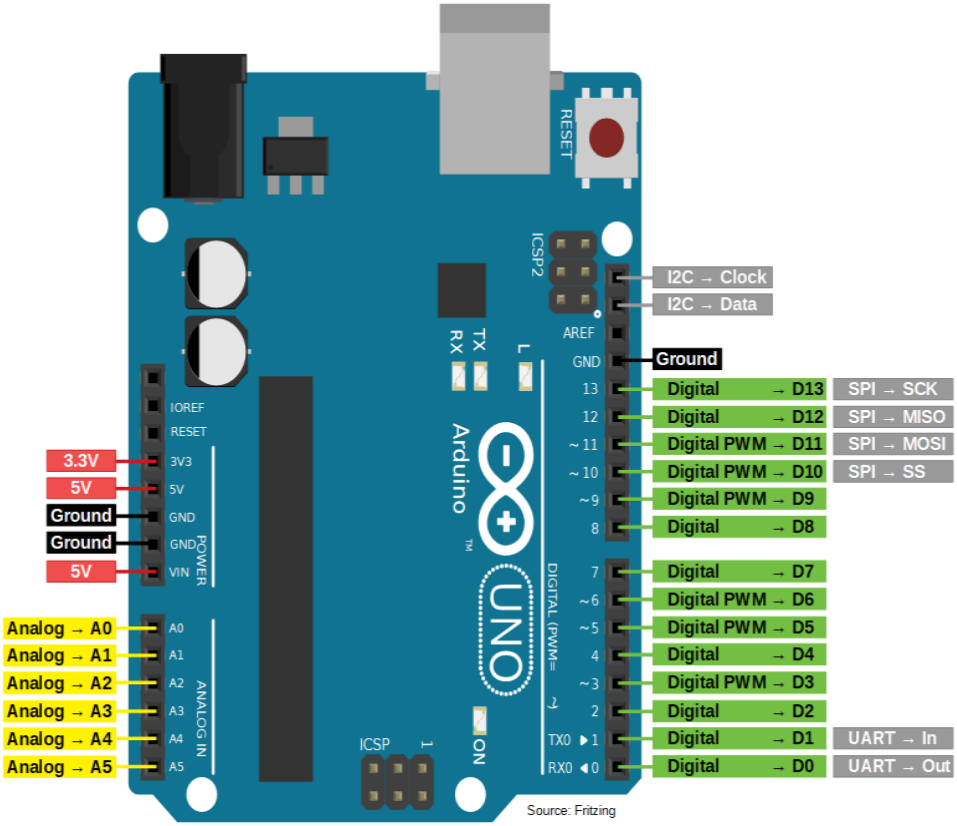


Figure 3. 5 Arduino UNO

3.3.4 Stepper motors

A stepper motor is a unique type of DC motor that rotates in fixed steps of a certain number of degrees. Step size can range from 0.9 to 90°. It consists of a rotor and stator. In this case, the rotor is a permanent magnet, and the stator is made up of electromagnets (field poles). The rotor will move (or step) to align itself with an energized field magnet. If the field magnets are energized one after the other around the circle, the motor can be made to move in a complete circle.

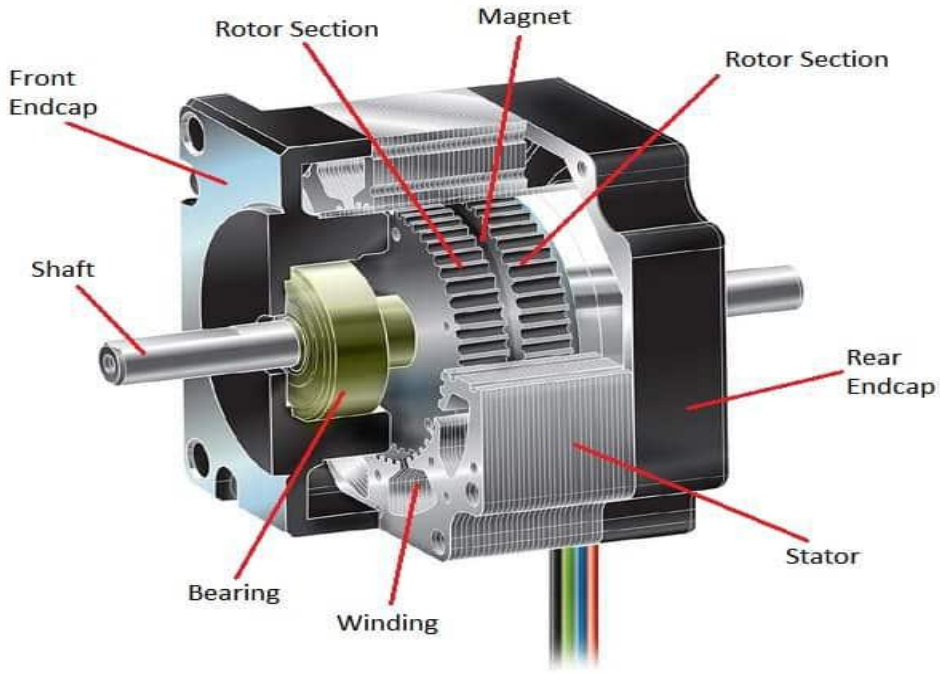


Figure 3. 6 Stepper Motor

In a two-phase bipolar stepping motor, when coils are excited in order one phase at a time, the motor rotates.
If excitation occurs in the opposite order, motor rotation in the opposite direction is possible.

3.3.4.1 Stepper motor features

NEMA 24 35 Kg-cm

- Step angle accuracy: + - 5%(full step, not load)
- Resistance accuracy: + - 10%
- Inductance accuracy: + - 20%
- Ambient temperature ----------20deg ~+50deg
- Insulation resistance:100MΩ Min,500VDC
- Insulation Strength--------500VAC for one minute • Step Angle: 1.8 degrees
- Steps per Revolution: 200
- 2-Phase
- Bipolar
- Voltage (V): 4.8
- Current (A): 3
- Phase Resistance: 1.6Ω
- Phase Inductance: 8.4mH
- Holding Torque: 35.0 kg.cm
- Weight: 1.3kg

3.3.4.2 Stepper motor implementation

1) Black-Brown wire connects to pin A+
2) Green-Orange wire connects to pin A-
3) Red-Yellow wire connects to pin B+
4) Blue-Blue wire connects to pin B-

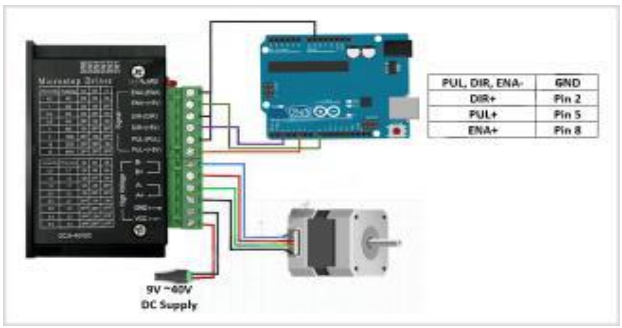


Figure 3. 7 Stepper motor connection

3.4 Motor driver module

The components' job is to react to the step command pulses coming from the machine controller and convert them into the proper on-off pattern required to drive the stepper motor. This electronic device will transform our movement instructions from a controller into a sequence where the winding in the stepper motor will be turned on or off while still providing enough power to it.

We used in our project motor driver DM556.This is a professional two-phase stepper motor driver. It supports speed and direction control. You can set its micro step and output current with 8 DIP switch. There are 16 kinds of micro steps and 8 kinds of current control (1.4A, 2.1A, 2.7A, 3.2A, 3.6A, 4.3A, 4.9A, 5.6A) in all.

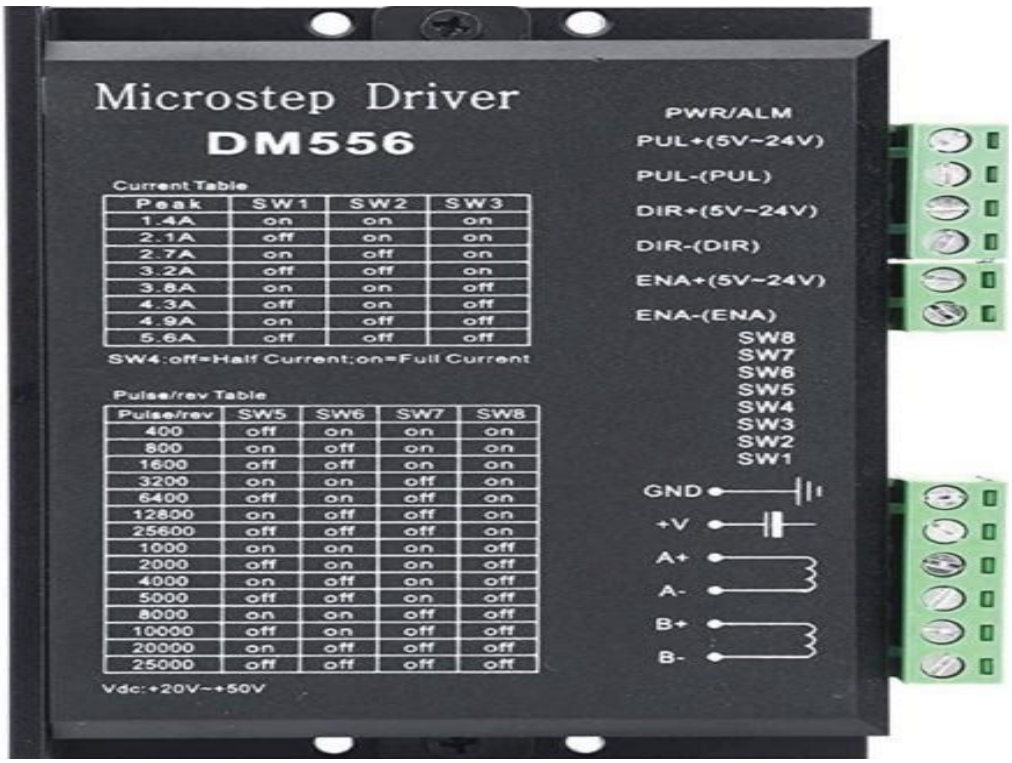


Figure 3. 8 Microstep Driver

3.4.1 Driver features

- Anti-Resonance provides optimum torque and nulls mid-range instabilities.
- Supply voltage up to +50 VDC (recommended not to exceed 45 V because of “back EMF”)
- Output current programmable, from 0.5 A to 5.6A .
- Automatic idle-current reduction (in standstill mode) to reduce motor heating; function switchable (reduction rate can be software configured).
- Motor self-test and parameter auto-setup technology offers optimum responses with different motors.
- Pulse input frequency up to 200 kHz.
- TTL compatible and optically isolated input
- Multi-Stepping allows a low-resolution step input to produce a higher Microstep output for smooth system performance.
- Microstep resolutions programmable, from full-step to 102,400 steps/rev.
- Suitable for 2-phase and 4-phase motors .
- Support PUL/DIR and CW/CCW modes.
- Over-voltage, over-current, phase-error protections.

3.4.2 Electrical Specification

Instruction	DM556N-L			
	Minimum	Typical	Maximum	Unit
Output Current	1.4	-	5.6 (Peak)	Amps
Input Voltage	20	36	50 (contain the ripple)	VDC
	15	24	40(contain the ripple)	VAC
Control Signal Voltage	3	3.3	5.5	VDC
Control Signal Current	7	10	16	mA
Signal Pulse Frequency	0	-	200	KHz
Insulation Resistance	500			MΩ

3.5 Programmable Synthesizer Card

A programmable synthesizer card is a critical component in modern RF systems, particularly in applications such as communication, radar, electronic warfare, and drone jamming systems. It functions as a digitally controlled signal source that can generate RF frequencies over a wide range, with high accuracy and speed. Unlike traditional fixed-frequency oscillators, a programmable synthesizer offers dynamic control of output frequency, allowing it to adapt in real time to the system's requirements.

3.5.1 Purpose and Function

In the context of this project, we are developing a system for detecting and jamming drones; the programmable synthesizer card is used to generate the jamming signal. It enables the system to sweep across different frequency bands (e.g., 1.1 to 1.7 GHz), target multiple drone communication protocols, and adapt its output based on detected threats. This flexibility is crucial for disrupting drones that use varying or frequency-hopping communication techniques.

The card typically interfaces with a microcontroller or an FPGA, which sends digital commands to adjust the output frequency and power level. Through this, it can be programmed to emit specific waveforms or modulated signals in real-time, allowing highly efficient and adaptive jamming.

3.5.2 Core Components

A typical programmable synthesizer card consists of:

- Phase-Locked Loop (PLL): Ensures frequency stability by locking to a reference.
- Voltage-Controlled Oscillator (VCO): Produces variable frequencies controlled by voltage.
- Reference Oscillator: Provides stable base frequency.
- Control Logic (e.g., MCU or FPGA): Sends digital instructions (usually via SPI or I2C).
- Output Stage: Includes amplifiers and filters to deliver a clean signal.

3.5.3 Main Types Available in the Market

Programmable synthesizers are available in several forms depending on the architecture, frequency range, and intended use. Below are the main categories:

3.5.3.1 PLL-Based Synthesizers

These use a phase-locked loop to generate high-frequency signals from a low-frequency reference. They are widely used due to their balance between performance and simplicity.

- **Example Chips**: Analog Devices ADF4351 (35 MHz – 4.4 GHz), TI LMX2572 (up to 7.5 GHz)
- **Use Cases**: General RF applications, drone jamming, signal generation
- **Pros**: Wide tuning range, good phase noise, easy to control

3.5.3.2 Direct Digital Synthesizers (DDS)

DDS modules generate signals using digital counters and DACs. They offer ultra-fine frequency resolution and rapid frequency switching.

- **Example Chips**: AD9910, AD9833
- **Use Cases**: Precision instrumentation, low-power signal sources
- **Pros**: High resolution, low spurious output
- **Cons**: Limited to lower frequencies (< 1 GHz)

3.5.3.3 Hybrid DDS + PLL Synthesizers

These combine the wideband capability of PLLs with the precision of DDS systems. The DDS provides fine control, while the PLL extends the frequency range.

- **Use Cases**: Advanced communication and jamming systems
- **Pros**: Wide frequency coverage with high accuracy

3.5.3.4 SDR-Based Synthesizers

Software-defined radios (SDRs) have built-in synthesizers and support for I/Q modulation. They are highly flexible and suitable for prototyping or adaptive jamming.

- **Example Devices**: Ettus USRP B210, LimeSDR Mini
- **Pros**: Fully software programmable, wide frequency range
- **Cons**: Expensive and may require external computing resources

3.5.3.5 Custom FPGA-Based Synthesizers

High-end systems often use FPGAs to implement custom synthesizer logic for specialized or secure applications. These may combine PLLs, DDS, and other RF blocks.

- **Use Cases**: Military jammers, secure RF systems
- **Pros**: Maximum flexibility and performance
- **Cons**: Complex design, higher development cost

Comparison Table

Type	Frequency Range	Control Interface	Speed	Typical Use Case
PLL-Based	Up to 7.5 GHz	SPI/I2C	Moderate	Jamming, communication
DDS	Up to ~500 MHz	SPI/I2C	Very Fast	Precision signal generation
Hybrid DDS + PLL	Up to 10 GHz	Mixed	Fast	High-performance systems
SDR-Based	Wideband (up to 6 GHz)	USB/Ethernet	Fast	Research, adaptive jamming
FPGA-Based	Custom (up to GHz)	Custom Logic	Very Fast	Military, secure applications

3.6 The Antenna

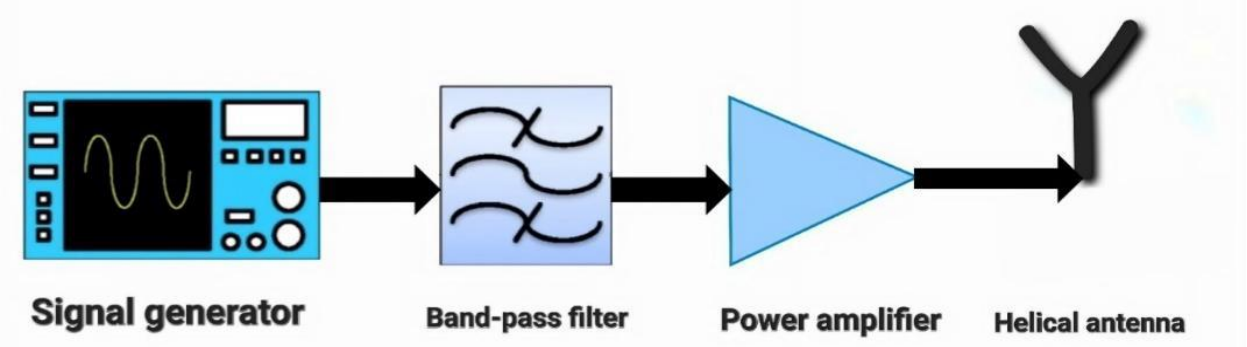


Figure 3. 9 Antenna System

3.6.1 Helical Antenna

Helical antenna is one of the types of broadband antenna which is also called helix antenna. This is one of the most primary, realistic & straightforward antennas which are designed with conducting wire-wound in a helical structure form.

3.6.1.2 What is Helical Antenna?

Definition of helical antenna: The simplest antennas which are used widely in ultra-high frequencies are known as helical antennas, so this antenna works in VHF & UHF ranges. These antennas are designed with conducting wire in a helix shape. This antenna has some unique characteristics like wide bandwidth, high gain & circular polarization.

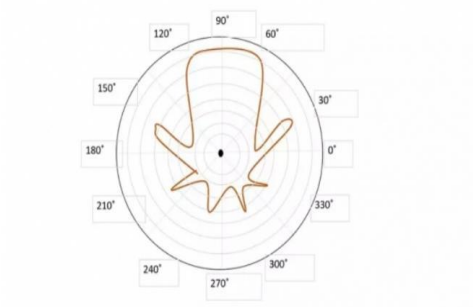
The frequency range of this antenna ranges from 30MHz0-3GHz. This antenna is used in space communication such as satellite relays and also in radio astronomy, wireless networking & satellite communications.

The most popular helical antenna (helix) is a travelling wave antenna in the shape of a corkscrew that produces radiation along the axis of the helix antenna. These helix antennas are referred to as axial-mode helical antennas. The benefits of this helix antenna are it has a wide bandwidth, is easily constructed, has a real impedance input, and can produce circularly polarized fields.

3.6.1.3 Modes in Helical Antenna

A. Normal mode ($C = \pi D << \lambda$)

In the normal mode of radiation, the radiation field is normal to the helix axis and the radiated waves are circularly polarized waves. This mode of radiation is obtained if the dimensions of a helix are small compared to the wavelength. The radiation pattern of this helical antenna is a combination of short dipole and loop antenna.

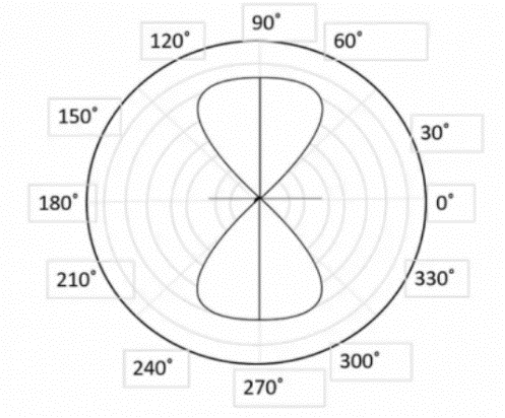


The figure shown is the radiation pattern for the normal mode of radiation in a helical antenna.

It depends upon the values of the diameter of helix D and it turns spacing S. Main problem with this mode of operation is that the radiation efficiency is low and the bandwidth is narrow. Hence it is practically limited and hardly used.

B. Axial mode ($C \approx \lambda$)

In the axial mode of radiation, the radiation is in the end-fire direction along the helical axis and the waves are circular or nearly circular polarized waves. This mode of operation is obtained by raising the circumference to the order of one wavelength (C/λ) and a spacing of approximately $\lambda/4$. The radiation pattern is broad and directional along the axial beam producing minor lobes at oblique angles



The figure shown is the radiation pattern for an axial mode of radiation in the helical antenna.

This antenna, when used for reception, if it is designed for right-handed circularly polarized waves, it will not receive left-handed circularly polarized waves and vice versa. This mode of operation is generated with great ease and is more practically used.

3.6.1.4 Features

The salient features of a helical antenna include the following.

1. This is a simple antenna used for circular polarization.
2. It is utilized in different bands like VHF &UHF.
3. It is mostly used in axial mode.
4. It is not chosen in normal mode where efficiency and beamwidth are small in this mode.
5. Its design is very simple & has maximum directivity.
6. In axial mode, it is a wideband antenna.
7. If the axial ratio is zero then, linear horizontal polarization occurs.
8. If the axial ratio is infinite, then linear vertical polarization occurs.
9. If the axial ratio is one, then circular polarization occurs.

3.6.2 Advantages of helical antenna

The advantages of a helical antenna include the following:

1. Design is simple.
2. Directivity is high.
3. Wide bandwidth.
4. Circular polarization can be obtained.
5. It can be used at VHF and HF bands.
6. Robust construction.
7. When it uses a circularly polarized pattern then it is acceptable through both vertical & horizontal polarized antenna types.
8. High gain

3.6.3 Disadvantages of helical antenna

The disadvantages of a helical antenna include the following:

1. Efficiency mainly depends on the number of turns so, because of the number of turns, the efficiency will be decreased.
2. High cost.

3.6.4 Applications

The applications of helical antennae include the following:

1. These antennas are applicable in satellite & space probe communications because of their circular polarization of the transmitted electromagnetic waves & maximum directivity.
2. A single or array of helical antennas are used for transmitting & receiving VHF signals.
3. Used for satellites at Earth stations.
4. Used for telemetry links through ballistic missiles.
5. Communication can be established between the moon & the earth.
6. Helical antennas are used in many satellites like data relay and weather.
7. This antenna is used for transmitting & receiving VHF waves, especially for ionospheric propagation.
8. It is used for different communications like radio astronomy, space telemetry, satellite, and space.
9. It is used in jamming depending on it’s high gain.

Chapter 4

Artificial Intelligence

Presented by/

A. AHMED SAMIR ZAKI 95855

B. NADA RAAFAT ALI 95905

4.1 Introduction

AI is what computers do to copy how people think. It's about figuring things out and learning from things that happened before. For a long time, people used AI in computer programs, but now it's showing up in lots of products and services. For example, today's cameras can look at pictures and find things because of smart software powered by AI. Experts think we'll see even cooler uses soon, like smart power grids.AI mixes ideas from math with chances (probability), money studies (economics), and step-by-step rules (algorithms) to solve real problems. It also uses ideas from computer science, math, mind understanding (psychology), and language (linguistics). Computer science is important for making algorithms, and math gives ways to solve hard problems.

The idea of smart machines goes back to the 1800s, but it was Alan Turing who talked about a "pretend game" in the middle of the 1900s to check how smart machines could be. But AI only started working well recently because computers got faster and we have more data to train these smart systems.

To understand how AI works, think about why humans are different from other animals. It's because we can notice experiences, learn from them, and use what we learn in new situations. This skill comes from our strong thinking abilities.

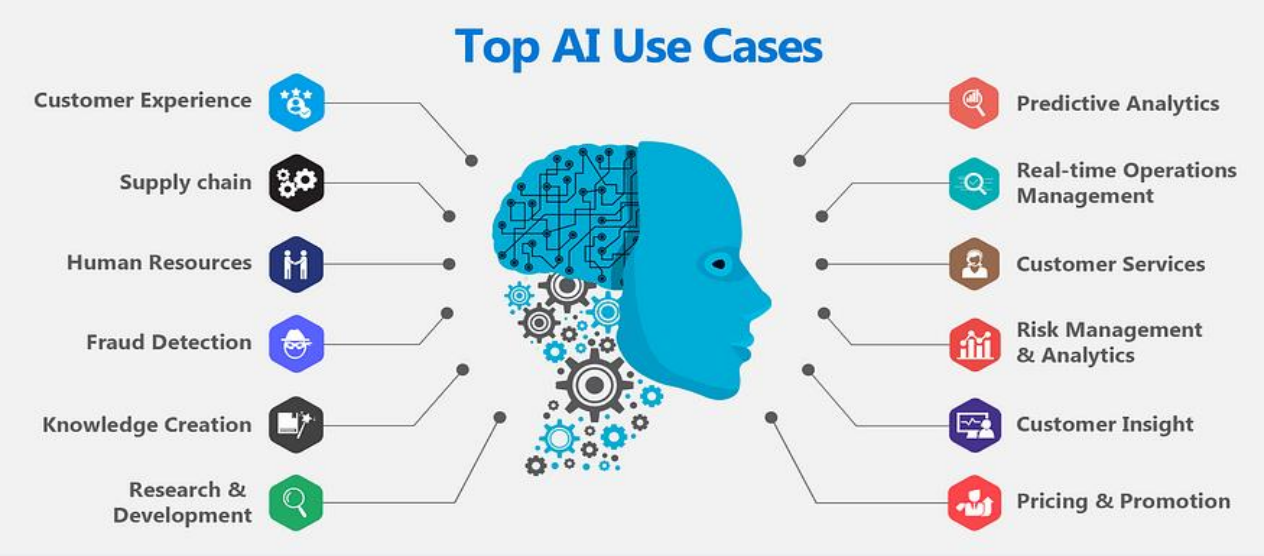


Figure 4. 1 AI uses

4.2 Various fields related to AI

Artificial intelligence is the most popular area of computer science today. Even so, as the Technology and research are developing extremely fast it may become difficult to understand what exactly is new and what is not. Moreover, there are many branches of AI, and each of them uses different algorithms. Therefore, it is important to understand that AI is not one large area, but several interrelated fields. Artificial intelligence (AI) is the broader term for the ability to get computers to mimic human cognition. It can be further divided into two primary branches, Machine Learning (ML), Neural Networks (NN) and Deep Learning (DL) etc. All these are branches of AI and each of them employs different techniques to address various challenges.

Figure 4. 2 Main branches of AI

4.3 Machine learning

A sub discipline of computer science, Machine Learning (ML) is the ability of computers to modify their behavior based on the input data provided to them making them ‘smart.’ In this case, ML uses statistics and probability theory to arrive at its conclusions. Machine learning is a branch of AI where algorithms learn from data without someone telling them exactly what to do. We can split machine learning algorithms into two groups: supervised and unsupervised. It can be used to apply what has been learned from previous data sets to new data sets; it can also find insights from datasets. The goal of machine learning algorithms is to attempt to establish linear and non-linear relationships in a data set. This is done using statistical methods that enable the algorithm to learn from a data set to classify or predict.

4.4 Deep Learning (DL)

Deep learning is a specialized field within machine learning (ML) that comes from research on artificial neural networks (NN). One of the foundational structures in deep learning is the multilayer perceptron (MLP). Deep learning constructs hierarchical representations by extracting higher-level features from lower-level ones. Its primary objective is to develop neural networks that function similarly to the human brain, allowing them to process and interpret distinct types of data, including images, speech, and text. Deep learning systems can analyze extensive human characteristics and behavioral patterns through supervised learning. This process involves:

- Recognizing and interpreting different human emotions and movements.
- Identifying humans and animals in images by analyzing distinctive features, marks, or patterns.
- Distinguishing between different speakers using voice recognition.
- Converting video and audio content into text format.
- Assessing the accuracy of actions, filtering out spam content, and detecting fake activities, such as false statements.

4.5 Neural Networks

Neural networks are modeled after the biological neurons in the human brain and consist of multiple layers of interconnected nodes, known as "neurons." These neurons use mathematical functions to process input data and generate predicted outputs. Like how humans learn from parents, teachers, and peers, artificial neural networks learn through examples. A neural network is typically structured with at least three layers: an input layer, one or more hidden layers, and an output layer. Each layer comprises neurons that receive weighted inputs, process them, and compute an output value.

4.6 How Does AI Work?

Artificial Intelligence (AI) is an advanced yet essential technology, but how exactly does it function? Simply put, AI operates by combining large amounts of data with intelligent processing algorithms. Through these algorithms, AI learns behavioral patterns within the dataset, enabling it to make informed decisions. It is crucial to recognize that AI is not just a single algorithm, it is an

entire machine learning system designed to solve problems and predict outcomes. Let us break down the process step by step:

1. Input

The first stage of AI involves gathering the necessary data for it to function effectively. This data can take various forms, including text, images, or speech. However, it's essential that the AI system can interpret the data correctly. Additionally, defining the context and the desired outcome at this stage ensures accurate results.

2. Processing

During this phase, AI analyzes the input data and determines the appropriate course of action. By leveraging pre-programmed information and previously learned behaviors, AI identifies patterns in real-time data based on its specific technology and application.

3. Data Outcomes

Once AI processes the data, it predicts outcomes. This step helps determine whether the AI's predictions align with expectations or if adjustments are necessary.

4. Adjustments

If the initial predictions are incorrect, AI refines its learning process. It modifies its algorithmic rules or adapts its approach based on past mistakes, ensuring a more accurate response in future iterations. The goal of this phase is to optimize the model and improve accuracy over time.

5. Assessment

In the final stage, AI evaluates its overall performance by analyzing the data, making inferences, and generating predictions. This phase also provides valuable feedback that can be used to further refine the AI model before it runs new iterations of the algorithm.

AI plays a significant role in various industries, particularly in business. However, selecting the right AI technology that aligns with specific business needs is key to achieving optimal results.

4.7 Application Fields of AI

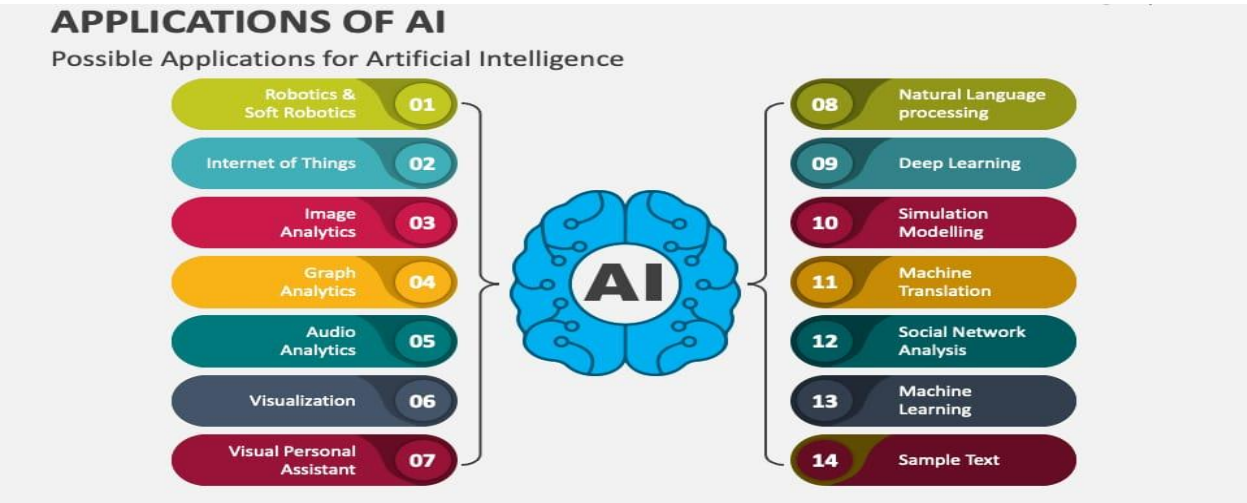


Figure 4. 3 Applications for AI

4.8 The Use of Artificial Intelligence in Our Project

Artificial intelligence (AI) is a powerful and rapidly evolving technology that has changed many different fields around the world. In our project, we use AI in two main ways: vision detection and sound detection. For vision detection, the system uses algorithms to detect various objects, including drones, UAVs, and birds. For sound detection, we created a smart and efficient model that detects the unique sounds of drones, helping the system to find them accurately even when visibility is low or during nighttime conditions.

4.9 Vision detection

4.9.1 Introduction

Vision detection is a part of computer science and artificial intelligence that helps computers see and understand images and videos like humans do. It allows systems to find and recognize objects, people, drones, and movements in pictures or live videos. This helps computers make smart decisions based on what they see. Vision detection uses image processing and advanced algorithms like machine learning and deep neural networks to improve accuracy, even in hard conditions like low light or busy backgrounds. It is used in many important areas such as security and surveillance to detect unauthorized people and drones, self-driving cars to understand their surroundings, smart factories to find product defects, and healthcare to analyze medical images. Vision detection is a key technology today and keeps growing fast with new advances in AI and computing, opening many new possibilities in different fields.

4.9.2 Image Processing

An image is a visual representation of an object, either captured, copied, or stored in electronic form. Mathematically, an image is a two-dimensional function, f(x, y), where x and y represent spatial coordinates. The value of f(x, y) at any given point corresponds to the intensity

of the image at that location. When both the spatial coordinates and intensity values are discrete and finite, the image is referred to as a digital image.

A digital image consists of many small elements known as pixels, each assigned to a specific value and position. Digital image processing involves automated manipulation, enhancement, and interpretation of these visual elements. Its primary purpose is to extract meaningful information and specific characteristics from an image, allowing for more effective analysis and classification. By modifying and enhancing image properties, digital image processing significantly improves the accuracy of results in various applications.

This field plays a crucial role across multiple disciplines, including science and technology, with applications spanning photography, television, robotics, remote sensing, and industrial inspection. It is also a core area of research within engineering and computer science.

Image processing is a technique used to perform various operations on an image to enhance it or extract useful information. As a form of signal processing, it takes an image as input and produces either an improved image or extracted features as output.

Key Steps in Image Processing:

- Image Acquisition: Capturing or importing an image using specialized tools.
- Image Analysis & Manipulation: Processing the image to extract meaningful data.
- Output Generation: Producing either a modified image or a report based on image analysis.

Image processing has long been a foundational area of research, particularly in tasks like text detection and recognition. A wide array of algorithms exists for image processing, and selecting the most suitable one manually can be challenging. To address this, developers have integrated Open-Source Computer Vision (OpenCV), a widely used library containing a large collection of pre-installed image processing algorithms. OpenCV automatically selects the most appropriate algorithm based on the given task, streamlining the process of applying advanced image processing techniques.

4.10 OpenCV

4.10.1 Introduction

OpenCV (Open-Source Computer Vision Library) is a library of programming functions primarily focused on image processing. Available under the open-source Berkeley Software Distribution (BSD) license, OpenCV was initially developed as a research project by Intel. The library offers a wide range of tools to address computer vision challenges, from low-level image processing functions to high-level algorithms for tasks such as face and object detection, feature matching, and tracking. OpenCV is compatible with C++, C, Python, and Java and supports multiple operating systems, including Windows, Linux, macOS, iOS, and Android. In simple terms, OpenCV enables computers to recognize and process objects in videos and images, similarly to how humans recognize them.

4.10.2 Features of OpenCV

OpenCV is equipped with a comprehensive set of features, including:

- Image and video processing algorithms
- Object detection and tracking.
- Facial recognition
- Machine learning and deep learning support
- Camera calibration and 3D reconstruction
- Augmented reality
- Gesture recognition
- Robotics support

4.10.3 Application of OpenCV with Python

By utilizing OpenCV in Python, it becomes possible to easily process images and videos, extracting valuable information through the wide array of functions available in the library. Some common applications include:

A. Image Processing

Image processing is a type of signal processing where the input is an image (e.g., a photograph or video frame), and the output can either be a modified image or a set of characteristics related to the image. OpenCV, a library primarily designed for image processing, is available under the open-source Berkeley Software Distribution license and was originally developed as a research project by Intel. It contains various tools for solving computer vision problems, including low-level image processing functions and high-level algorithms for tasks like face detection, feature matching, and tracking.

Some key image processing techniques available in OpenCV are:

- Image Filtering:
  - Image filtering is a technique used to modify or enhance an image. It can be categorized into two types:
    1. Linear Image Filtering: The output pixel value is a linear combination of neighboring input pixels.
    2. Non-Linear Image Filtering: The output value is not a linear function of the input.
- Image Transformation:

  Image transformation involves generating a "new" image from two or more sources, often focusing on specific features or properties that are more apparent than in the original input. Basic image transformations perform simple arithmetic operations on the image data. One common use is image subtraction, which highlights changes between images captured on different dates. Some common image transformation methods include:
  - Radon Transform: Used to reconstruct images from fan-beam or parallel-beam projection data.
  - Discrete Cosine Transform (DCT): Utilized in image and video compression.

- Discrete Fourier Transform (DFT): Applied in filtering and frequency analysis.
- Wavelet Transform: Used for discrete wavelet analysis, denoising, and image fusion.

B. Object Tracking

Object tracking refers to the process of identifying and following an object (or multiple objects) across a sequence of images. This technique is crucial in various computer vision applications such as surveillance, human-computer interaction, and medical imaging. By tracking the position of objects over time, it allows for continuous monitoring and analysis.

C. Feature Detection

A feature is defined as an "interesting" part of an image that is significant for further analysis. These features serve as the starting point for many computer vision algorithms. Since they are the key elements used in subsequent algorithms, the quality of the overall algorithm heavily depends on the effectiveness of its feature detector. Feature detection involves identifying specific characteristics of a visual stimulus, such as lines, edges, or angles. It is a valuable technique for making local decisions regarding the image's structure and contents.

OpenCV Modules for Image Processing

- CORE Module: Contains basic data structures and functions used by other modules.
- IMGPROC Module: Provides functions for image processing tasks, including linear and non-linear image filtering and geometrical image transformations.
- VIDEO Module: Includes motion estimation and object tracking algorithms.
- ML Module: Offers machine learning interfaces.
- HighGUI Module: Provides basic I/O interfaces and multi-platform windowing capabilities.

D. Face Detection

Face detection is a process where an input image is analyzed to locate faces. Once a face is found, the image can be processed further to crop and extract the face for various applications. OpenCV provides a built-in face detection method known as the Haar Cascade classifier. When given an input image (e.g., from a camera or live video), the face detector evaluates the image to classify regions as either a face or non-face. The classifier uses an XML file *(haarcascade_frontalface_default.xml)* to determine the classification. In OpenCV 2.4.10, the XML file is typically located in the directory path: *opencv/sources/data/haarcascades.*

Figure 4. 4 Face Detection

E. Face Recognition

Face recognition is the subsequent phase after detecting a face. In this process, the detected face image is compared to stored images in a face database to identify the individual. OpenCV provides a built-in face detection framework that works with high accuracy (90-95%) on clear images. However, the system may encounter challenges in cases where the person is wearing glasses, or the image is blurred.

4.11 OpenCV for Object Detection

It is significantly easier to write code for images captured under controlled lighting conditions than for images taken in dynamic environments where lighting cannot be guaranteed. When you can control both the environment and lighting, it becomes possible to hardcode parameters like:

- Amount of blurring
- Edge detection boundaries
- Thresholding limits.

This approach uses your knowledge of the specific environment, allowing you to write code that is customized to it, rather than having to account for every potential variation. However, it is important to note that controlling the environment and lighting is not always feasible.

For this project, OpenCV with Python will be used for image preprocessing and object detection.

The steps are as follows:

- Image Acquisition: Capture the initial image using a laptop with a camera.
- Image Preprocessing: Rotate the image to portrait mode and convert it to grayscale.

- Smoothing: Apply GaussianBlur to smooth the image.
- Edge Detection: Use the Canny edge detector to detect edges in the image.
- Morphological Operations: Apply operations like thickening, thinning, and filling to modify object shapes.
- Contour Detection: Identify all contours in the image using the edge information.
- ROI Selection: Select only the contours corresponding to the Region of Interest (ROI) for drone detection.
- Perspective Warping: Adjust the perspective to get a better view of the detected drone
- Object Recognition: Use YOLO or another object detection model to recognize and track the detected drone.

4.12 Ultralytics and Its Role in AI Detection Models

Ultralytics is a leading software development company focused on advancing machine learning and computer vision technologies. Known for its contributions to the YOLO (You Only Look Once) family of models, Ultralytics has optimized and developed state-of-the-art solutions for real-time object detection and tracking. Their implementations of YOLO, such as YOLOv5, have significantly improved speed, accuracy, and ease of use, making it an ideal choice for a variety of applications, including drone detection. With its open-source framework, Ultralytics provides robust tools and pre-trained models that can be easily adapted to specific use cases, such as detecting UAVs in complex environments. Through continuous development, they have enhanced model performance, reduced inference time, and simplified the training process, allowing users to deploy new AI systems for object detection and classification with minimal effort.

4.13 Detection Algorithms in Computer Vision

Detection algorithms are essential in computer vision for identifying and localizing objects in images and videos. These algorithms can be broadly categorized into one-stage and two-stage detection methods. Each category has its strengths and weaknesses, making them suitable for different applications based on speed, accuracy, and computational efficiency.

4.13.1 Two-Stage Detection Algorithms

Two-stage detectors first generate regional proposals and then classify and refine them in a second stage. These models prioritize accuracy over speed, making them suitable for applications requiring precise object detection, such as medical imaging and security surveillance.

A. Features of Two-Stage Detectors:

- Use a Region Proposal Network (RPN) to suggest **possible** object locations.
- Perform classification and bounding box regression in a second step.
- Offer higher accuracy compared to one-stage detectors.
- Slower due to the additional processing stage.

B. Popular Two-Stage Algorithms:

- **R-CNN** (Region-based Convolutional Neural Network): Extracts region proposals and classifies them using a deep network, but it is computationally expensive.
- **Fast R-CNN**: An improved version of R-CNN that uses ROI (Region of Interest) Pooling to speed up processing.

ROI (Region of Interest) Pooling is a technique used in Fast R-CNN and Faster R-CNN to extract fixed-size feature maps from different-sized proposals (ROIs) generated by RPN. It allows the network to efficiently classify objects and refine their bounding boxes.

RPN (Region Proposal Network) is a neural network used in Faster R-CNN to identify potential object regions in an image. It generates Anchors (boxes of varied sizes), classify them as "object" or "background," and adjusts their coordinates for better accuracy. RPN helps reduce unnecessary.

- **Faster R-CNN:** Introduces an RPN (Region Proposal Network) to generate region proposals efficiently, making it significantly faster than previous R-CNN versions while maintaining high accuracy. The RPN replaces traditional methods like Selective Search and Edge Boxes, which were used to generate object proposals before RPN. This improvement makes object detection.

Selective Search: Segments the image into regions based on color, texture, and size, then merges similar regions to form object proposals. It's accurate but slow.

Edge Boxes: Detects object proposals based on edge density, assuming objects have well-defined boundaries. Faster than Selective Search but less precise.

- **Mask R-CNN**: An extension of Faster R-CNN that adds an additional mask prediction branch for instance segmentation.

4.13.2 One-Stage Detection Algorithms

One-stage detectors directly predict object locations and class labels from the input image in a single pass through the network. These models prioritize speed over accuracy, making them suitable for real-time applications such as autonomous driving, Surveillance of air vehicles and control systems.

A. Features of One-Stage Detectors:

- Perform object classification and localization in a single step.
- Use a thick sampling approach over the image to detect objects.
- Achieve high speed due to fewer computational operations.
- Generally, they have lower accuracy compared to two-stage detectors.

B. Popular One-Stage Algorithms:

- **YOLO** (You Only Look Once): One of the fastest object detection algorithms, capable of detecting objects in real-time. It divides the image into a grid and predicts bounding boxes and class probabilities simultaneously.
- **SSD** (Single Shot MultiBox Detector): Uses multiple feature maps at different scales to detect objects with varying sizes, improving detection performance.
- **RetinaNet**: A one-stage detector that incorporates a Focal Loss function to improve detection of small and hard-to-detect objects.

4.14 Why YOLO?

YOLO is the better choice for real-time object detection due to these reasons:

YOLO processes the entire image in a single forward pass, making it significantly faster than SSD, especially for real-time applications. It achieves higher mean Average Precision (mAP) than SSD, especially for small object detection like UAVs. It is better at detecting objects across different scales and backgrounds due to its advanced anchor-free approach (in newer versions). YOLO is optimized for speed and efficiency, making it suitable for edge devices and real-time drone detection.

4.15 YOLO

With the rapid advancement of artificial intelligence and computer vision technologies, real-time object detection and tracking have become crucial in various applications, such as security and surveillance. Among the available detection algorithms, You Only Look Once (YOLO) is one of the most efficient approaches, as it balances accuracy and processing speed.
YOLO is a deep learning-based object detection model that analyzes an entire image in a single pass, allowing it to extract spatial and temporal information quickly and accurately. Unlike traditional methods that rely on multiple processing steps, YOLO divides the image into a grid and directly predicts object locations and bounding boxes in a single operation.

In this project, YOLO is employed for high-precision drones and UAVs detection and tracking, with enhancements such as image augmentation and noise reduction techniques to improve performance under different environmental and imaging conditions.

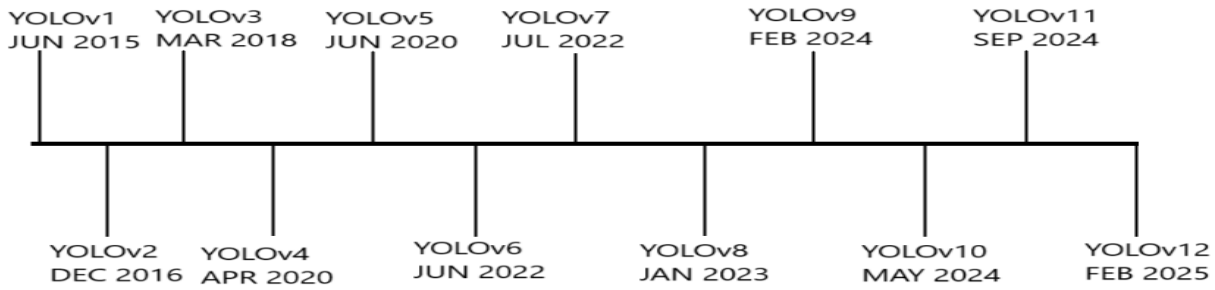


Figure 4. 5 YOLO Evolution Timeline

4.16 How YOLO Works

YOLO works by looking at the entire image in one go, instead of scanning it in parts like other methods. It divides the image into a grid, and each grid cell is responsible for detecting objects whose center is in that cell. For each cell, the model predicts several bounding boxes that include the object's center, size, and a confidence score to show how sure it is about the object. It also predicts the class of the object inside the box. Once it has made all predictions, YOLO uses a technique called Non-Maximum Suppression (NMS) to remove duplicate boxes. It keeps only the boxes with the highest confidence score, using a measure called Intersection over Union (IoU) to decide which boxes are too close to each other.

In the end, YOLO gives a list of the detected objects, including what they are, how confident the model is, and where they are in the image. This method makes YOLO great for real-time applications like detecting and tracking drones.

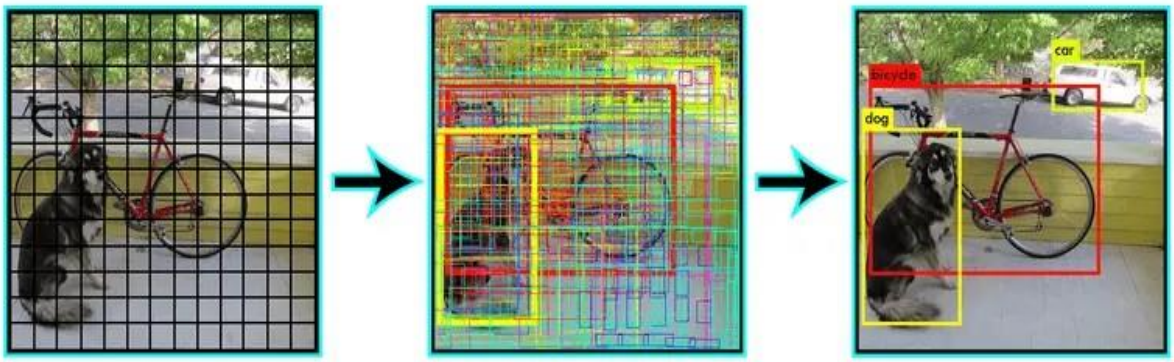


Figure 4. 6 How YOLO Works

4.17 YOLO versions in project

In our project, we tested several versions of YOLO (You Only Look Once) for optimal drone and UAV detection, specifically YOLOv9, YOLOv11, and YOLOv12. Here's an overview of each version's performance:

YOLOv9 introduced significant improvements in the backbone architecture, leading to an increase in accuracy compared to previous versions. However, these enhancements came at the cost of speed, with the model being slower than its predecessors. Additionally, YOLOv9 had some issues when detecting multiple objects simultaneously, making it less ideal for crowded or complex scenes. Despite these challenges, it remained a solid choice for its high accuracy in drone detection.

YOLOv11 provided a better balance between accuracy and speed compared to YOLOv9. It outperformed YOLOv10 (which we did not consider for our project due to its reduced accuracy and performance trade-offs) by offering better detection precision while being faster. YOLOv11 became the version we continued working with, as it allowed us to achieve more accurate results without sacrificing real-time performance.

YOLOv12 brought significant improvements, achieving the highest accuracy and speed among the versions we tested. Based on benchmarks from the COCO dataset, YOLOv12 achieved a mean Average Precision (mAP) of 40.6%, making it the most accurate and efficient version for our application. Additionally, YOLOv12 performed well in night-time image detection and processed images at an impressive rate of 1.64 milliseconds per image. These advancements made YOLOv12 the best choice for our system.

We decided not to use YOLOv10 due to its lower accuracy and slower performance compared to YOLOv9 and YOLOv11. The graphs and performance metrics from Ultralytics clearly show that YOLOv12 outperforms all previous versions in terms of both speed and accuracy as shown in Fig.4.7 and Fig.4.8

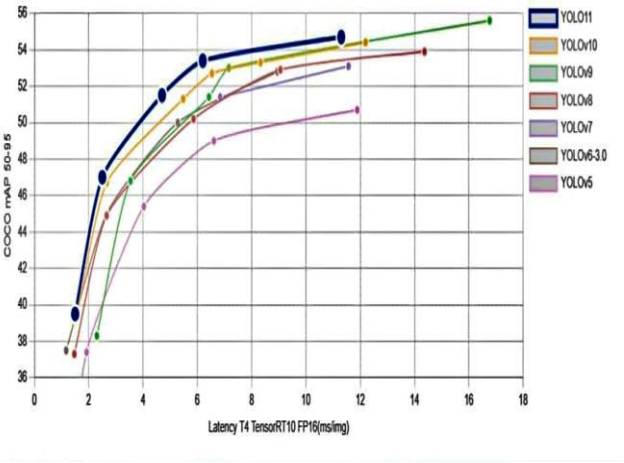


Figure 4. 8 YOLO versions in Project

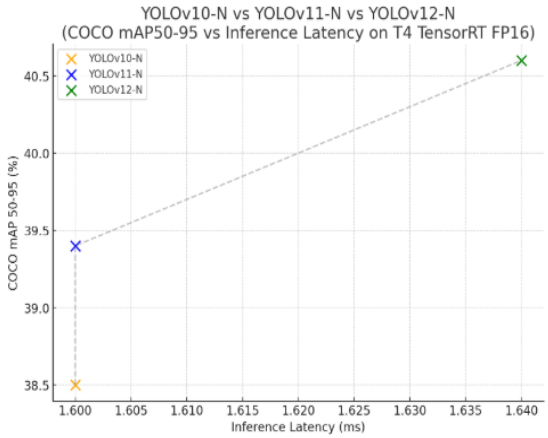


Figure 4. 7 YOLO versions in Project

4.17.1 Dataset Preparation

A custom dataset of **23,401** images was created to train the YOLOv9 model, covering three distinct categories: drones, UAVs, and birds. Contains **8,368** images of drones, **7760** images of UAVs and **7273** images of birds.

Each image was manually labeled, ensuring accurate and high-quality bounding box annotations.

The dataset was carefully organized to include:

- Different drone models, UAVs, and birds were included in the dataset.
- Diverse environmental conditions such as varying lighting, backgrounds, and altitudes.

This comprehensive dataset ensures that the model can generalize well across real-world aerial scenarios.

We have decided to classify the targets into three classes for these reasons: to avoid the high confidence score problem which affects classification and to consume power while jamming on drones and UAVs by stopping jamming when the system detected birds

4.17.2 Data Splitting Strategy

To evaluate the model's performance and avoid overfitting, the dataset was divided into three subsets:

- Training Set: **16,509** images, used to train YOLO versions to recognize drones, UAVs, and birds.
- Validation: **3473** images, used for fine-tuning hyperparameters and monitoring model performance.
- Testing Set: **3419** images, reserved for independent evaluation to assess the model's real-world effectiveness.

This structured division ensured balanced training and reliable performance assessment.

4.17.3 Model Training

The training process was conducted using Ultralytics implementation of YOLO versions, a high-level framework built on PyTorch that simplifies training and deployment. This framework provided an efficient and user-friendly environment for dataset management, model training, and evaluation.

Key advantages of using Ultralytics included:

- Simplified dataset handling and training process.
- Pre-configured hyperparameter optimization.
- Seamless integration with different hardware platforms.

The model was trained on a Windows platform with GPU acceleration, reducing training time significantly over 100 epochs.

4.17.4 Hyperparameter Tuning and Optimization

To improve model accuracy and stability, various hyperparameters were fine-tuned, including:

- Learning rate: Adjusted to balance training speed and stability.
- Batch size: Optimized for memory efficiency and better convergence.
- Optimization algorithms: Fine-tuned to enhance model performance.

4.17.5 Equations

1. Precision (P)

Precision measures how many of the predicted positive cases are actually correct.
it is calculated as:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Posites\ (TP) + False\ Positives\ (FP)}$$

A high precision means that when the model predicts a detection, it is likely correct.

where:

True Positive (TP): Correctly detected objects (detections that match actual objects).

False Positive (FP): Incorrectly detected objects (detections that do not correspond to actual objects).

False Negative (FN): Objects that were present but not detected by the model.

True Negative (TN): Background areas correctly identified as not containing objects (less relevant in object detection, more common in classification tasks).

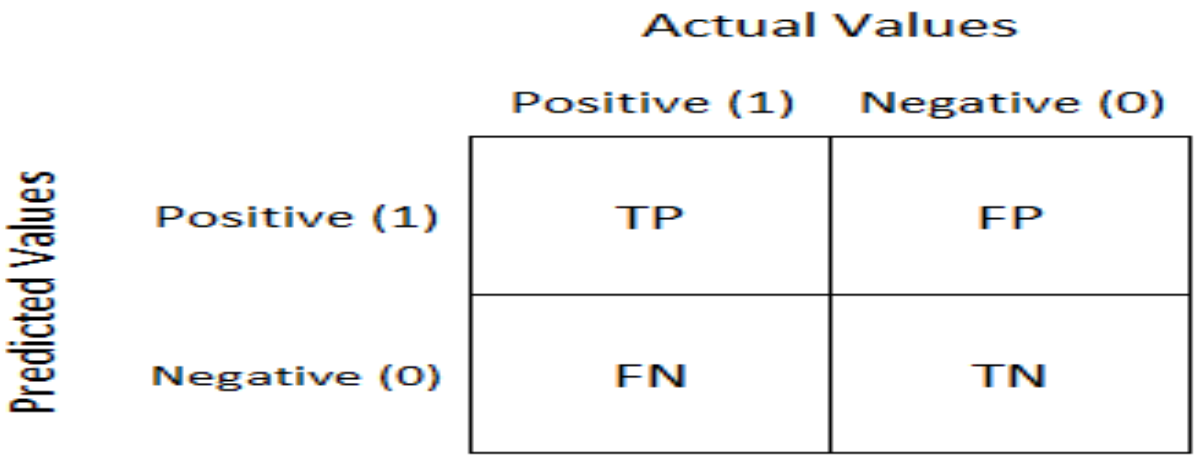


Figure 4. 9 Performance Evaluation Using Confusion Matrix

2. Recall (R)

Recalling measures show how many of the actual positive cases were detected by the model.

It is calculated as:

$$Recall = \frac{True\ Positives\ (TP)}{True\ Posites\ (TP) + False\ Negative\ (FN)}$$

A high recall means that the model detects most of the actual objects but may also include false positives.

3. Mean Average Precision (mAP)

mAP is the main metric used for evaluating object detection models. It is the average of the Average Precision (AP) across all classes.

(a) Average Precision (AP)

AP is the area under the Precision-Recall curve:

$$AP = \int_0^1 P(R)dR$$

This integral is approximated by summing precision values at different recall levels.

(b) Mean Average Precision @50 (mAP@0.5)

mAP@50 is the mean Average Precision when Intersection over Union (IoU) threshold = 0.5.

This means that a detection is considered correct if it overlaps with the ground truth by at least 50%.

$$mAP_{50} = \frac{1}{N}\sum_{i=1}^{N} AP_i$$

where N is the number of object classes.

(c) mAP@50:95 (mAP@0.5:0.95)

mAP@50:95 is the average of mAP computed at different IoU thresholds from 0.5 to 0.95 in steps of 0.05:

$$mAP_{50:95} = \frac{1}{10}\sum_{t=0.5}^{0.95} mAP_t$$

This provides a more comprehensive evaluation of the model's performance across different IoU thresholds

Intersection over Union (IoU) is a key metric used to evaluate object detection models. It measures how much a predicted bounding box overlaps with the ground truth bounding box.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Where:

- Area of Overlap = The intersection between the predicted and ground truth bounding boxes.
- Area of Union = The total area covered by both bounding boxes combined.

4. Confidence

Confidence is the probability assigned by the model to a detected object, indicating how certain the model is about its prediction. It is typically used as a threshold for filtering detections.

- A high confidence threshold (e.g., 0.9) reduces FP but may increase FN.
- A low confidence threshold (e.g., 0.3) increases recall but may reduce precision.

Each detected object has a confidence score, and predictions are usually kept only if their confidence is above a set threshold.

4.18 YOLOv9

YOLOv9 is a modern version of the You Only Look Once (YOLO) algorithm, specifically designed for real-time object detection with an emphasis on high-speed performance and precision. It features significant improvements in neural network architecture and feature extraction techniques, allowing it to maintain a strong balance between detection accuracy and computational efficiency. These advancements enable YOLOv9 to detect even small objects with greater precision, making them suitable for complex environments with varied object sizes. The model excels in handling dynamic scenes and can perform well in tasks requiring rapid processing and high accuracy, such as surveillance, autonomous driving, and real-time object tracking. Its ability to process images at remarkable speeds while maintaining quality makes YOLOv9 a robust choice for real-world applications.

4.18.1 Architecture

- **Backbone:**
  The Backbone is responsible for extracting essential features from the input image using a deep convolutional neural network (CNN). It analyzes the image to identify key patterns like edges and shapes, enabling fast and efficient feature extraction.
- **Neck:**
  The Neck aggregates and refines features from the Backbone, improving multi-scale feature fusion. Techniques like Feature Pyramid Networks (FPN) help combine features at different scales, enhancing the model's ability to detect objects of many sizes.
- **Head:**
  The Head makes the final predictions, including bounding boxes, class labels, and confidence scores. YOLOv9 uses Anchor Boxes to predict multiple objects in different regions of the image at once.

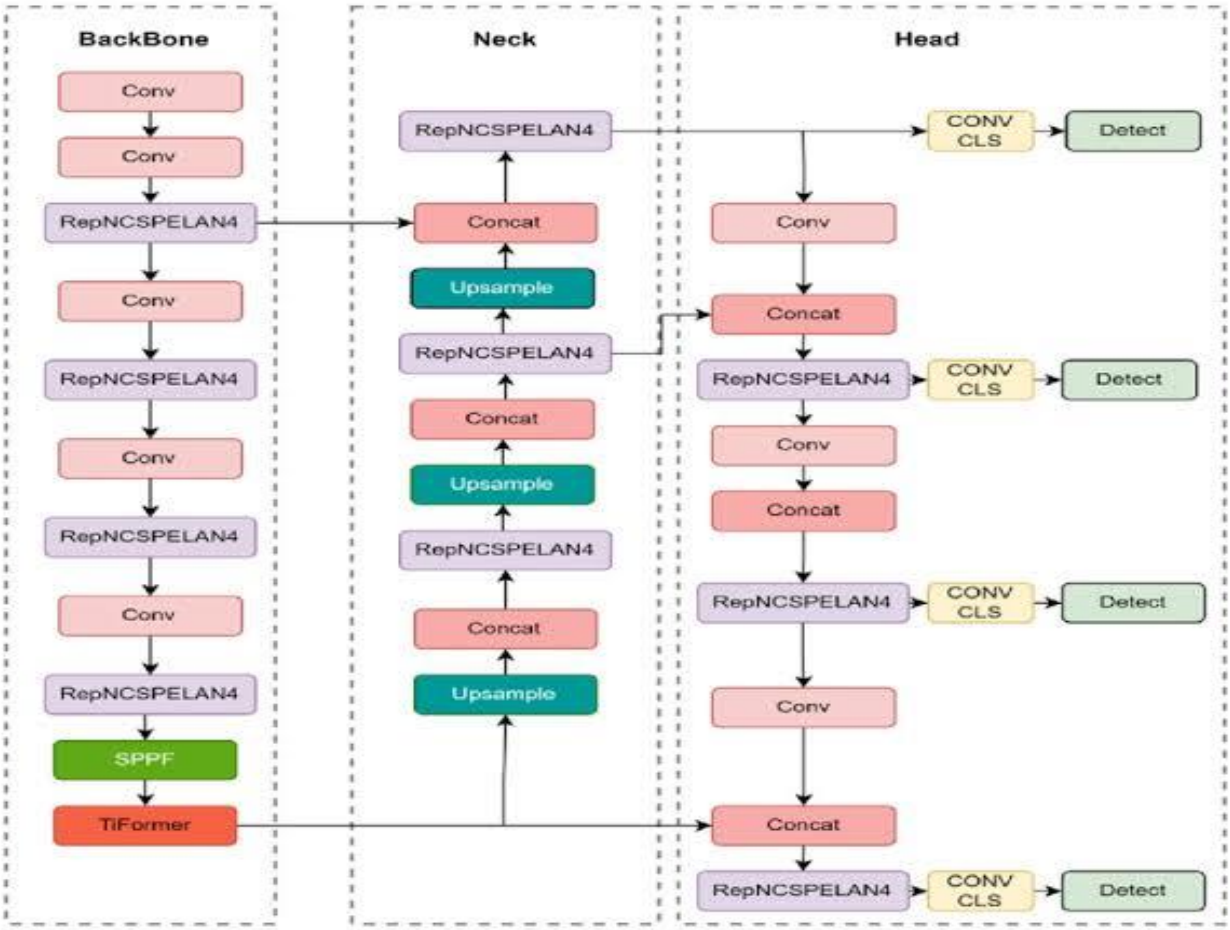


Figure 4. 10 YOLOv9 Architecture

4.18.2 Key Features

- Real-time Performance: YOLOv9 is optimized for fast object detection, ensuring quick results without compromising on accuracy.
- Small Object Detection**:** It has enhanced ability to detect smaller objects, which is crucial in crowded or complex scenes.
- Multi-scale Detection with PANet and FPN: The use of PANet and FPN improves the model's capability to detect objects of many sizes effectively.
- Anchor Boxes: YOLOv9 uses anchor boxes for more accurate bounding box predictions, improving their detection of objects with different shapes and sizes.

4.18.3 Dataset and Training

In this part of the project, we used the same dataset described previously in section 2.17.2, maintaining the exact same data split into training, validation, and testing sets. No modifications were made to the dataset or its distribution, ensuring consistency across all experiments conducted with different YOLO versions.

Furthermore, the training procedure followed here is identical to the one detailed in section 4.17.3. This includes the same model configuration, number of training epochs, learning rate, and all other training parameters, as well as the tools and libraries used throughout the training process. By keeping the dataset and training method unchanged, we aimed to create a fair and reliable comparison of YOLOv9's performance relative to other versions under identical conditions.

One key observation during the training phase was that a single epoch took approximately 45 minutes to complete. This was a notably long duration, especially considering the model was trained over 100 epochs, resulting in an overall lengthy training time. However, this extended training period yielded high accuracy, which indicates that the model benefited from the intensive

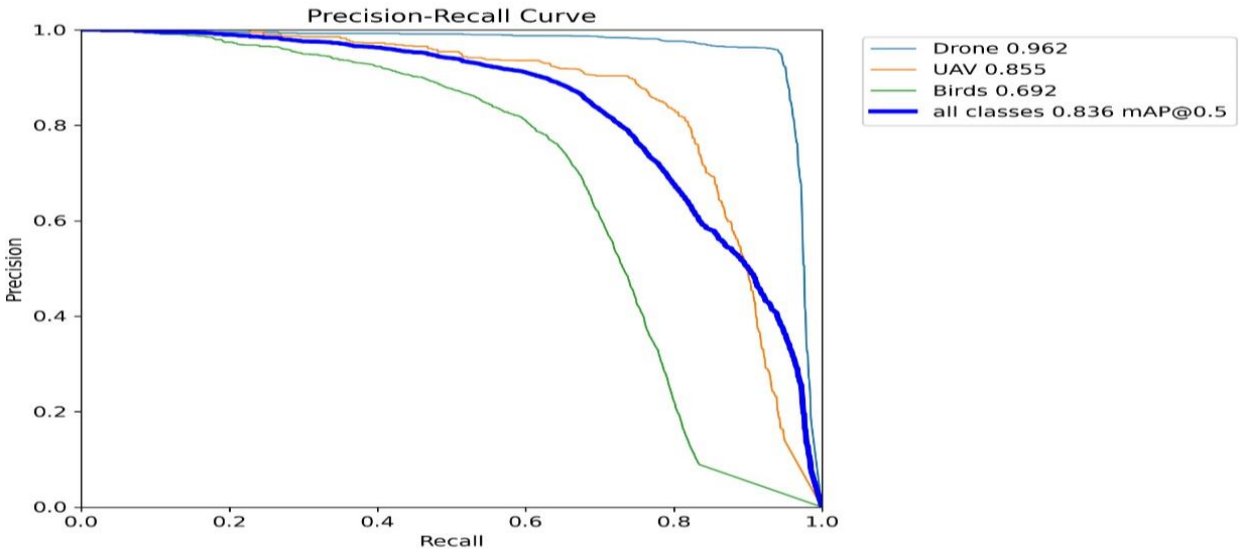


Figure 4. 11 Precision-Recall Curve

training despite the time cost.

4.18.4 Results

A. Precision-Recall Curve.

Fig.4.11. represents Precision-Recall curve for the model, showing the trade-off between the model's precision (its ability to correctly identify drones and UAVs) and its recall (its ability to

find all drones and UAVs in the images). For **drones**, the model has a high precision of **0.962** with a corresponding mAP@0.5 value of **0.962.** This indicates that the model is highly accurate in detecting drones with good bounding box overlaps. For **UAVs**, the model achieves a precision of

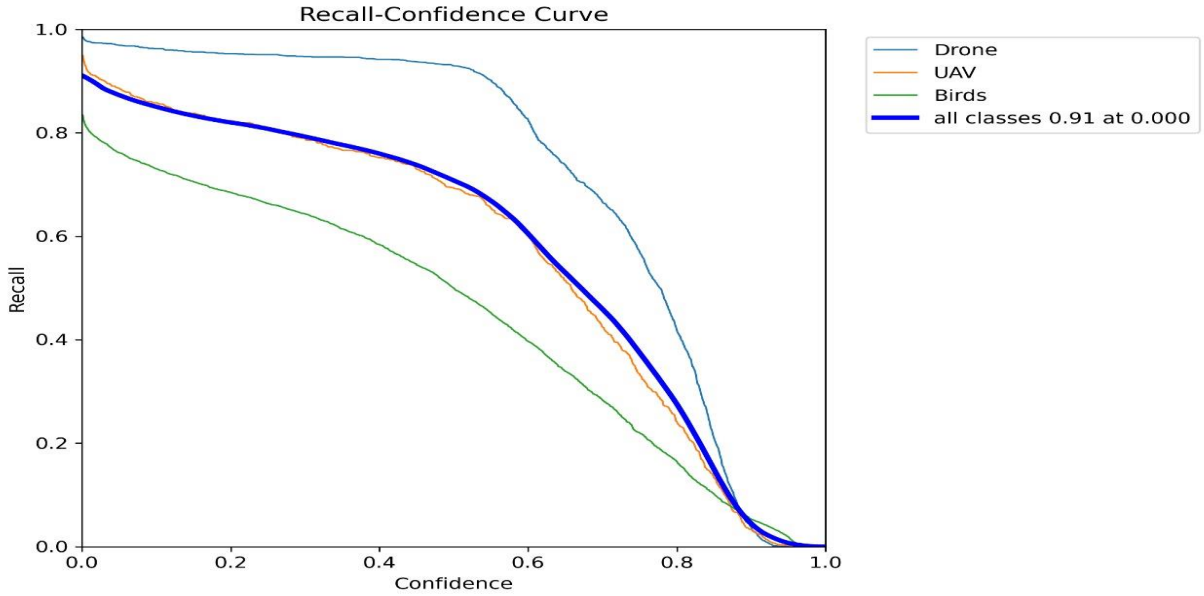


Figure 4. 12 Recall-Confidence Curve

0.855 with a corresponding mAP@0.5 value of **0.855.** For **Birds**, the model achieves a precision of **0.692** with a corresponding mAP@0.5 value of **0.692**.

The **mAP@0.5 for all classes** is **0.836**, representing the average precision for detecting both drones and UAVs when the IoU threshold is set to 0.5.
This means the model performs well overall, correctly identifying and locating both classes with bounding boxes overlapping the ground truth by at least 50.

B. Recall-Confidence Curve.

This Fig.4.12 represents Recall-Confidence Curve which illustrates the trade-off between the model's recall (its ability to correctly detect objects) and confidence (its certainty in making detections).

The model achieves a high recall of **0.91** across all classes at a confidence threshold of 0.00, indicating that it can identify **91%** of objects in the dataset without applying strict confidence

constraints. Among the individual classes: Drones have the highest recall, followed by UAVs, while Birds exhibit the lowest recall performance. As confidence increases, recall decreases, showing the typical behavior where stricter thresholds filter out more detections.

C. Precision-confidence Curve.

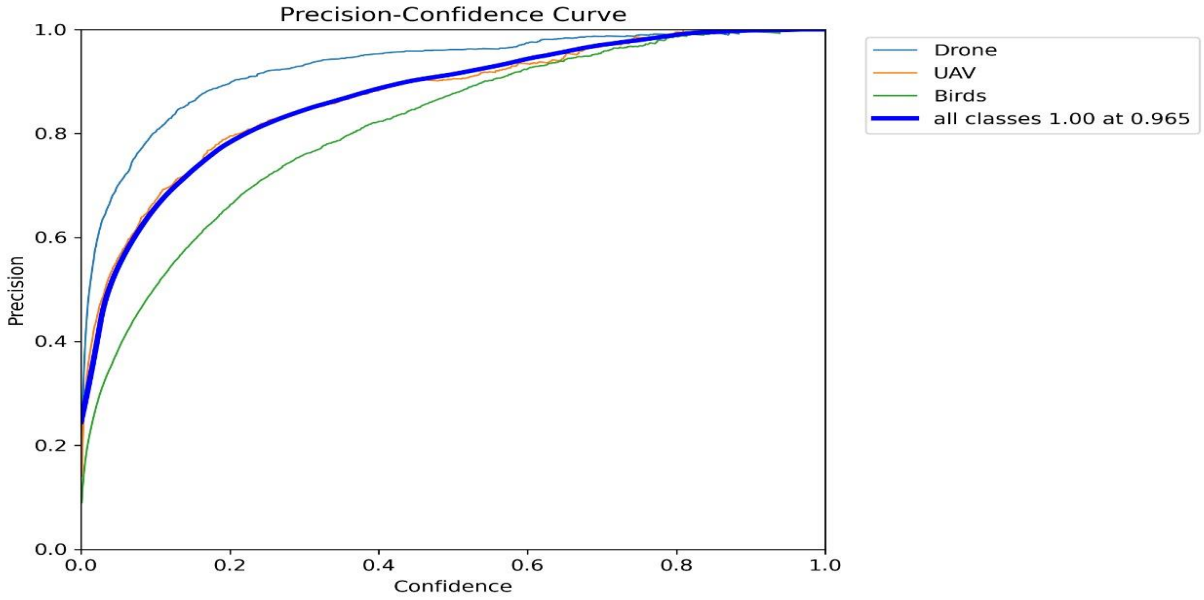


Figure 4. 13 Precision-Confidence Curve

This Fig.4.13 shows Precision-Confidence Curve which represents the relationship between precision (the proportion of true positive detections among all detections) and confidence (the model's certainty in its predictions).

This curve provides insight into the model's ability to balance recall and precision in different scenarios, making it useful for applications where detecting as many objects as possible is critical.

This curve illustrates the trade-off between precision and confidence, showing how stricter confidence thresholds impact the accuracy of detections.

The model demonstrates high precision, achieving a precision of **1.00** at a confidence threshold of 0.965, indicating that when the model's confidence in its detections surpasses **0.965**, the predictions are always correct.

Among the different classes, Drones exhibit the highest precision, closely followed by UAVs, while Birds have the lowest precision at lower confidence levels.

Although the mAP@0.5 value is not explicitly provided, the curve suggests that the average precision across all confidence levels is close to 1.00, highlighting the model's effectiveness in achieving accurate detections with a high degree of reliability.

D. F1-Confidence Curve.

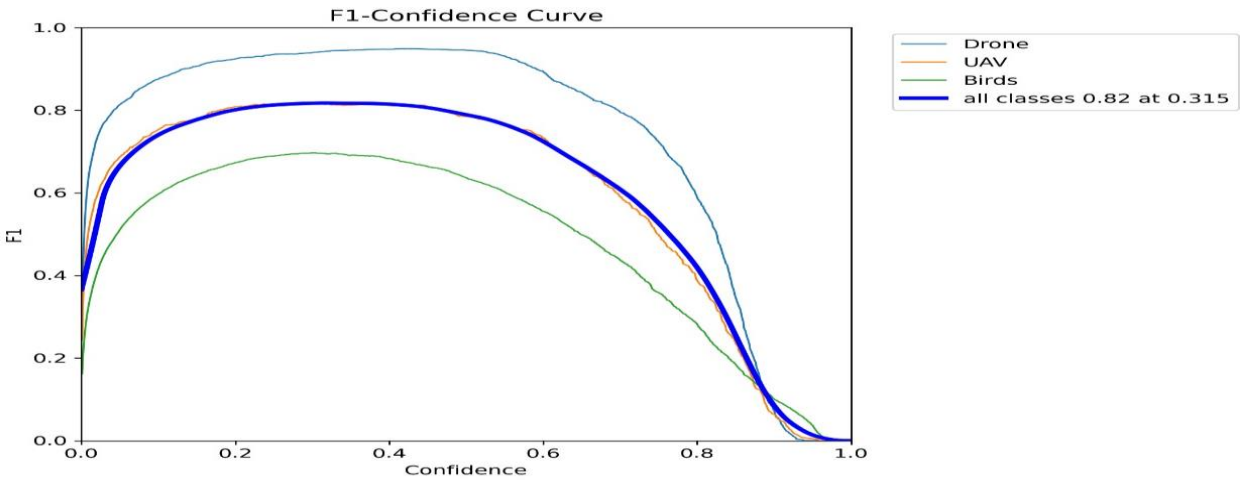


Figure 4. 14 F1-Confidence Curve

This Fig.4.14 represents F1-Confidence Curve, which illustrates the variation of the F1-score as a function of the confidence threshold. The F1-score, being the harmonic means of precision and recall, represents the trade-off between false positives and false negatives at different confidence levels.
Among the different classes:

The F1-score for Drones is the highest, peaking around **0.9**, while the F1-score for UAVs is slightly lower but closely follows the Drone curve. In contrast, the F1-score for Birds is significantly lower than that of both Drones and UAVs. The overall best F1-score achieved across all classes is **0.82** at a confidence threshold of **0.315**, indicating that this threshold provides the optimal balance between precision and recall for the model's performance.

E. Confusion Matrix.

This Fig.4.15 represents a confusion matrix which is a table used to evaluate the performance of a classification model by comparing its predicted labels with the actual ground truth labels. When comparing each class against the background, the matrix helps assess how well the model differentiates objects from the background. This binary classification approach provides insights into the model's ability to correctly detect each class while minimizing false detections

(TPR): This value represents the True Positive Rate, also known as recall, which is 96%, 80% and 71% for drones, UAVs and Birds, respectively.
This indicates the percentage of actual drones, UAVs and Birds in the images that were correctly identified by the model.

(FPR): This value represents the False Positive Rate, which is 6% ,8% and 86% for drones, UAVs and Birds, respectively. This indicates the percentage of times the model incorrectly identified something as a drone or UAV when it wasn't.

(FNR): This value represents the False Negative Rate, which is 3% ,18% and 29% for drones, UAVs and Birds, respectively. This indicates the percentage of times the model failed to identify drones, UAVs and Birds.

(TNR): This value represents the True Negative Rate, which is 0% for all classes. This indicates that the model never correctly identified the absence of drones, UAVs or Birds when they didn't exist.

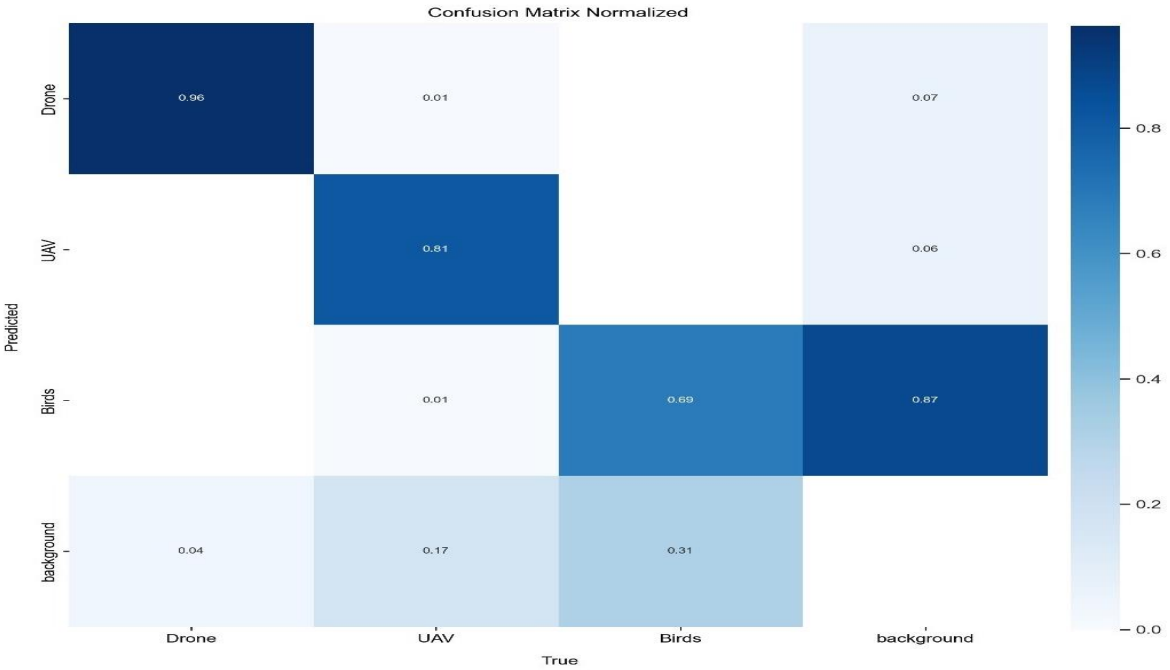


Figure 4. 15 Confusion Matrix

F. Performance Evaluation during Object Detection Model Training

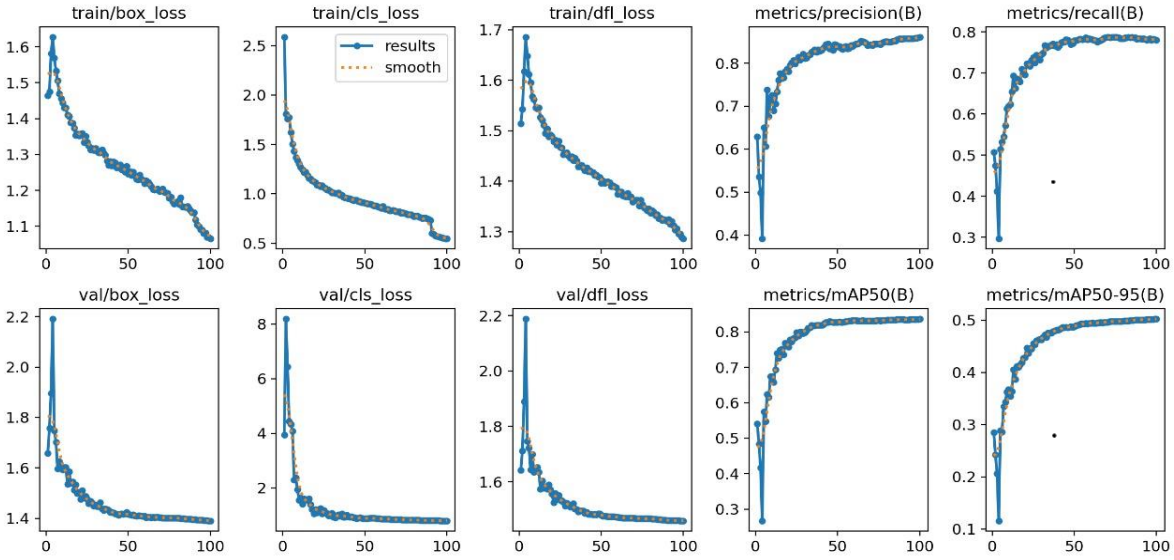


Figure 4. 16 Performance Evaluation

Model Performance Analysis

This section presents an analysis of the training and validation metrics for a YOLO-based object detection model. The evaluation focuses on loss reduction during training and key performance metrics related to precision, recall, and mean Average Precision (mAP).

Loss Metrics.

Loss functions play a critical role in model optimization, and their reduction over time indicates effective learning.

1. Bounding Box Loss (Train & Validation): The training box loss starts at approximately **1.6** and steadily decreases to less than **1.0**, indicating an improvement in bounding box regression.
   Similarly, the validation box loss decreases from **2.2** to less than **1.4**, showing a consistent learning pattern between training and validation phases.
2. Classification Loss (Train & Validation): Initially, the training classification loss is **2.5**, but it rapidly drops below **0.5**, demonstrating that the model effectively differentiates between object classes.
   In contrast, the validation classification loss starts at **8** and decreases to less than **2**, which, while showing improvement, remains higher than the training loss.
3. DFL (Distribution-Focused Learning) Loss: The training DFL loss decreases from **1.7** to near **1.0**, and the validation DFL loss drops from **2.2** to **~1.2**, reflecting enhanced localization accuracy.

Performance Metrics

To evaluate the effectiveness of the trained model, several key performance indicators are analyzed:

1. Precision: The model achieves a final precision of approximately **0.8**, meaning **80%** of its detections are correct.
   This suggests a low rate of false positives.
2. Recall: The recall value reaches ~ **0.78**, indicating that the model detects approximately **78%** of all actual objects present in the dataset.
3. Mean Average Precision (mAP@0.5)**:** The model attains an mAP@0.5 score of ~ **0.88**, demonstrating strong detection performance when using a 50% Intersection over Union (IoU) threshold. It indicates that the model is well-optimized for object detection at a moderate overlapping threshold.
4. Mean Average Precision at Varying IoU Thresholds (mAP@0.5:0.95): The performance declines to ~ 0.50 when evaluated across a range of IoU thresholds from 0.5 to 0.95. This suggests that while the model is highly effective under moderate conditions, its precision decreases for stricter IoU requirements.

4.18.5 Evaluate Results: (Drones, UAVs and Birds)

Figure 4. 17 Results: (Drones)

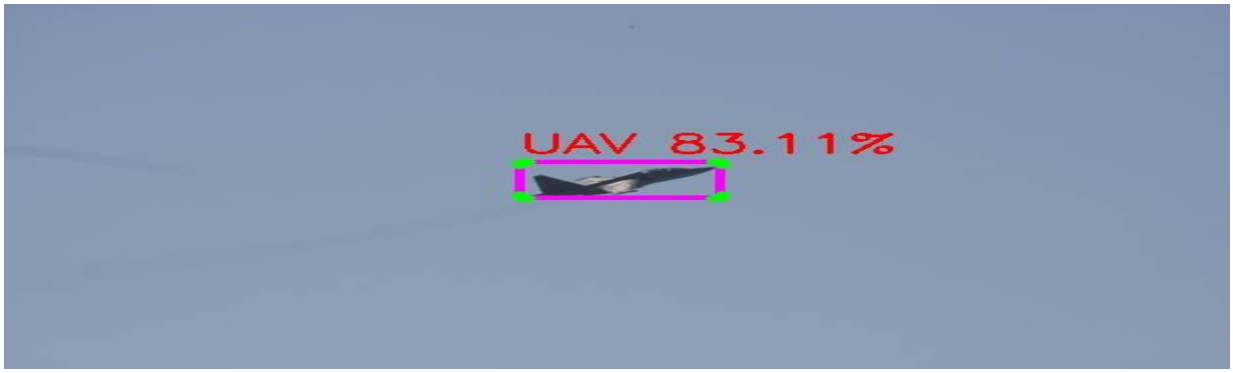


Figure 4. 18 Results: (Uavs)

Figure 4. 19 Results: (Drones, UAVs, Birds)

4.19 YOLOv11

YOLOv11 is a major advancement in the YOLO series, featuring refined architecture and advanced deep learning techniques for improved detection in complex environments. With enhanced feature extraction, Transformer-based layers, and efficient filtering, it achieves greater accuracy, especially for overlapping or similar objects. Its adaptability makes it well-suited for applications like aerial surveillance, security, and real-time video analysis.

2.19.1 Architecture:

- **Backbone:**
  The Backbone in YOLOv11 is a deep neural network that enhances feature extraction for better small-object detection and complex pattern recognition.
- **Neck:**
  The Neck in YOLOv11 enhances feature fusion across scales using advanced techniques like PANet and FPN, improving detection performance for objects of varied sizes.
- **Head:**
  The Head makes the final predictions—like bounding boxes, class labels, and confidence scores—using a better version of Anchor Boxes to handle overlapping and crowded objects more accurately.

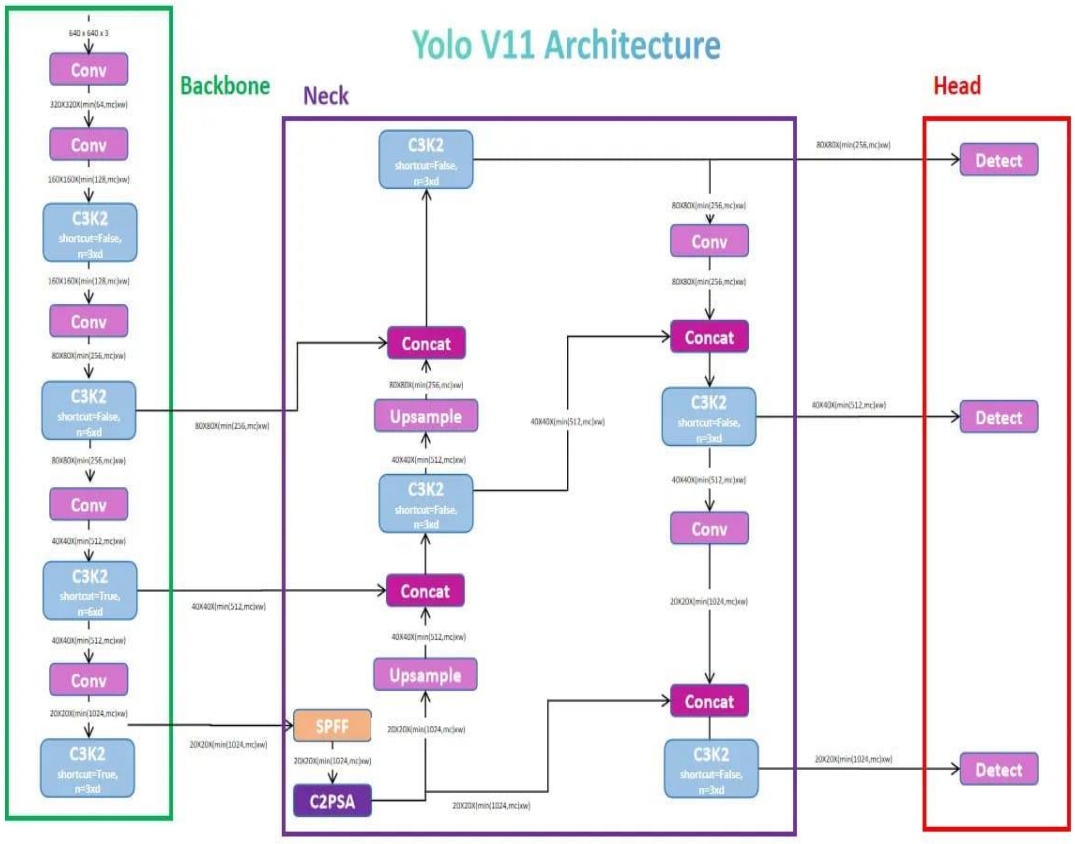


Figure 4. 20 YOLOv11 Architecture

4.19.2 Key Features

- Improved Backbone: YOLOv11 features a more advanced backbone for better feature extraction, enhancing accuracy, especially for smaller and complex objects.
- Multi-scale Detection: YOLOv11 also integrates PANet and FPN, further enhancing multi-scale feature fusion for improved detection across different object sizes.
- Transformer-based Layers: The addition of Transformer-based layers improves feature aggregation and object classification in complex scenes.
- Optimized Anchor Boxes: YOLOv11's optimized anchor boxes allow for better handling of overlapping and crowded objects.
- Higher Accuracy in Complex Environments: YOLOv11 performs exceptionally in detecting objects in tricky situations, such as obstructions and overlaps, with enhanced precision.

4.19.3 Dataset and Training

After evaluating the performance of YOLOv9, YOLOv11 was selected to further enhance detection accuracy by focusing exclusively on drones and UAVs, thus reducing the incidence of false alarms caused by birds. YOLOv11 incorporates advancements in model architecture, including improvements in the handling of small objects and objects in cluttered environments. These enhancements make YOLOv11 particularly suitable for detecting drones in complex aerial scenarios where background noise and overlapping objects are common. By optimizing model architecture, YOLOv11 provides more precise and reliable results, making it ideal for real-time detection applications in UAV monitoring systems.

In this part of the project, we used the same dataset described previously in section 2.17.2, maintaining the exact same data split into training, validation, and testing sets. No modifications were made to the dataset or its distribution, ensuring consistency across all experiments conducted with different YOLO versions.

Furthermore, the training procedure followed here is identical to the one detailed in section 2.17.3. This includes the same model configuration, number of training epochs, learning rate, and all other training parameters, as well as the tools and libraries used throughout the training process. By keeping the dataset and training method unchanged, we aimed to create a fair and reliable comparison of YOLOv11's performance relative to other versions under identical conditions.

An important observation during the training of YOLOv11 was that a single epoch took approximately 25 minutes, which is significantly faster than the 45 minutes per epoch observed with YOLOv9. This supports the claim that YOLOv11 is more efficient in terms of training speed. However, this raises a critical question regarding whether the reduction in training time any notable impact on accuracy had, which we will explore in the subsequent evaluation sections.

4.19.4 Results

A. Precision-Recall Curve.

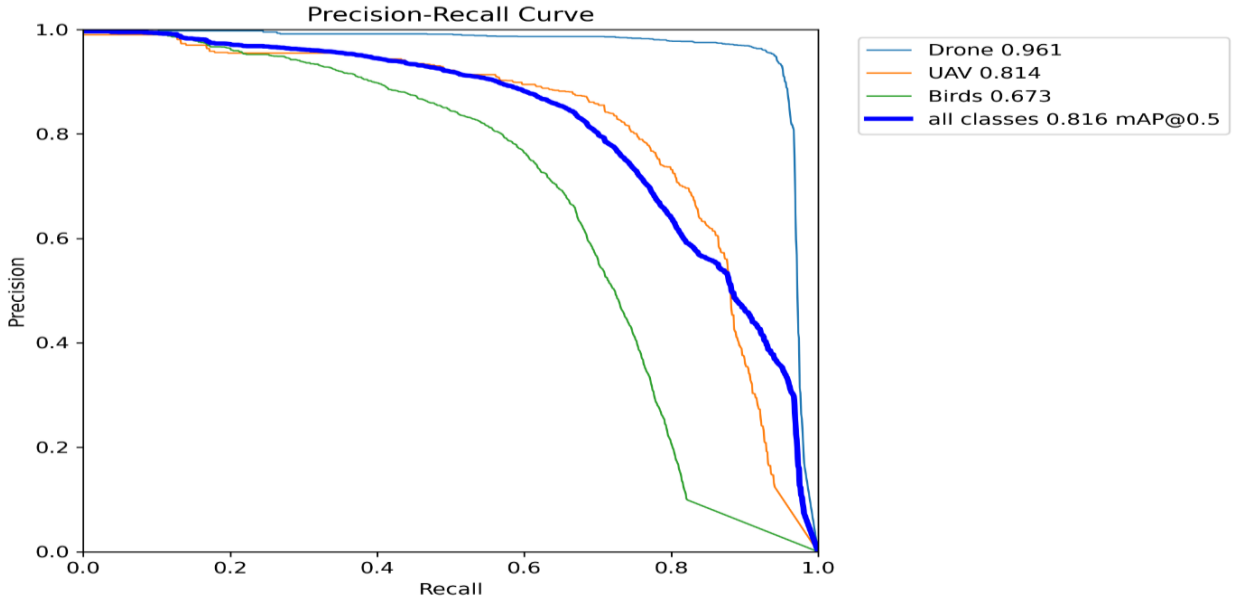


Figure 4. 21 Precision-Recall Curve

This Fig.4.21 represents a precision-recall curve for the model, showing the trade-off between the model's precision (its ability to correctly identify drones and UAVs) and its recall (its ability to find all drones and UAVs in the images). For **drones**, the model has a high precision of **0.961** with a corresponding mAP@0.5 value of **0.961.**

This indicates that the model is highly accurate in detecting drones with good bounding box overlaps.

For **UAVs**, the model achieves a precision of **0.814** with a corresponding mAP@0.5 value of **0.814.** For **Birds**, the model achieves a precision of **0.673** with a corresponding mAP@0.5 value of **0.673.**

The **mAP@0.5 for all classes** is **0.836,** representing the average precision for detecting both drones and UAVs when the IoU threshold is set to 0.5.

This means the model performs well overall, correctly identifying and locating both classes with bounding boxes overlapping the ground truth by at least 50%.

B. Recall-confidence Curve.

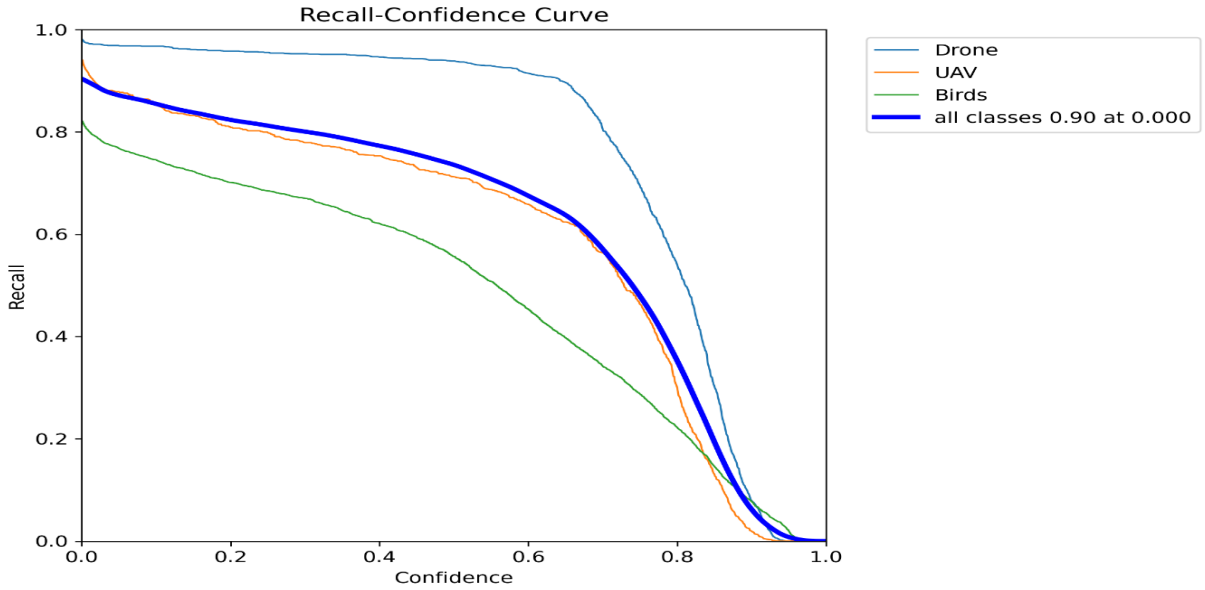


Figure 4. 22 Recall - Confidence Curve

This Fig.4.22 illustrates the trade-off between the model's recall (its ability to correctly detect objects) and confidence (its certainty in making detections).

The model achieves a high recall of **0.90** across all classes at a confidence threshold of **0.00**, indicating that it can identify **90%** of objects in the dataset without applying strict confidence constraints.

Among the individual classes:

Drones have the highest recall, followed by UAVs, while Birds exhibit the lowest recall performance. As confidence increases, recall decreases, showing the typical behavior where stricter thresholds filter out more detections.

This curve provides insight into the model’s ability to balance recall and precision in different scenarios, making it useful for applications where detecting as many objects as possible is critical.

C. Precision-confidence Curve.

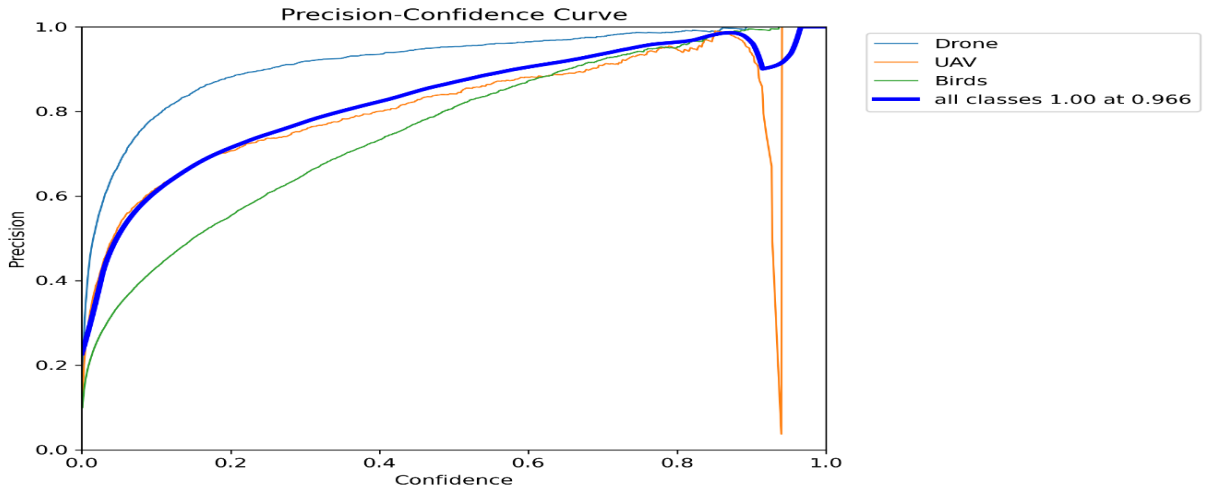


Figure 4. 23 Precision- Confidence Curve

This Fig.4.23 represents the relationship between precision (the proportion of true positive detections among all detections) and confidence (the model's certainty in its predictions).This curve illustrates the trade-off between precision and confidence, showing how stricter confidence thresholds impact the accuracy of detections. The model demonstrates high precision, achieving a precision of **1.00** at a confidence threshold of **0.966**, indicating that when the model's confidence in its detections surpasses **0.966**, the predictions are almost always correct.

Among the different classes, Drones exhibit the highest precision, closely followed by UAVs, while Birds have the lowest precision at lower confidence levels. Although the mAP@0.5 value is not explicitly provided, the curve suggests that the average precision across all confidence levels is close to 1.0, highlighting the model's effectiveness in achieving accurate detections with a high degree of reliability.

D. F1-Confidence Curve.

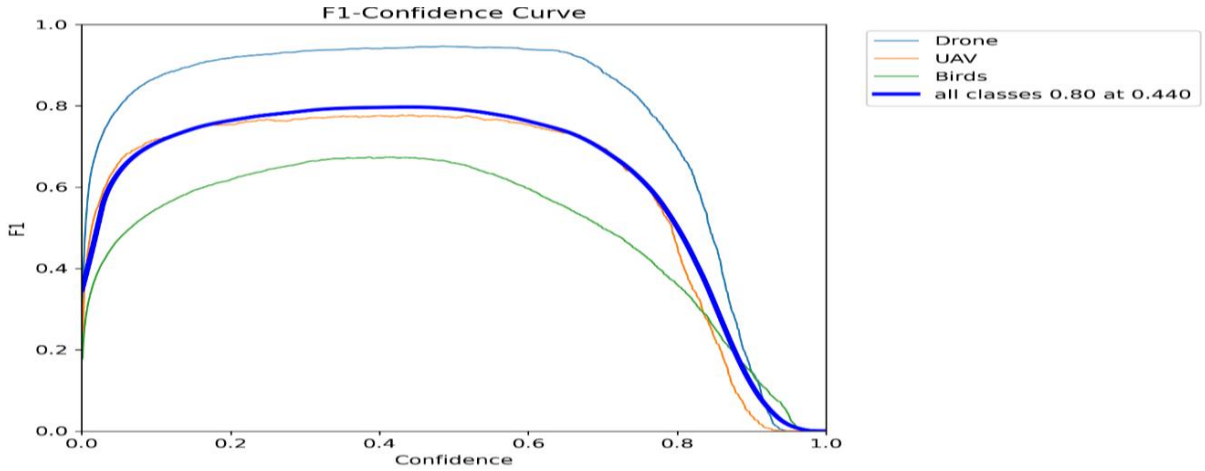


Figure 4. 24 F1- Confidence Curve

This Fig.4.24 represents F1-Confidence Curve, which illustrates the variation of the F1-score as a function of the confidence threshold. The F1-score, being the harmonic means of precision and recall, represents the trade-off between false positives and false negatives at different confidence levels.

Among the different classes:

The F1-score for Drones is the highest, peaking around **0.9**, while the F1-score for UAVs is slightly lower but closely follows the Drone curve. In contrast, the F1-score for Birds is significantly lower than that of both Drones and UAVs.

The overall best F1-score achieved across all classes is **0.82** at a confidence threshold of **0.315**, indicating that this threshold provides the optimal balance between precision and recall for the model's performance.

E. Confusion matrix.

This Fig.4.25 represents a confusion matrix which is a table used to evaluate the performance of a classification model by comparing its predicted labels with the actual ground truth labels. When comparing each class against the background, the matrix helps assess how well the model differentiates objects from the background.

This binary classification approach provides insights into the model's ability to correctly detect each class while minimizing false detections.

(TPR): This value represents the True Positive Rate, also known as recall, which is 96%, 80% and 71% for drones, UAVs and Birds, respectively.

This indicates the percentage of actual drones, UAVs and Birds in the images that were correctly identified by the model.

(FPR): This value represents the False Positive Rate, which is 6% ,8% and 86% for drones, UAVs and Birds, respectively.
This indicates the percentage of times the model incorrectly identified something as a drone or UAV when it wasn't.

(FNR): This value represents the False Negative Rate, which is 3% ,18% and 29% for drones, UAVs and Birds, respectively.
This indicates the percentage of times the model failed to identify drones, UAVs and Birds.

(TNR): This value represents the True Negative Rate, which is 0% for all classes. This indicates that the model never correctly identified the absence of drones, UAVs or Birds when they didn't exist.

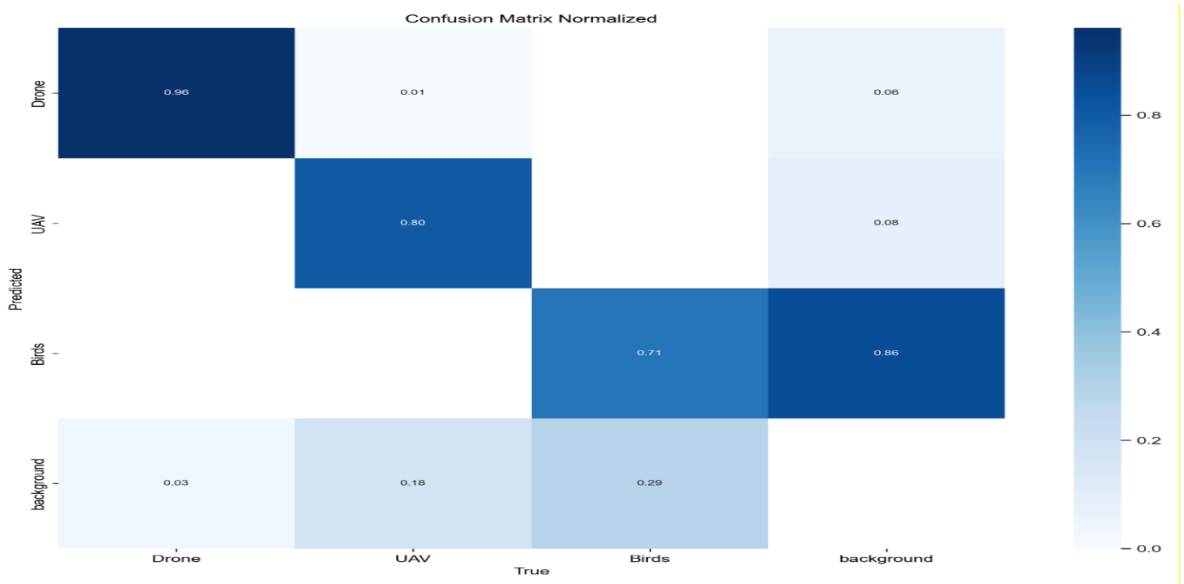


Figure 4. 25 Confusion Matrix

F. Performance Evaluation during Object Detection Model Training.

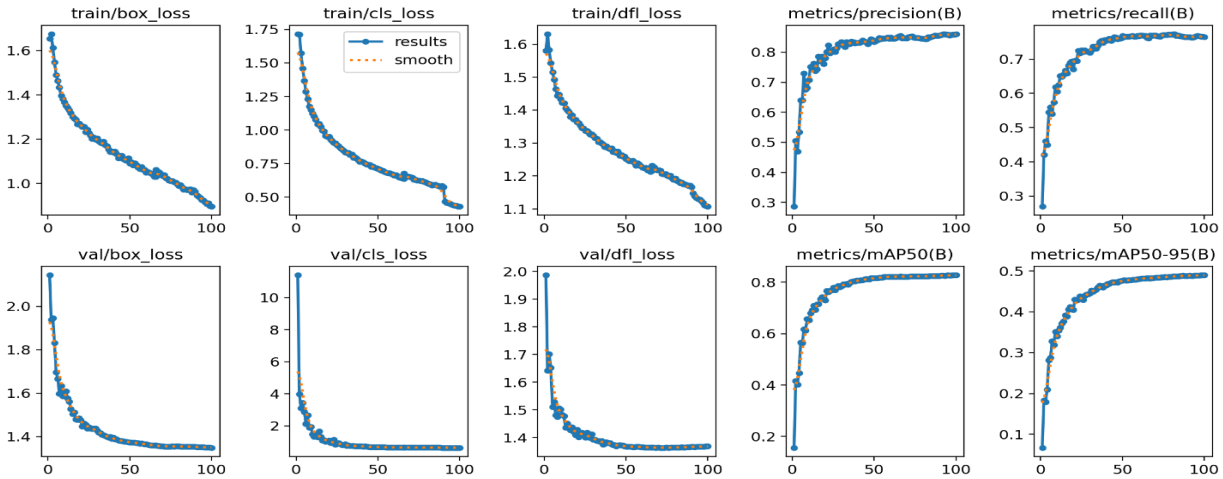


Figure 4. 26 Performance Evaluation

Model Performance Analysis

This section presents an analysis of the training and validation metrics for a YOLO-based object detection model. The evaluation focuses on loss reduction during training and key performance metrics related to precision, recall, and mean Average Precision (mAP).

Loss Metrics.

Loss functions play a critical role in model optimization, and their reduction over time indicates effective learning.

1. Bounding Box Loss (Train & Validation): The training box loss starts at approximately **1.7** and steadily decreases near to **0**, indicating an improvement in bounding box regression.
   Similarly, the validation box loss decreases from ~ **2.2** to < **1.3**, showing a consistent learning pattern between training and validation phases.
2. Classification Loss (Train & Validation): Initially, the training classification loss is **1.7**, but it rapidly drops below **0.5**, demonstrating that the model effectively differentiates between object classes.
   In contrast, the validation classification loss starts at **12** and decreases to less than **2**, which, while showing improvement, remains higher than the training loss.
3. DFL (Distribution-Focused Learning) Loss: The training DFL loss decreases from **1.7** to less than **1.0**, and the validation DFL loss drops from **2** to less than **1.4**, reflecting enhanced localization accuracy.

Performance Metrics

To evaluate the effectiveness of the trained model, several key performance indicators are analyzed:

4. Precision: The model achieves a final precision of **0.90**, meaning **90%** of its detections are correct.
   This suggests a low rate of false positives.
5. Recall: The recall value reaches ~ **0.78**, indicating that the model detects approximately **78%** of all actual objects present in the dataset.
6. Mean Average Precision (mAP@0.5)**:** The model attains an mAP@0.5 score of ~ **0.88**, demonstrating strong detection performance when using a 50% Intersection over Union (IoU) threshold. It indicates that the model is well-optimized for object detection at a moderate overlapping threshold.
7. Mean Average Precision at Varying IoU Thresholds (mAP@0.5:0.95): The performance declines to **0.5** when evaluated across a range of IoU thresholds from 0.5 to 0.95. This suggests that while the model is highly effective under moderate conditions, its precision decreases for stricter IoU requirements.

4.19.5 Evaluate results: (Drones, UAVs and Birds)

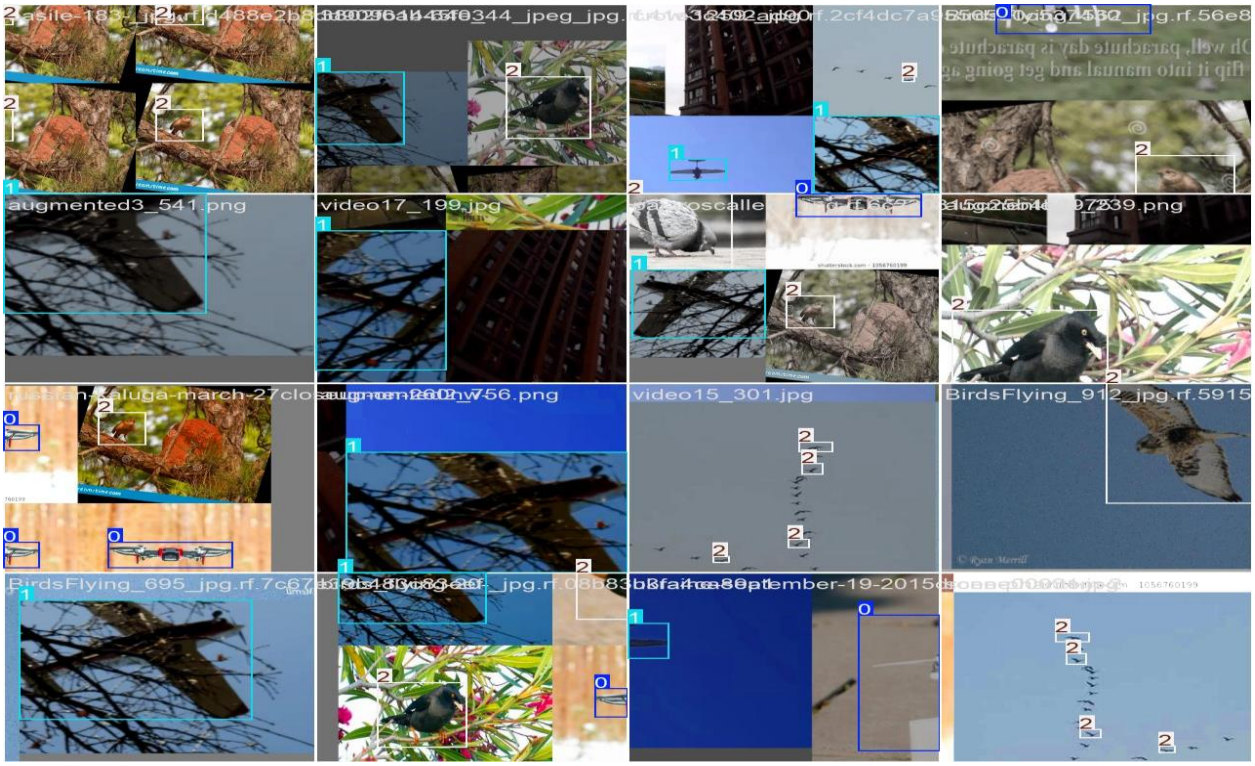


Figure 4. 27 Results: (Drones, Uavs, Birds)

Figure 4. 29 Results: (Drones, Uavs, Birds)

4.20 YOLOv12

YOLOv12 is the newest version of YOLO. It's faster and more accurate than the older versions. It uses a better design to find objects clearly, even in hard situations like dark images or when many objects appear together. It has a stronger backbone, better mixing features, and uses attention to focus on the important parts of the image. That's why it works well for real-time tasks like drone detection and night monitoring.

4.20.1 Architecture

• Backbone:

The YOLOv12 backbone features an improved R-ELAN architecture, which strengthens multi-scale feature extraction while maintaining computational efficiency. It captures more detailed spatial information through deeper layers and optimized convolutional blocks.

• Neck:

YOLOv12's neck incorporates R-ELAN+A2 Modules, which enhance feature aggregation across different scales. The design emphasizes better up sampling and concatenation strategies, leading to superior multi-scale detection capabilities.

• Head:

The head in YOLOv12 uses something called Flush Attention, which helps the model focus better on the important parts of the image. This makes it more accurate, especially when there are many objects close together or when the image is complicated.

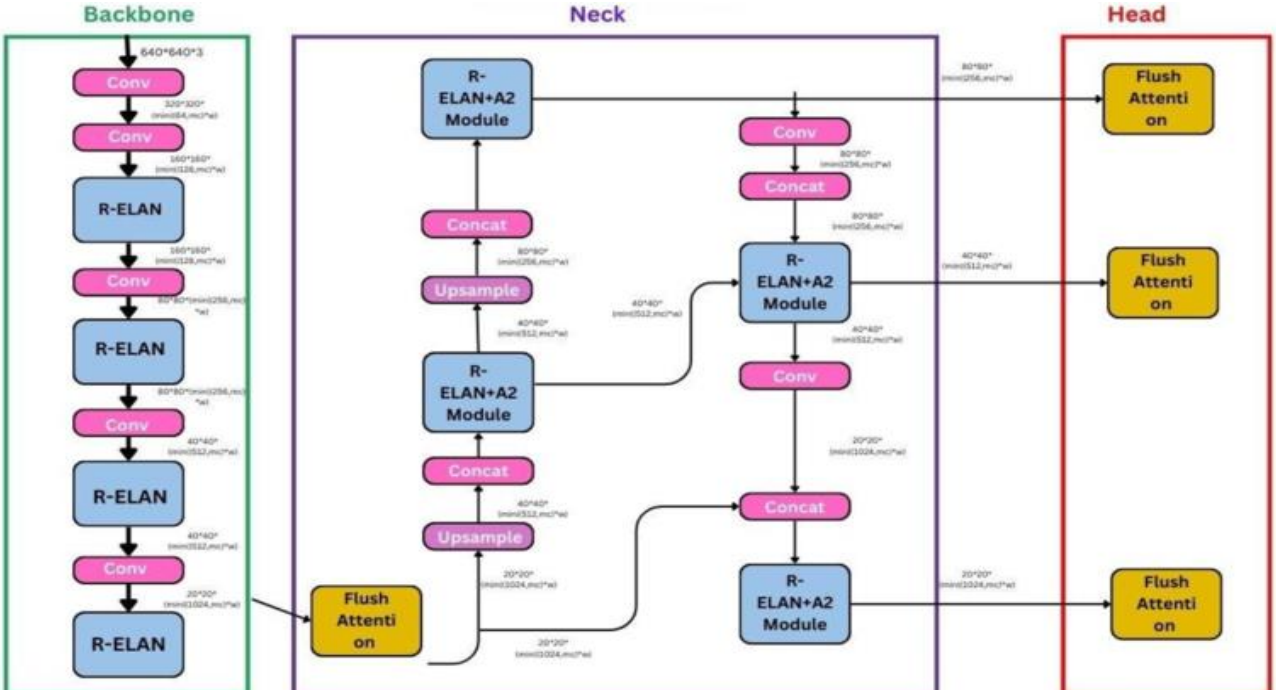


Figure 4. 30 YOLOv12 Architecture

4.20.2 Key Features

- Stronger Backbone: Uses an improved R-ELAN backbone to detect small and distant objects more accurately.
- Better Feature Fusion: A2 modules help combine features from different layers for clearer object detection.
- Flush Attention Head: Focuses on the important parts of the image, improving results in crowded or dark scenes.
- High Accuracy: Achieved 40.6% mAP on the COCO dataset—better than previous YOLO versions.
- Faster Processing: Analyzes each image in about 1.64 ms, making it ideal for real-time tasks.
- Great in Low Light: Performs well in dark or night-time conditions.

4.20.3 Dataset and Training

After working with YOLOv11, YOLOv12 was adopted in our project to take advantage of its higher accuracy and faster processing speed. YOLOv12 showed significant improvements in detecting drones and UAVs in low-light and complex environments, where older models sometimes struggled. Its advanced architecture, especially Flush Attention and improved feature fusion, helped reduce false detections and improved object tracking in real-time. Thanks to these enhancements, YOLOv12 became the most efficient model in our system, offering fast, accurate, and reliable drone detection suitable for continuous UAV monitoring.

In this part of the project, we used the same dataset described previously in section 2.17.2, maintaining the exact same data split into training, validation, and testing sets. No modifications were made to the dataset or its distribution, ensuring consistency across all experiments conducted with different YOLO versions.

Furthermore, the training procedure followed here is identical to the one detailed in section 2.17.3. This includes the same model configuration, number of training epochs, learning rate, and all other training parameters, as well as the tools and libraries used throughout the training process. By keeping the dataset and training method unchanged, we aimed to create a fair and reliable comparison of YOLOv11's performance relative to other versions under identical conditions.

Notably, YOLOv12 showed a significant improvement in training efficiency, with each epoch taking approximately 30 minutes, which is faster than YOLOv9. More importantly, it achieved the highest accuracy among all versions tested, especially in low-light conditions, highlighting its robustness and improved feature detection capabilities under challenging environments.

4.20.4 Results

A. Precision-Recall Curve.

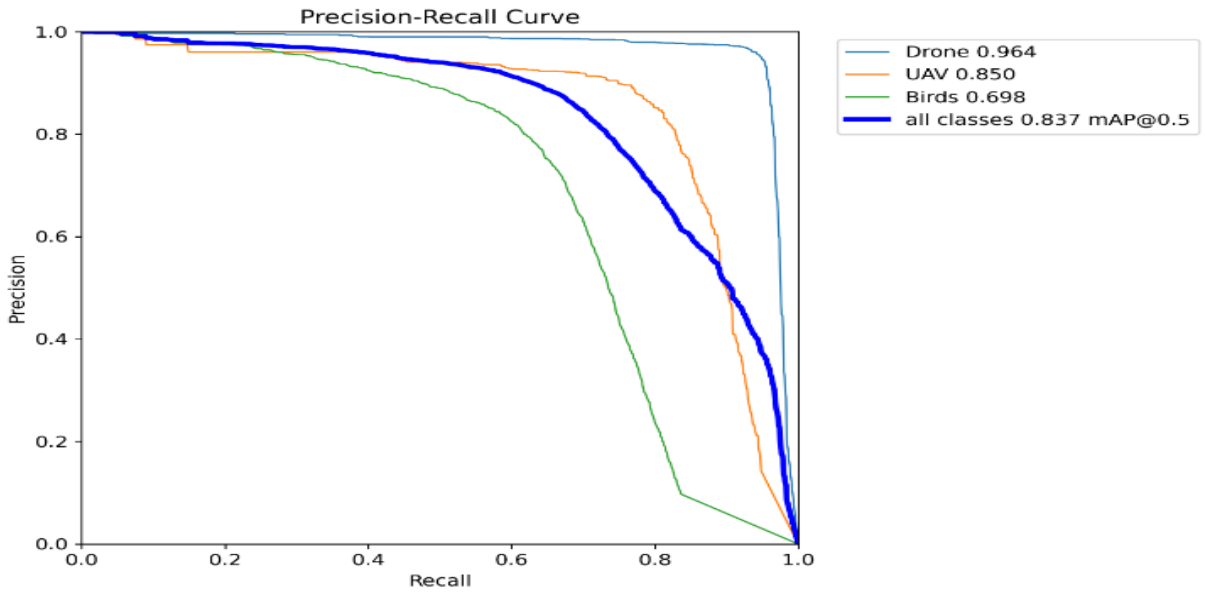


Figure 4. 31 Precision-Recall Curve

This Fig.4.31 represents a precision-recall curve for the model, showing the trade-off between the model's precision (its ability to correctly identify drones and UAVs) and its recall (its ability to find all drones and UAVs in the images). For **drones**, the model has a high precision of **0.964** with a corresponding mAP@0.5 value of **0.964.** This indicates that the model is highly accurate in detecting drones with good bounding box overlaps.

For **UAVs**, the model achieves a precision of **0.85** with a corresponding mAP@0.5 value of **0.85.** For **Birds**, the model achieves a precision of **0.698** with a corresponding mAP@0.5 value of **0.698.** The **mAP@0.5 for all classes** is **0.836,** representing the average precision for detecting both drones and UAVs when the IoU threshold is set to 0.5.

This means the model performs well overall, correctly identifying and locating both classes with bounding boxes overlapping the ground truth by at least 50%.

B. Recall-confidence Curve.

Fig.4.32 illustrates the trade-off between the model's recall (its ability to correctly detect objects) and confidence (its certainty in making detections). The model achieves a high recall of **0.91** across all classes at a confidence threshold of **0.00**, indicating that it can identify **91%** of objects in the dataset without applying strict confidence constraints.

Among the individual classes:
Drones have the highest recall, followed by UAVs, while Birds exhibit the lowest recall

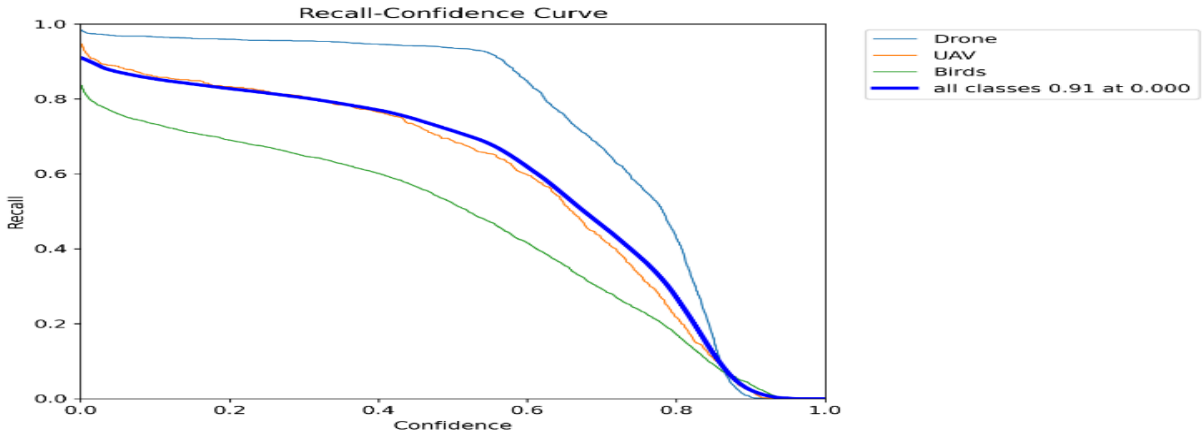


Figure 4. 32 Recall - Confidence Curve

performance. As confidence increases, recall decreases, showing the typical behavior where stricter thresholds filter out more detections.

This curve provides insight into the model's ability to balance recall and precision in different scenarios, making it useful for applications where detecting as many objects as possible is critical.

C. Precision-confidence Curve.

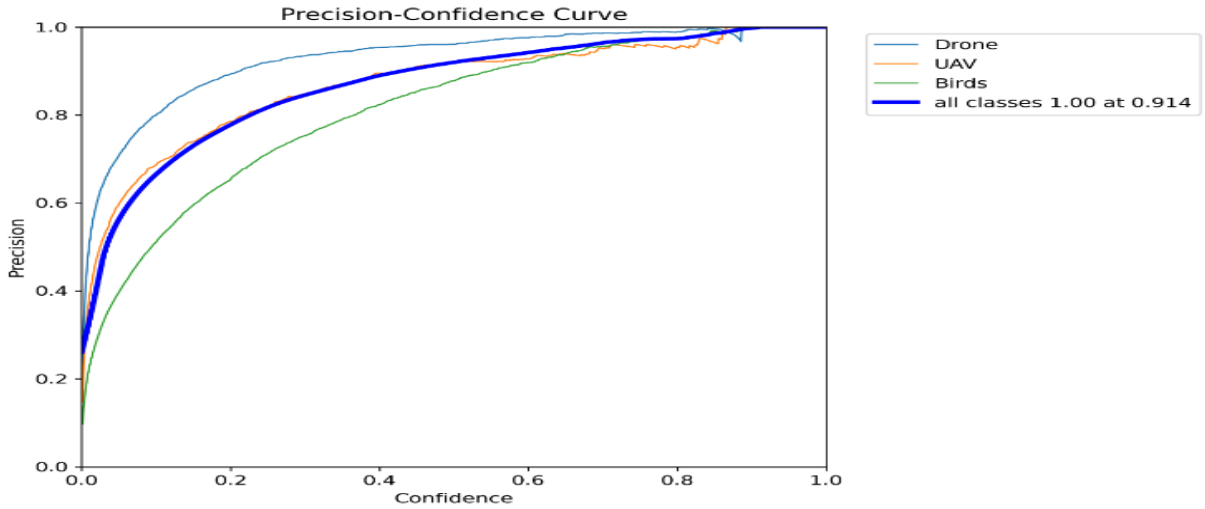


Figure 4. 33 Precision- Confidence Curve

This Fig.4.33 represents the relationship between precision (the proportion of true positive detections among all detections) and confidence (the model's certainty in its predictions).This curve illustrates the trade-off between precision and confidence, showing how stricter confidence thresholds impact the accuracy of detections.

The model demonstrates high precision, achieving a precision of **1.00** at a confidence threshold of **0.914**, indicating that when the model's confidence in its detections surpasses **0.914**, the predictions are almost always correct.

Among the different classes, Drones exhibit the highest precision, closely followed by UAVs, while Birds have the lowest precision at lower confidence levels. Although the mAP@0.5 value is not explicitly provided, the curve suggests that the average precision across all confidence levels is close to 1.0, highlighting the model's effectiveness in achieving accurate detections with a high degree of reliability.

D. F1-Confidence Curve.

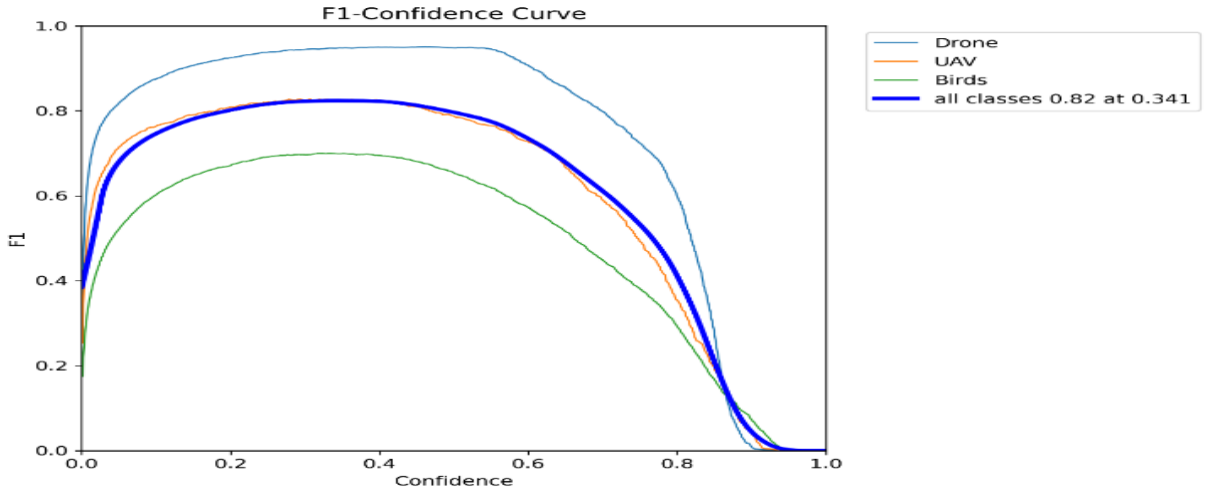


Figure 4. 34 F1- Confidence Curve

This Fig 4.34. shows F1-Confidence Curve, which illustrates the variation of the F1-score as a function of the confidence threshold. The F1-score, being the harmonic means of precision and recall, represents the trade-off between false positives and false negatives at different confidence levels.

Among the different classes:

The F1-score for Drones is the highest, peaking around **0.82**, while the F1-score for UAVs is slightly lower but closely follows the Drone curve. In contrast, the F1-score for Birds is significantly lower than that of both Drones and UAVs.

The overall best F1-score achieved across all classes is **0.82** at a confidence threshold of **0.341**, indicating that this threshold provides the optimal balance between precision and recall for the model’s performance.

E. Confusion matrix.

This Fig.4.35 represents a confusion matrix which is a table used to evaluate the performance of a classification model by comparing its predicted labels with the actual ground truth labels. When comparing each class against the background, the matrix helps assess how well the model differentiates objects from the background.

This binary classification approach provides insights into the model's ability to correctly detect each class while minimizing false detections.

(TPR): This value represents the True Positive Rate, also known as recall, which is 97%, 82% and 69% for drones, UAVs and Birds, respectively.
This indicates the percentage of actual drones, UAVs and Birds in the images that were correctly identified by the model.

(FPR): This value represents the False Positive Rate, which is 3% ,17% and 30% for drones, UAVs and Birds, respectively. This indicates the percentage of times the model incorrectly identified something as a drone or UAV when it wasn't.

(FNR): This value represents the False Negative Rate, which is 3% ,18% and 31% for drones, UAVs and Birds, respectively.
This indicates the percentage of times the model failed to identify drones, UAVs and Birds.

(TNR): This value represents the True Negative Rate, which is 0% for all classes. This indicates that the model never correctly identified the absence of drones, UAVs or Birds when they didn't exist.

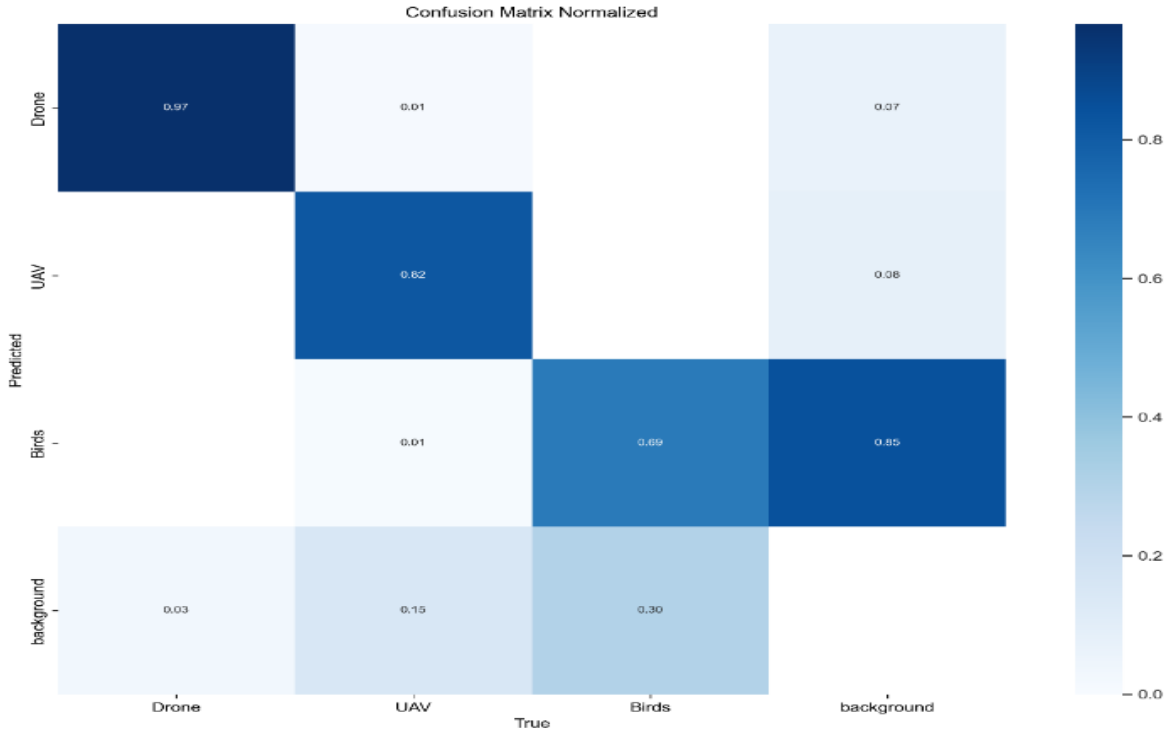


Figure 4. 35 Confusion Matrix

F. Performance Evaluation during Object Detection Model Training.

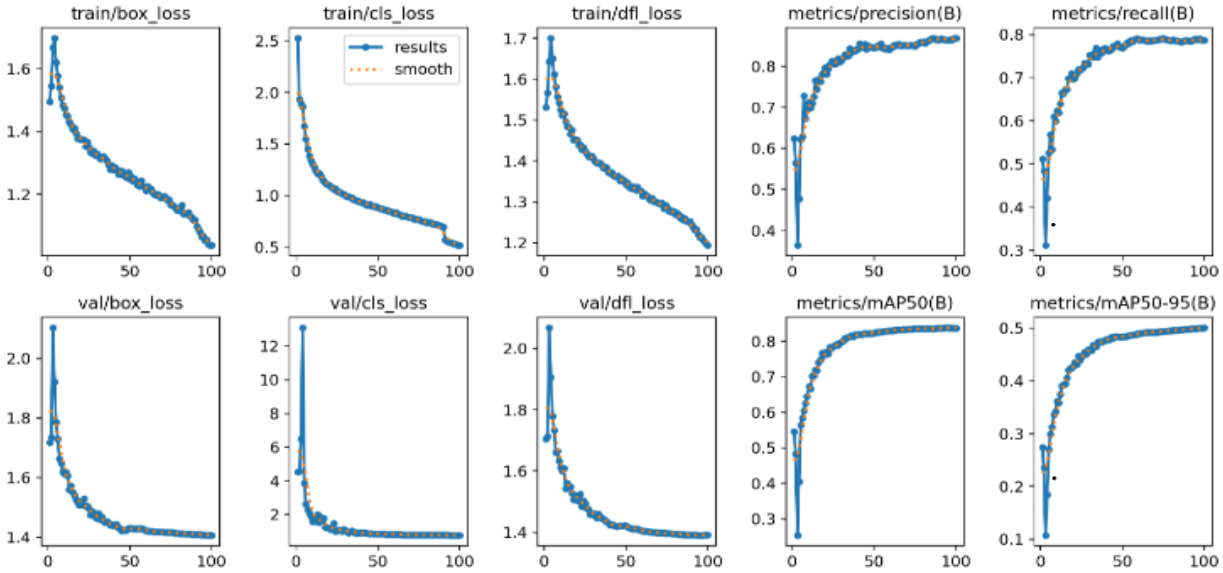


Figure 4. 36 Performance Evaluation

Model Performance Analysis

This section presents an analysis of the training and validation metrics for a YOLO-based object detection model. The evaluation focuses on loss reduction during training and key performance metrics related to precision, recall, and mean Average Precision (mAP).

Loss Metrics.

Loss functions play a critical role in model optimization, and their reduction over time indicates effective learning.

1. Bounding Box Loss (Train & Validation): The training box loss starts at approximately **1.8** and steadily decreases near to **0**, indicating an improvement in bounding box regression.

   Similarly, the validation box loss decreases from ~ **2.2** to < **1.4**, showing a consistent learning pattern between training and validation phases.
2. Classification Loss (Train & Validation): Initially, the training classification loss is **2.5**, but it rapidly drops below **0.5**, demonstrating that the model effectively differentiates between object classes.

   In contrast, the validation classification loss starts at **14** and decreases near to **0**, which, while showing improvement, remains higher than the training loss.
3. DFL (Distribution-Focused Learning) Loss: The training DFL loss decreases from **1.7** to less than **1.2**, and the validation DFL loss drops from **2.1** to less than **1.4**, reflecting enhanced accuracy.

Performance Metrics

To evaluate the effectiveness of the trained model, several key performance indicators are analyzed:

4. Precision: The model achieves a final precision of **0.95**, meaning **95%** of its detections are correct.
   This suggests a low rate of false positives.
5. Recall: The recall value reaches ~ **0.78**, indicating that the model detects approximately **78%** of all actual objects present in the dataset.
6. Mean Average Precision (mAP@0.5)**:** The model attains an mAP@0.5 score of ~ **0.85**, demonstrating strong detection performance when using a 50% Intersection over Union (IoU) threshold. It indicates that the model is well-optimized for object detection at a moderate overlapping threshold.
7. Mean Average Precision at Varying IoU Thresholds (mAP@0.5:0.95): The performance declines to **0.5** when evaluated across a range of IoU thresholds from 0.5 to 0.95. This suggests that while the model is highly effective under moderate conditions, its precision decreases for stricter IoU requirements.

4.20.5 Evaluate results: (Drones, UAVs and Birds)

Figure 4. 37 Results: (Drones, Uavs, Birds)

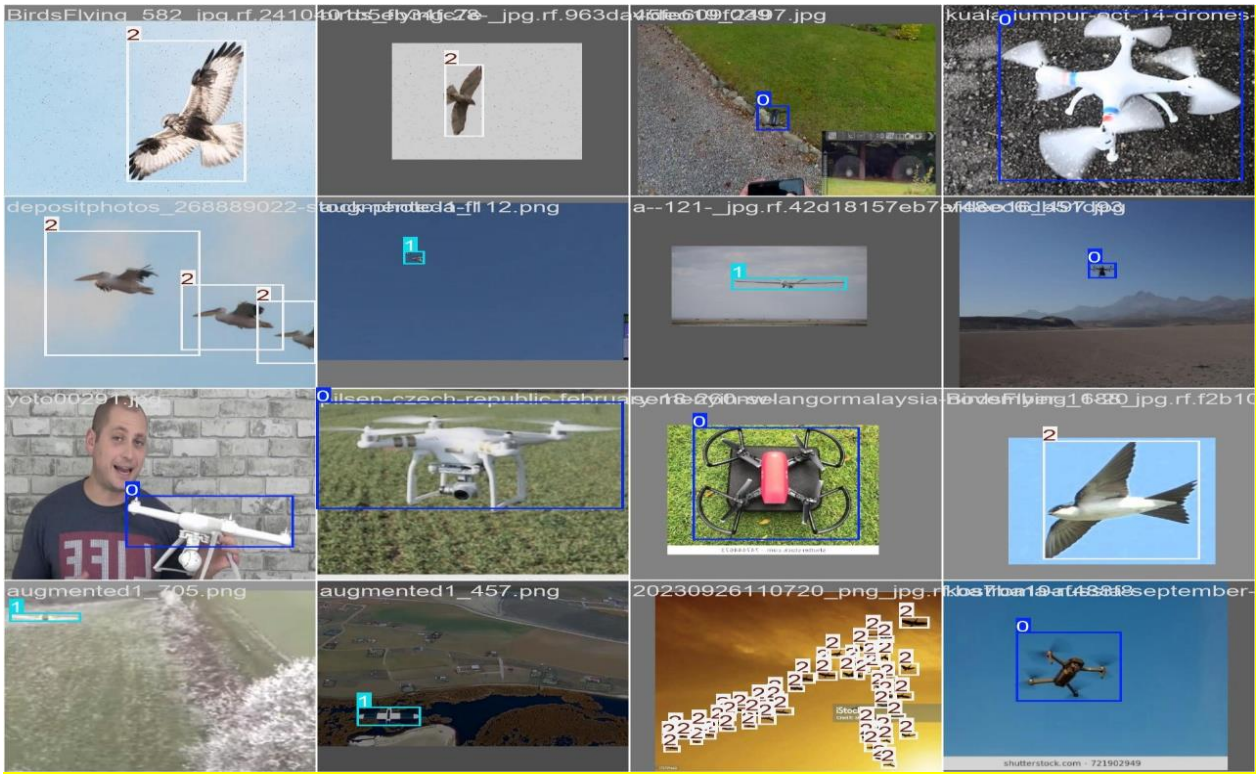


Figure 4. 38 Results: (Drones, Uavs, Birds)

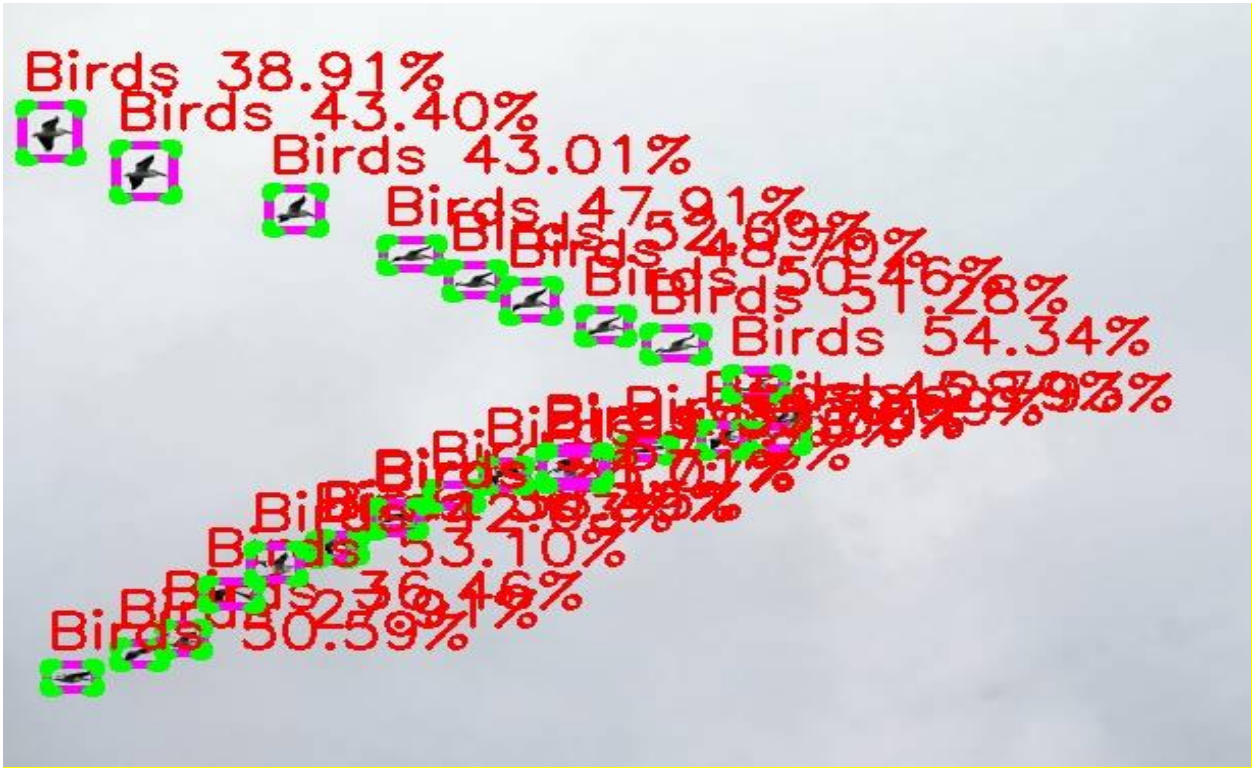


Figure 4. 39 Results: (Birds)

4.21 Models Performance

Version	Target	Precision	Recall	mAP50
YOLOv9	Drones UAVs Birds	0.96 0.81 0.69	0.91	0.836
YOLOv11	Drones UAVs Birds	0.96 0.80 0.71	0.90	0.816
YOLOv12	Drones UAVs Birds	0.97 0.82 0.69	0.91	0.837

The table above summarizes the performance of YOLOv9, YOLOv11, and YOLOv12 across three target classes: Drones, UAVs, and Birds. The metrics presented include Precision, Recall, and mAP50, which collectively offer a comprehensive view of each model's detection accuracy and reliability.

From the results, we observe that YOLOv12 achieved the highest overall performance, with a slight edge in both precision and recall for most targets. Specifically, it recorded the highest precision for drones (0.97) and the highest recall for UAVs (0.82), along with the highest mAP50 (0.837), outperforming both YOLOv9 and YOLOv11. These results demonstrate YOLOv12's enhanced ability to detect objects accurately, especially in complex scenarios.

This consistent improvement suggests that YOLOv12 offers better generalization and robustness compared to previous versions, making it the most effective model among those evaluate.

While all three models perform well in detecting drones, there are slight fluctuations in performance when it comes to UAVs and birds. YOLOv11 shows slightly better precision for birds, whereas YOLOv12 maintains a balance across all categories, particularly excelling in UAV detection with the highest recall.

The consistency of YOLOv12 across different object types and lighting conditions highlights its stability and adaptability, which are crucial in real-world drone detection systems.

Moreover, the improvement in accuracy did not come at the cost of training time. YOLOv12 not only performed better but also trained faster than its predecessors, making it more efficient and practical for iterative development and deployment.

4.22 Sound Detection

4.22.1 Introduction

Sound detection is the process of identifying the presence or nature of sound within a given environment using various sound analysis techniques. In recent years, sound detection has become a vital area of research with applications in security and surveillance, speech recognition, human-device interaction, and environmental monitoring. It involves the ability of computational systems to process and analyze audio signals, extracting useful information, whether the signals are from natural sounds (like environmental sounds or human voices) or artificial sounds (such as machine or vehicle noises).

The process of sound detection requires a deep understanding of the characteristics of the sound signal, including frequency, time, and energy. These characteristics can be utilized to extract detailed information about sound events. Given the variety of sounds in the environment, the technology used for detection must be capable of distinguishing between different types of sounds, ranging from unwanted noise like ambient noise to important sounds like alarms or alerts.

Sound detection techniques have evolved significantly, starting from traditional methods based on filtering and spectral analysis of the audio signal, using artificial intelligence (AI) and deep learning to analyze sound data more accurately and efficiently. As these technologies have advanced, they have become better at interacting with sound in a smart and precise manner, making them an essential component of modern applications such as smart voice assistants (like Google Assistant and Siri), speech recognition systems, security systems, and even human-robot interaction in industrial settings.

Many modern systems leverage spectral representations such as Mel Spectrograms, which transform the sound signal into a visual format that can be processed by deep neural networks (DNNs) such as CNNs and RNNs. These systems enable accurate sound pattern recognition and effective sound analysis.

These techniques are pivotal in the field of artificial intelligence and are integral to advancing machine learning and data analysis. They play a crucial role in enhancing the human-machine interaction and improving various applications in daily life and industry.

4.22.2 Applications for Sound Detection

Sound detection plays a vital role in many systems and technologies, as audio signals carry essential information about the surrounding environment, events, and human activities. The ability to detect and interpret sounds is crucial in various domains, enhancing safety, comfort, and operational efficiency. Below are some of the main applications of sound detection:

- **Safety and Emergency Response:** Sound detection systems help identify critical sounds such as alarms, explosions, or distress calls, enabling faster reaction to incidents and improving safety levels.

- **Surveillance and Security:** It is used in security systems to detect unusual sounds like glass breaking or unauthorized entry attempts, helping to identify potential threats.
- **Healthcare:** Patient conditions can be monitored by tracking sounds such as coughing or breathing irregularities.
- **Speech and Communication Support:** Individuals with hearing or speech difficulties benefit from tools based on sound detection, including hearing aids and pronunciation learning applications.
- **Environmental and Wildlife Monitoring:** Used to detect sounds of birds, animals, or natural phenomena like rain or landslides in remote areas.
- **Industrial Maintenance:** Some maintenance systems rely on detecting changes in machinery sounds to predict failures before they occur.
- **Military Applications:** Sound detection is used in defense systems to identify suspicious sources such as drones, gunfire, or human movement in critical zones, enhancing situational awareness and response in combat environments.
- **Assistance for the Visually Impaired:** Audio signals help detect motion or serve as alerts, supporting the interaction of visually impaired individuals with their surroundings.

4.23 Types of Sound Detection Techniques

Sound detection techniques can be categorized into two main types: traditional methods and AI-based methods. The choice of technique depends on the type of sound to be detected, the required system accuracy, and the complexity of the sound environment.

4.23.1 Traditional Detection Methods

These methods rely on direct properties of the audio signal, without the use of learning models. Some of the most common methods include:

- **Energy-Based Detection:**
  Sound presence is determined when a specific energy threshold is exceeded. This method is typically used in simple applications such as alarm systems.
- **Frequency Analysis:**
  Analyzing the frequency spectrum of the sound using tools like FFT (Fast Fourier Transform) helps in detecting specific characteristics in the audio.
- **Simple Acoustic Event Detection:**
  This method identifies specific events, such as knocking or clapping, based on predefined characteristics without using learning models.

Although these methods are fast and simple, they may struggle in noisy environments or when distinguishing between multiple sound types.

4.23.2 AI-Based Detection Methods

AI-based methods have the advantage of learning from data and adapting to complex or similar sounds. Some of the most popular techniques include:

- **Deep Learning Networks (CNN, RNN):**
  These networks analyze sound representations like Mel Spectrograms to identify sound patterns accurately.
- **Traditional Machine Learning Models:**
  Models like SVM (Support Vector Machine) or Random Forest are used with extracted sound features such as MFCC (Mel-Frequency Cepstral Coefficients) to classify sounds.
- **Audio Event Recognition Systems:**
  These systems are used in security and surveillance applications where they can distinguish specific sounds like breaking glass or alarm sirens.
- **Speech Recognition:**
  Commonly used in voice assistants and applications that rely on voice commands, such as Siri or Google Assistant.

4.24 Using Deep Neural Networks in Sound Detection

In the era of artificial intelligence, Deep Neural Networks (DNNs) have become a fundamental tool for effectively analyzing sound data. These networks offer advanced methods to understand complex signals such as audio, which involve variations over both time and frequency. Among the most used architectures in sound detection are Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), each offering distinct advantages depending on the nature of the task.

4.24.1 CNNs in Sound Detection

CNNs are used for pattern recognition in image data, but they are also highly effective in audio analysis, especially when the audio signal is visually represented using Mel Spectrograms. These spectrograms transform the audio into a two-dimensional image, enabling CNNs to detect time-frequency patterns and extract essential features needed to classify and recognize different types of sounds.

4.24.2 RNNs in Sound Detection

RNNs, on the other hand, are designed to work with sequential data. They are particularly useful when analyzing audio signals in their raw form or as feature sequences, as they consider the temporal context of the sound. This makes them suitable for applications like speech recognition or detecting audio events that unfold over time. However, RNNs may sometimes struggle with long sequences or subtle details without further enhancements.

2.25 CNN Architecture in Sound Detection

Convolutional Neural Networks (CNNs) are a special type of deep learning model mainly used for image processing, but they are also highly effective in audio analysis, especially when the sound is transformed into a visual form such as a Mel Spectrogram. CNN learns important patterns in the data by passing it through multiple layers.

The main layers that make up the CNN architecture are as follows:

1. **Input Layer:**
   This layer receives data, such as a 2D image representing sound with a Mel Spectrogram, which shows how sound frequencies change over time.

2. **Convolutional Layers:**
   These layers use filters (small windows) to scan the image and detect patterns such as edges or shapes. In sound spectrograms, they help identify time-frequency patterns that distinguish different sounds.

3. **Activation Function (ReLU):**
   After the filters operate, a function called ReLU is applied to keep only the positive values and discard the negative ones, helping the model understand more complex patterns.

4. **Pooling Layers:**
   These layers reduce the data size by keeping only the most important information. The most common type is Max Pooling, which retains the highest value in a small area.

5. **Fully Connected Layers (Dense):**
   These layers combine the important features found in previous layers to make the final decision. They help the model determine the type of sound.

6. **Output Layer:**
   This is the final layer that produces the prediction, such as which class the sound belongs to (e.g., drone sound, speech, or other sounds). A SoftMax function is often used here to provide probabilities for each class.

7. **Dropout (optional):**
   Dropout is a technique used during training that randomly disables parts of the model to prevent overfitting, making the model more generalizable and reliable when handling new data.

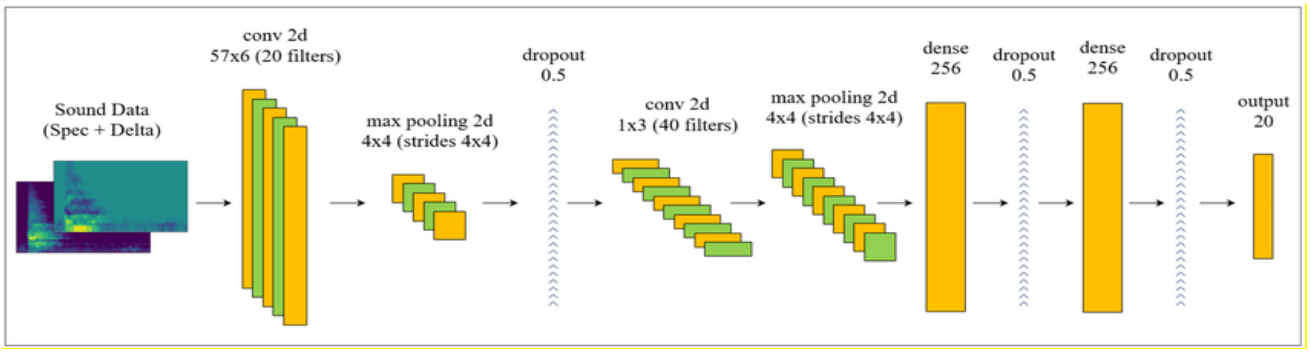


Figure 4. 40 CNN Architecture

4.26 Common Forms of Audio Data in AI

Several types of audio data are used in artificial intelligence applications, and the type of data selected depends on the nature of the task and the method of processing. Below are the most common types of audio data used in sound analysis and detection:

- **Raw Audio Signal:**

  This is the audio in its original form as recorded directly from the microphone, represented as a series of numerical values that capture sound vibrations over time. It is often used in models that rely on temporal analysis, such as RNN, but it requires careful processing due to the dense information it contains.

- **Spectrogram:**

  A spectrogram is a visual representation that shows how the frequencies of a sound change over time. It is created from the raw audio signal using spectral transformations (such as STFT). This type is often used as the starting point for advanced analysis.

- **Mel Spectrogram:**

  This is an improvement over the traditional spectrogram, where the frequencies are mapped onto the Mel scale, which mimics human hearing perception. This representation is most used in CNN models because it provides precise frequency information in an easy-to-analyze visual format.

- **MFCC (Mel-Frequency Cepstral Coefficients):**

  MFCC is a more condensed digital representation used to extract important audio features. It is widely used in speech recognition and spoken language processing. It provides a compact summary of the essential properties of the sound.

4.27 Libraries and Tools Used in Audio Processing and Model Building

- **Librosa for Feature Extraction:**

Librosa is an open-source Python library widely used for audio and music signal processing. It provides a rich set of tools to analyze audio data and extract meaningful features for use in machine learning models.

One of its key capabilities is converting raw audio signals into more informative representations such as the Spectrogram and Mel Spectrogram, which are commonly used as inputs for AI models. It can also extract features like energy, tempo, pitch, and MFCCs (Mel-Frequency Cepstral Coefficients), which help distinguish between different types of sounds.

Librosa allows easy loading of audio files, resampling, slicing the signal into time frames, applying spectral transformations, and analyzing the structure and characteristics of audio with high precision.

- **Matplotlib for Visualizing Spectrograms:**

Matplotlib is a widely used Python library for data visualization. In the context of audio processing, it is often used to display the Spectrogram, Mel Spectrogram, and other visual representations of sound. These visuals help illustrate how the frequency content of audio changes over time.

Visualizations created with Matplotlib play a key role in analyzing and verifying the quality of audio data before it is used in model training. They also help in identifying issues such as noise or clipping and are useful for comparing different audio samples.

Furthermore, spectrogram images generated using Matplotlib can be saved and used directly as input to image-based models like CNNs.

- **TensorFlow for Building and Training AI Models:**

TensorFlow is a powerful open-source platform developed by Google for building and training machine learning and deep learning models. It supports a wide range of neural network architectures, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), and provides full flexibility in customizing model structure, training strategies, and performance tracking.

With TensorFlow, developers can configure every aspect of the model from the number and type of layers, activation functions, and loss functions, to optimization algorithms like Adam or SGD.

It also offers built-in tools for evaluating model performance using accuracy, precision, recall, and confusion matrices, which are critical for understanding model effectiveness. TensorFlow supports both CPU and GPU acceleration, making it suitable for training complex models on large datasets efficiently.

- **Mel Spectrogram Representation:**

Mel spectrogram is one of the most used methods to represent sound in artificial intelligence applications, especially with convolutional neural networks (CNNs). It transforms raw audio from its time-domain waveform into a two-dimensional image that displays frequencies over time. The difference between a regular spectrogram and a Mel spectrogram lies in the use of the Mel scale, which mimics human perception of pitch. This makes it more accurate in capturing important acoustic features.

Mel spectrogram helps identify unique patterns within sound and is widely used in tasks such as:

- Sound classification
- Speech recognition
- Environmental sound analysis

- **MFCC (Mel-Frequency Cepstral Coefficients):**

  MFCC is a common method used to extract important features from audio signals. It breaks the audio into short frames and converts each frame into numbers that represent the main sound characteristics. MFCC uses the Mel scale, which mimics how humans hear different frequencies, making the features more meaningful. These features help machine learning models understand and classify sounds better, even with background noise.

4.28 Sound Detection Methodology in the Project

In this project, sound detection is used as an extra part of the main drone detection system, which mainly depends on cameras and artificial intelligence. Adding sound detection helps the system work better in places where visual detection is hard, like in low light or when things block the camera. Drones are a major new security threat, especially in military and civilian areas. By using an advanced microphone and techniques like Mel Spectrogram, MFCC, and deep learning, drone sounds can be detected and told apart from other background noises. Sounds from drones, like engine and propeller noise, are analyzed with deep learning models such as Convolutional Neural Networks (CNN) for early detection in different environments. This technology helps improve real-time security and monitoring, allowing better responses to security challenges and reducing risks from drones in sensitive areas.

4.28.1 Drone Sound Characteristics

Drone sounds are typically characterized by a unique combination of acoustic features resulting from their propeller rotation, motor vibrations, and aerodynamic design. These sounds often fall within a narrow frequency range, generally between 100 Hz to 6 kHz, and exhibit a constant buzzing or humming pattern. The acoustic signature may vary based on the number of rotors, motor type, and drone size. Unlike other environmental sounds, drone noise usually maintains a steady amplitude and frequency modulation, making it detectable even from a distance using specialized sound detection algorithms and spectrogram analysis.

4.28.2 Microphone Integration with Project

In our project to detect drones by sound, we use a high-quality microphone connected to the laptop through an external USB Sound Card with an Audio Cable (AUC). The AUC cable has two male ends made of copper: one end connects to the microphone, and the other connects to the USB sound card. To enhance sound capture quality, the microphone is positioned at a 45-degree angle. This setup helps to capture sounds clearly and directs the microphone toward the drones. The microphone is powered by a portable power source with a capacity of 20,000 mAh, which allows it to operate for long periods without needing to be plugged into a fixed power outlet. This makes the system flexible and able to work in different locations.

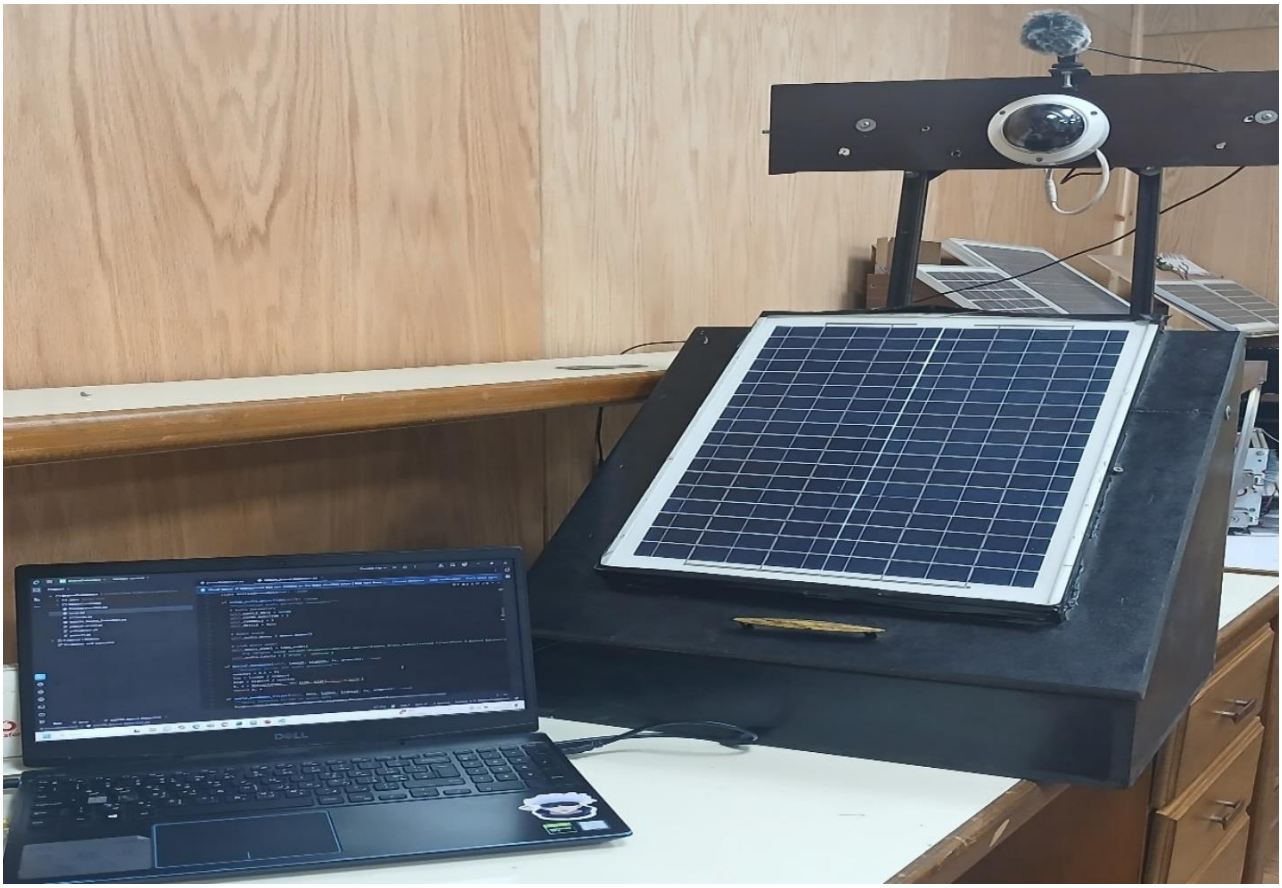


Figure 4. 41 Mic Integration

4.29 Stages of Model Training and Sound Data Preparation

4.29.1 Phase 1: Training in Initial Dataset

At the beginning of the drone sound detection system, a ready-made dataset was used that had drone sound recordings in the (yes drone) folder and other environment sounds in the (unknown) folder. There were 1332 drone sound files and 10372 unknown sound files. All files were cut into one-second segments. The model was trained using this data, but when tested, it could not clearly tell drone sounds apart from other sounds. This means the model did not learn the unique sound patterns well enough, likely because the data was unbalanced and not diverse enough.

4.29.2 Phase 2: Creating Enhanced Dataset with Added Sound Effects

To improve the model's performance in Phase 1, we created a more diverse dataset. Drone sound recordings were collected and mixed with background noises like wind, rain, and desert environments. Augmentation techniques such as time shifting and time stretching were applied to increase diversity. All clips were cut to one second. The final dataset included 10,186 drone clips

and 10,000 unknown clips. Despite these improvements, the model still struggled to classify the sounds accurately during testing.

4.29.3 Phase 3: Recording Data Using the Microphone

Because the results from the combined or changed data were not good, we started recording audio using the system's microphone. Since flying real drones was not possible, drone sounds were played through speakers and recorded by the microphone connected to the computer. The recorded audio clips were then cut into five-second parts instead of one second because one second was too short to capture enough sound details. This created a dataset with 96 drone sound clips and 74 clips of unknown sounds.

4.29.4 Model Training Methodology in the first two Phases

The model was trained to classify drone sounds using a standardized methodology, with data adjustments across different experiments to improve performance:

1. **Data Preparation:**
   In each experiment, audio files were converted into Mel Spectrograms using the librosa library. This feature helps transform the audio data into a visual representation that the model can understand. All audio files were segmented into one-second time windows to standardize the input size.

2. **Data Augmentation:**
   In Phase 2, Augmentation techniques like time-shifting and time-stretching were applied to the data to increase diversity and better represent different environments.

3. **Data Splitting:**
   The data was split into a training set (80%) and a test set (20%) using train_test_split from the scikit-learn library.

4. **Model Architecture:**
   A CNN model was used, consisting of Conv2D and MaxPooling2D layers to extract features from the Mel Spectrograms. A Flatten layer was added to transform the data into a format that could be fed into Dense layers for classification. Dropout was used to reduce overfitting, and Adam Optimizer was employed with binary cross entropy as the loss function.

5. **Model Training:**
   The model was trained on the data using the fit () function, with specified epochs 100 and batch-size 32. A validation_split was used to hold out a portion of the training data to test the model during training.

4.30.4 Model Training Methodology in Phase 3

In Phase 3, the drone sound classification model was enhanced by using MFCC (Mel Frequency Cepstral Coefficients) features instead of Mel Spectrograms, along with applying bandpass filtering during testing to improve sound quality.

1. **Data Preparation**

   Audio files were processed into MFCC features using the **librosa** library. Each audio clip was converted into 40 MFCC coefficients with a fixed length of 215 frames to standardize the input.

2. **Data Splitting**
   The dataset consisted of two classes: drone' and 'unknown'. The data was split into 80% training and 20% validation sets.

3. **Model Architecture**
   A CNN was built using Conv2D and MaxPooling2D layers to extract important features from the MFCCs, Dropout layers to reduce overfitting, and Dense layers with softmax activation for classification. The model used the Adam optimizer and categorical cross entropy loss.

4. **Model Training**
   The model was trained using the training data for 50 epoch with a batch size of 32. Validation data was used during training to monitor the model's performance and prevent overfitting.

5. **Performance Evaluation**
   The model was evaluated on the validation set using accuracy, loss, confusion matrix, precision-recall curves, and classification reports to measure how well it distinguished drone sounds from other sounds.

6. **Testing with Bandpass Filtering**
   During testing, a bandpass filter (100 Hz to 8000 Hz) was applied to the audio to reduce noise and focus on drone-specific sound frequencies. MFCC features were extracted from the filtered audio before making predictions with the trained model. The output included the predicted class (drone or unknown) and a confidence score.

4.31 Results

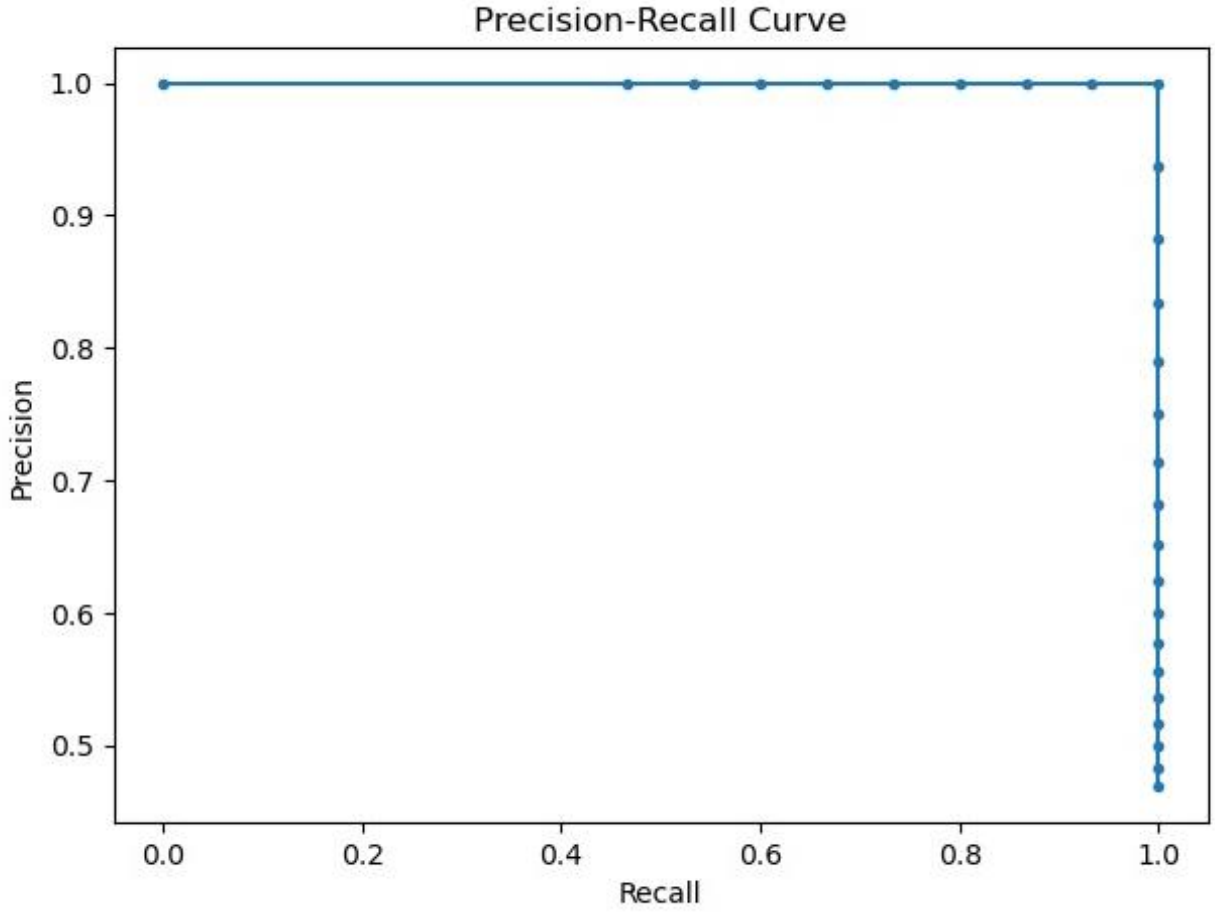


Figure 4. 42 Precision-Recall Curve

A. Precision-Recall This Fig4.42. shows the relationship between precision and recall, which are important for measuring how well the detection model works. The model performs very well, keeping a high precision (about 1.00) across most recall values. This means it gives very few false alarms. A sharp drop in precision appears only at the end, when recall reaches 1.0, and precision falls to around 0.47. This happens because the model tries to detect every possible object, even the unclear ones. Overall, the results show that the model is confident in its predictions and performs strongly in detecting targets. The average precision is about 0.97, which is a high value and shows the model is effective for detection and tracking.

B. Precision-Confidence Curve

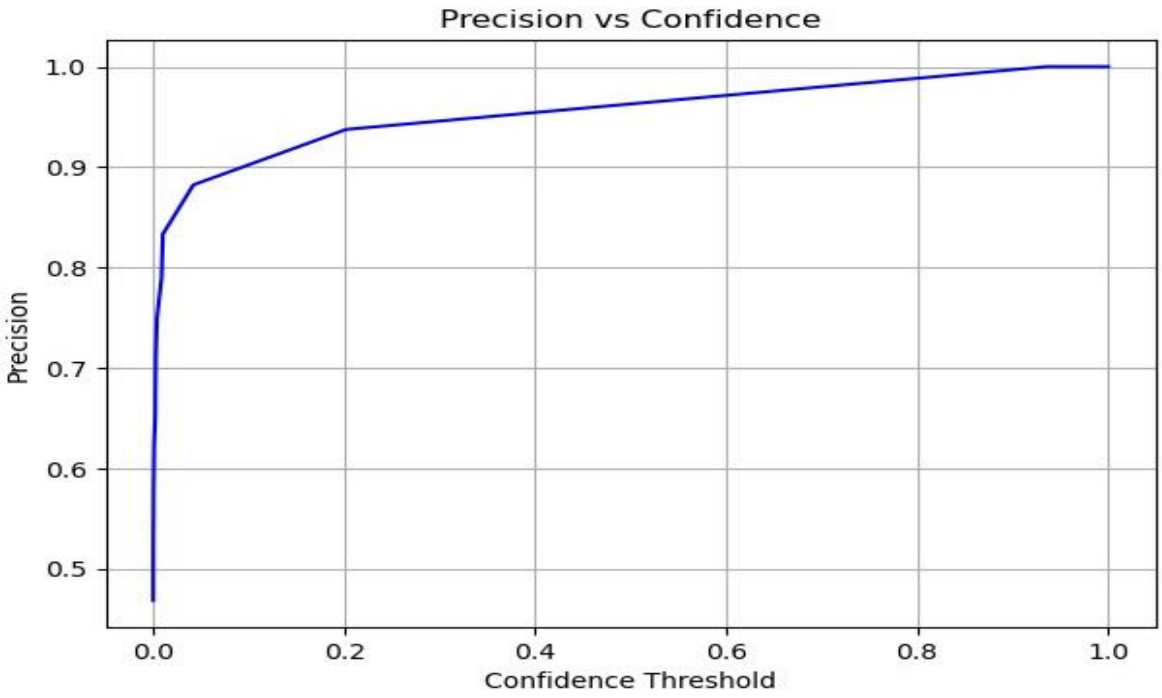


Figure 4. 43 Precision-Confidence Curve

Fig4.43 represents the Precision vs Confidence Curve visualizes how precision rises with confidence, demonstrating the model's robustness.

This curve offers a clear view of how the model balances precision with confidence in its predictions. It shows how the model's accuracy improves as the confidence threshold increases, ensuring that predictions are more reliable when the model is more certain.

At a confidence threshold between 0.9 and 1.0, the model reaches perfect precision (1.00), indicating that when the model's confidence in its prediction is very high, the detections are consistently accurate and error-free.

C. Recall-Confidence Curve

Figure 4. 44 Recall-Confidence Curve

Fig.4.44 represents the **Recall-Confidence Curve** and illustrates the trade-off between the model's recall (its ability to correctly detect drones) and confidence (its certainty in making detections).

The model achieves **a perfect recall of 1.00** at lower confidence thresholds (from **0.0** to approximately **0.95**), meaning it successfully detects all drones in the dataset without applying strict confidence constraints.

As the confidence threshold approaches **1.0**, recall sharply drops to approximately **0.9**, and then drops further to nearly **0.0**, indicating that at very high confidence levels, the model may miss many true positive detections.

This behavior reflects the typical trend where increasing the confidence threshold improves the precision but may reduce recall, as stricter confidence filtering can exclude correct detections.

D. Metrics-Confidence Curve

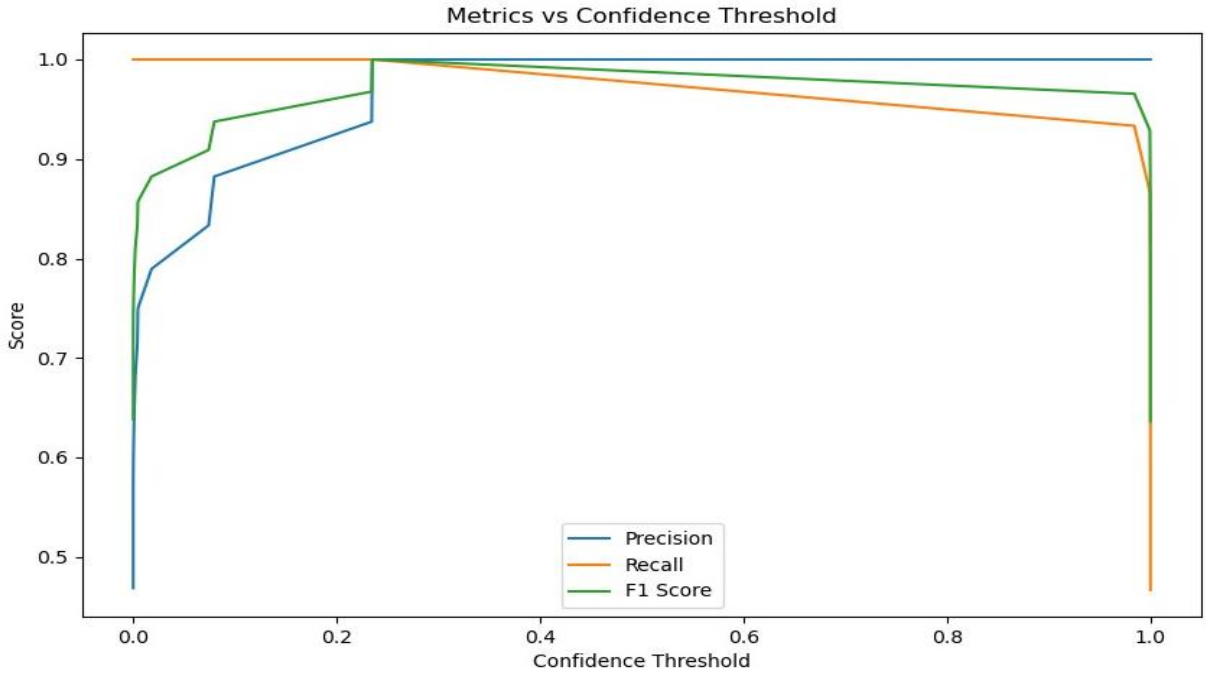


Figure 4. 45 Metrics-Confidence Curve

Fig.4.45 shows how the key performance metrics of an object detection model (Precision, Recall, and F1 Score) change as the confidence threshold is varied.

Curve Analysis:

1. **Precision (Blue Line)**

- At very low confidence (near 0), precision is low (~0.2). The model makes many false detections.
- As the confidence threshold increases, precision quickly improves and reaches 1.0 at around 0.25.
- Precision stays at 1.0 for a large range of thresholds, meaning that detections above ~0.25 are almost always correct.

2. **Recall (Orange Line)**

- At a confidence threshold of 0, recall is 1.0, meaning the model detects all objects (including false ones).
- As the confidence threshold increases, recall slowly drops after ~0.25 because the model misses some true objects.
- Recall drops very fast as confidence approaches 1.0, as the model becomes too strict.

3. **F1 Score (Green Line)**
   - F1 combines precision and recall. It starts high (~0.65) at a threshold of 0.
   - It reaches its highest value (close to 1.0) at around 0.25, where both precision and recall are good.
   - After that, F1 drops because recall becomes low.

E. Confusion Matrix

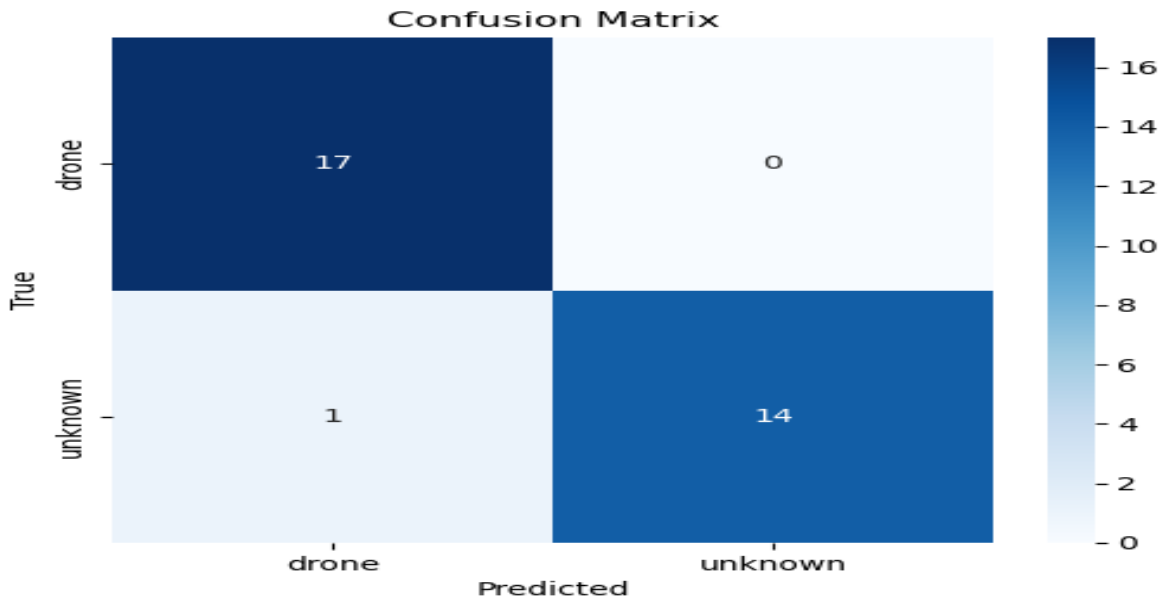


Figure 4. 46 Confusion Matrix

Fig.4.46 shows a Confusion matrix table that is used to evaluate the performance of a classification model by comparing its predicted labels with the actual ground truth labels. When comparing each class against the background, the matrix helps assess how well the model differentiates objects from the background. This binary classification approach provides insights into the model's ability to correctly detect each class while minimizing false detections.

(TPR): This value represents the True Positive Rate, also known as recall, which is 100% for drones. This indicates the percentage of actual drones in the data that were correctly identified by the model.

(FPR): This value represents the False Positive Rate, which is 6.67% for drones. This indicates the percentage of times the model incorrectly identified an unknown object as a drone.

(FNR): This value represents the False Negative Rate, which is 0% for drones. This indicates the percentage of times the model failed to identify actual drones.

(TNR): This value represents the True Negative Rate, which is 93.33% for unknown objects. This indicates the percentage of times the model correctly identified the absence of drones when they didn't exist.

F. Loss And Accuracy Over Epochs

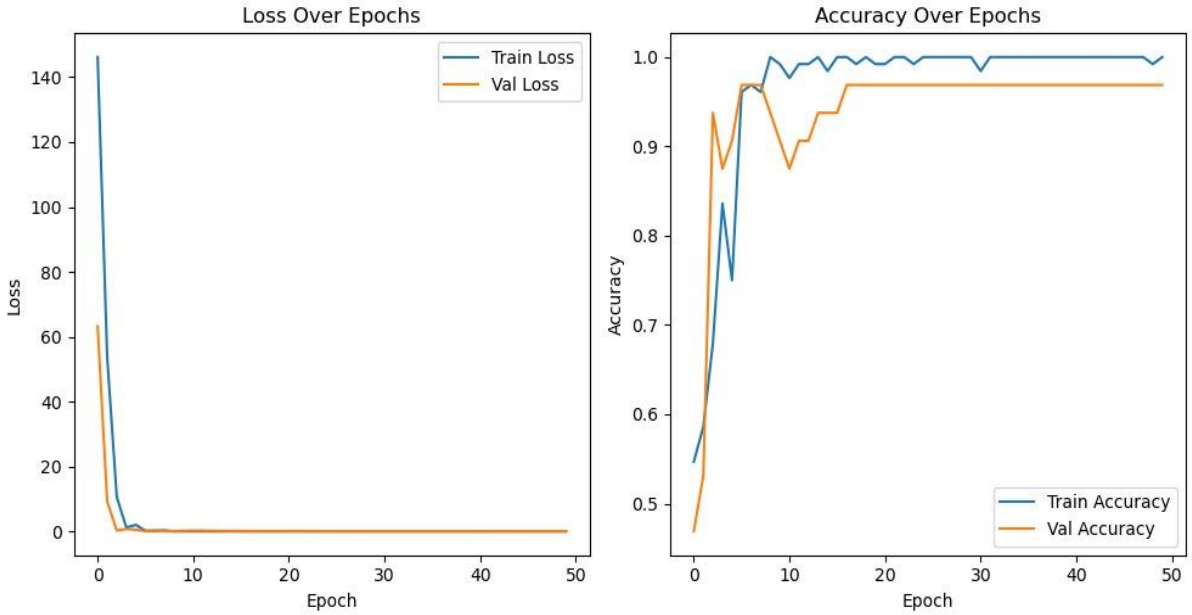


Figure 4. 47 Loss and Accuracy Curve

Model Performance Analysis Over Epochs:

The Fig.4.47 above shows how the model's loss and accuracy changed during 50 training epochs for both the training and validation sets.

At the start of training, the model had a very high loss: the training loss was about 145.9. However, during the first 5 to 10 epochs, the loss dropped quickly. This means the model was learning fast and improving its predictions. By the end of the training, the training loss was about 0.001 and the validation loss was about 0.002. This shows that the model's learning improved over time and became stable.

Similarly, the accuracy started low. At the beginning, the training accuracy was about 0.55 and the validation accuracy was about 0.25. But as the training continued, the accuracy got much better. By the end, the training accuracy reached about 0.998 and the validation accuracy was about 0.975. The validation accuracy stayed stable after the first 10 epochs. This shows the model did not overfit and kept performing well on new data it did not see during training.

4.32 Future Work

In the future, several enhancements can be made to improve the performance and scalability of the current vision and sound-based drone and UAV detection and tracking system. While our current work includes the implementation of YOLOv9, YOLOv11, and the most recent YOLOv12 for visual detection and tracking, there is still room for further development.

First, the sound detection system, which currently identifies drones only, can be extended to recognize UAVs as well, using a more diverse and comprehensive sound dataset. Improving the microphone directionality and sensitivity, for instance, by using a directional mic capable of capturing long-range signals—could significantly increase the detection accuracy and range in noisy environments. Similarly, upgrading to a higher-resolution camera would enhance object tracking, especially for small or distant UAVs.

From a deployment perspective, making the system standalone by integrating it with embedded AI hardware such as Jetson Nano is a promising direction. Alternatively, training the models on high-performance computing platforms can accelerate the training process and allow for more complex model architecture and larger datasets.

Lastly, future work could include the implementation of real-time alerting systems or automatic response mechanisms, enabling the system not only to detect and track but also to react or notify users instantly upon identifying a potential threat.

Chapter 5

Blocking Techniques

Presented by/

B. AMR ASHRAF IBRAHEM ANANY 95593

C. HAMZA FARAHAT MOHAMED HAMZA 91713

D. MARIEZ SAMY EISSA EISSA 97903

5.1 Communications jamming

The purpose of communication is to move information from one location to another. All the following types of transmitted signals are considered communication:

- Voice communication.
- Computer-to-computer communication.
- Command links.
- Datalinks.
- Weapon-firing links.
- Cell phones.

The purpose of communication jamming is to prevent the transfer of information. Communication jamming requirements depend on the signal modulation, the geometry of the link, and the transmitted signal power. Figure 3.1 shows the communication jamming geometry. Whereas a typical radar has both the transmitter and the associated receiver at the same location, a communication link; because its job is to take information from one location to another, always has its receiver in a different location from that of the transmitter. Note that you can only jam the receiver. Of course, communication is often done using transceivers (each including both transmitter and receiver), but only the receiver at location B in the figure is jammed.

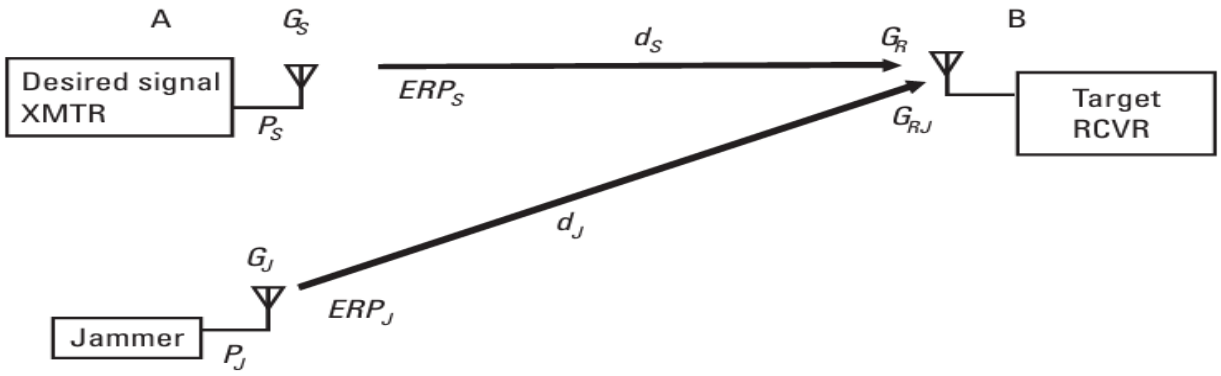


Figure 5. 1communication jamming geometry

If transceivers are in use and you want to jam the link in the other direction, the jamming power must reach location A. There are some important communications cases in which transceivers are not used for example in UAV links as shown in Figure 3.2.This figure shows the data link (or "downlink") being jammed. Again, you jam the receiver.

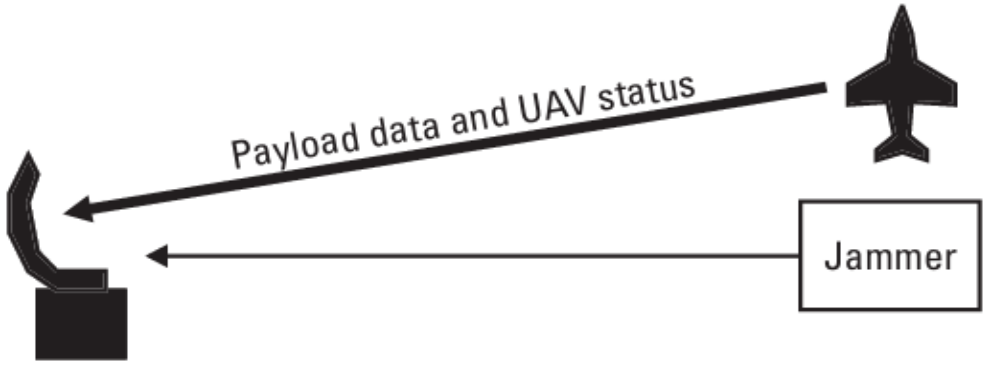


Figure 5. 2 UAV link jamming geometry

5.2 Stand-In Jamming

Stand-in jamming is the placement of a jammer close to the target receiver as shown in Figure 5.3. The effect is to reduce the jammer-to-receiver distance, increasing the J/S by the square (or fourth power) of the reduced distance).

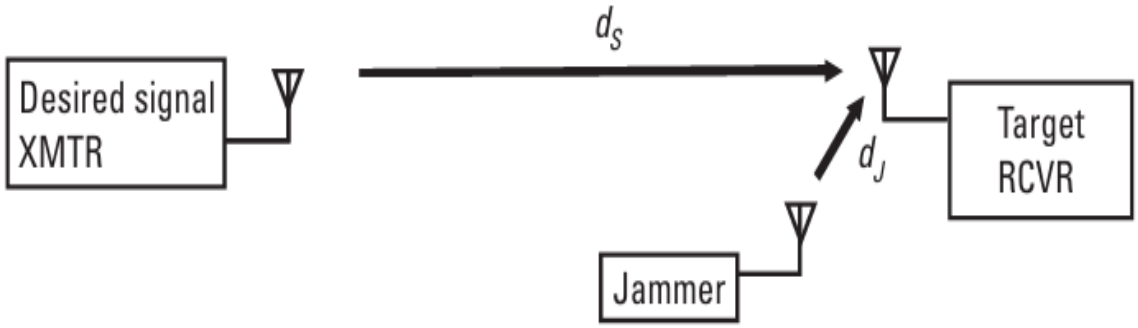


Figure 5. 3 Stand in jamming

This technique has the advantage of allowing decreased jamming power to achieve the same J/S. A derivative advantage is that friendly receivers—which are presumably much farther from the jammer than the target enemy receiver—will not be jammed. This prevents fratricide, or the unintentional jamming of friendly communications. Stand-in jamming techniques include emplaced jammers, jamming payloads on UAVs, and artillery-delivered jammers. Stand-in jamming can be particularly advantageous against spread spectrum communication where jamming from long distances is difficult because of the requirement to overcome processing gain in the receiver.

5.3 Spread spectrum communications

A. Introduction

All the modulation techniques discussed in Chapter II have been designed to communicate digital information from one place to another as effectively as possible in an AWGN environment. Although many real-world communication channels are accurately modeled as AWGN channels, there are other important channels that do not fit this model. For example, a military communication system that might be jammed by a continuous wave tone near the modem's (an acronym for modulator-demodulator) center frequency or by a distorted retransmission of the modem's own signal. We cannot model either of this interference as AWGN. But we can use spread spectrum techniques to eliminate the effects of these types of interference. Spread spectrum techniques can be very useful in solving a wide range of communications problems. The amount of performance improvement that is achieved using spread spectrum is defined as the "processing gain" of a spread spectrum system. Processing gain is the difference between system performance using spread spectrum techniques and system performance not using spread spectrum techniques. An often-used approximation for processing gain is the ratio of spread bandwidth (Wss) to the information rate (R).

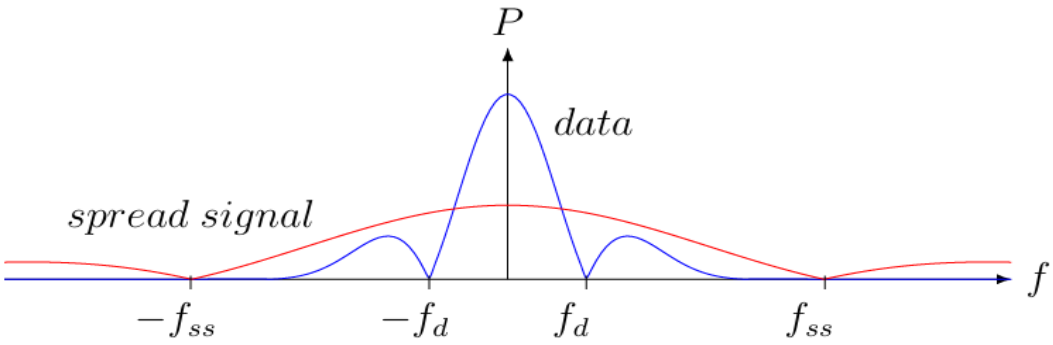


Figure 5. 4 power spectrum of data and of spread signal

B. SPREAD SPECTRUM SYSTEM

A system is defined to be a spread spectrum system if it fulfills the following requirements:
1. The transmitted signal occupies a bandwidth much more than the minimum bandwidth necessary to send the information.

2. Spreading is accomplished by means of spreading signal, often called a code signal, which is independent of the data.
3. At the receiver, dispreading (recovering the original data) is accomplished by the correlation of the received spread signal with a synchronized replica of the spreading signal used to spread the information.

Spread spectrum signals are used for:
1. Combating or suppressing the detrimental effect of interference due to jamming, interference arising from other users of the channel, and self-interference due to multipath propagation.
2. Hiding a signal by transmitting it a low power and, thus, making it difficult for an unintended listener to detect in the presence or background noise.
3. Achieving message privacy in the presence of other listeners.

C. SPREAD SPECTRUM TECHNIQUES
There are five types of spread spectrum techniques, which are:

- Direct sequence
- Frequency hopping
- Time hopping
- Hybrid
- Chirp

The two types of spread spectrum techniques most employed are direct sequence (DS) spread spectrum and frequency hopping (FH) spread spectrum, described below.

1. Direct Sequence Spread Spectrum:

Direct sequence systems are the best known and most widely used spread spectrum systems. This is because of their relative simplicity from the standpoint that they do not require a high-speed frequency synthesizer (A frequency synthesizer is a device which converts a stable frequency into the various hopping frequencies.)

In a direct sequence spread spectrum system, the signal is multiplied by a high-rate pseudo-random binary sequence, with the result that the transmitted signal's power spectrum is spread over a significantly larger bandwidth. This lowers the magnitude of the transmitted signal's spectrum and makes it appear more noise-like to an observer. By making the transmitted signal's spectrum appear more noise-like, it is more difficult for a hostile observer to detect (successful determination of signal presence or absence) and intercept (to convert the transmitted signal back to its original form) the signal as compared to a conventional signal. Therefore, a direct sequence spread spectrum system may be both a low probability of detection (LPD) and a low probability of intercept (LPI) system as compared to conventional communication systems. Figure 5.5 is a basic block diagram of a direct sequence spread spectrum system.

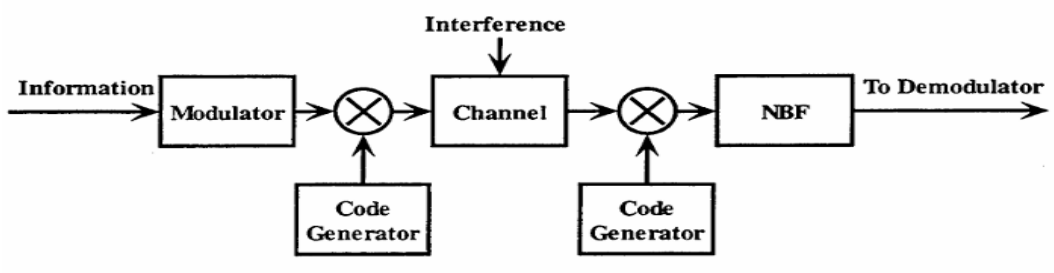


Figure 5. 5 Basic block diagram of a direct sequence spread spectrum system

This technique can be defined in terms of a code sequence c(t) in the form of positive and negative pulses of unit amplitude (representing a binary code sequence of ones and zeros.) These symbols occur at a rate Rc, the chip rate, where a chip is one symbol and Tc is the chip time. Usually, Rc is much larger than R_I, the rate at which information symbols occur.

The spreading is accomplished by multiplying the information-modulated carrier by the code. The dispreading is accomplished by multiplying the received waveform by the same code synchronized to the incoming code.

As can be seen from Figure 3.4, there is an interference term added to the channel. When the received signal is multiplied by the code sequence, the interference bandwidth increases greatly above that of the information-modulated carrier. The narrow band filter (NBF) rejects almost all the power of the undesired signal. The reduction is approximately by the factor of R_I/Rc , which in effect causes a corresponding processing gain (PG) of Rc/R_I.

1. Frequency Hopping Spread Spectrum:

In a frequency-hopped (FH) spread spectrum system, the carrier frequency of the signal is changed (hopped) in a pseudo-random fashion over a large band of frequencies. Although a potential adversary may be able to detect the signal, an FH signal will be more difficult to intercept than conventional signals, because how the carrier frequency is varied is not known by the adversary. Therefore, a FH spread spectrum system may be considered to be a LPI system as compared to conventional communication systems. If in an FH spread spectrum system more than one symbol is transmitted per frequency hop, then it is called slow FH spread spectrum system. If one symbol is transmitted by one or more consecutive hops, then it is called fast FH spread spectrum system. Figure 5.6 is a basic block diagram of a frequency hopping spread spectrum system.

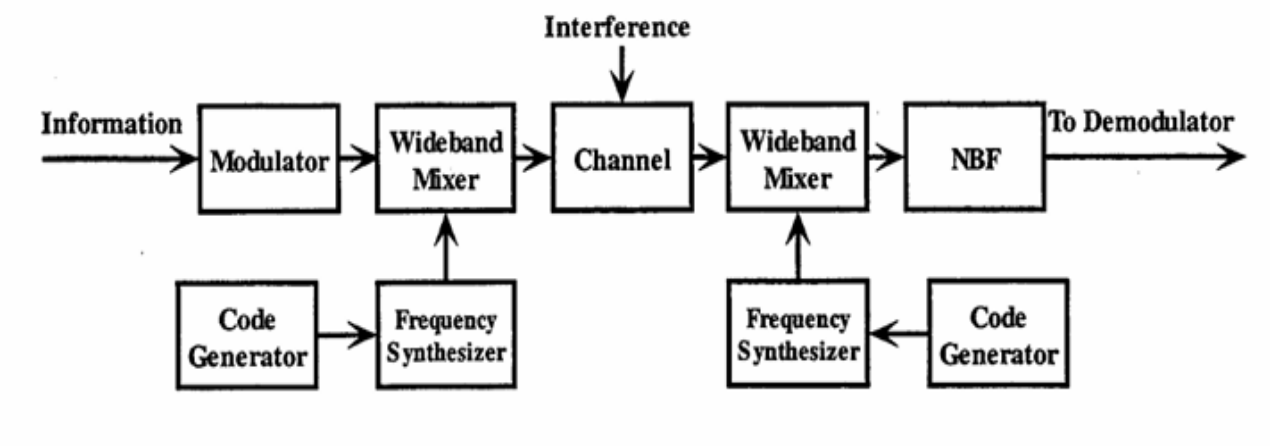


Figure 5. 6 Basic block diagram of a frequency hopping spread spectrum system

A frequency hopping system consists basically of a wideband mixer, a code generator and frequency synthesizer capable of responding to the coded output from the code generator. A great deal of effort has been expended in developing rapid-response frequency synthesizers for spread spectrum systems. The objective of a frequency hopping system is to produce a transmitted signal (which has a frequency spectrum of bandwidth W_I associated with it) such that the center frequency f_i is changed in discrete steps in an apparently random manner. That is, the frequency seems to hop at a hopping rate of R, over a total allocated bandwidth Wss. The ratio Wss/W_I is called processing gain.

In conclusion, the spread spectrum is essential for secure and reliable wireless communication. It provides interference resistance, anti-jamming security, and efficient multiple access (CDMA)**,** making it crucial for GNSS, military, and modern networks like Wi-Fi and 5G. By spreading signals over a wider bandwidth, it enhances signal robustness, accuracy, and efficiency**,** ensuring secure and interference-free transmission in various applications as summarized in table:

Evaluation of Spread Spectrum (SS)	Positive	Negative
1	Signal hiding (lower power density, noise-like), non-interference with conventional systems and other SS systems	No improvement in performance in the presence of Gaussian noise
2	Secure communication (privacy)	Increased bandwidth (frequency usage, wideband receiver)
3	Code Division Multiple Access CDMA (multi-user)	Increased complexity and computational load
4	Mitigation (rejection) of multipath, hold only the direct path	
5	Protection to intentional interference (Jamming)	
6	Rejection of unintentional interference (narrowband)	
7	Low probability of detection and interception (LPI)	
8	Availability of license-free ISM (Industrial, Scientific and Medical) frequency-bands	

Figure 5. 7 Equivalation of spread spectrum

5.4 GNSS Interference

Interference refers to the phenomenon where unwanted signals or disturbances disrupt the reception or transmission of a desired signal in a communication system. It occurs when external signals, noise, or other electromagnetic disturbances overlap with the intended signal, degrading its quality or making it difficult to decode. Interference can arise from various sources.

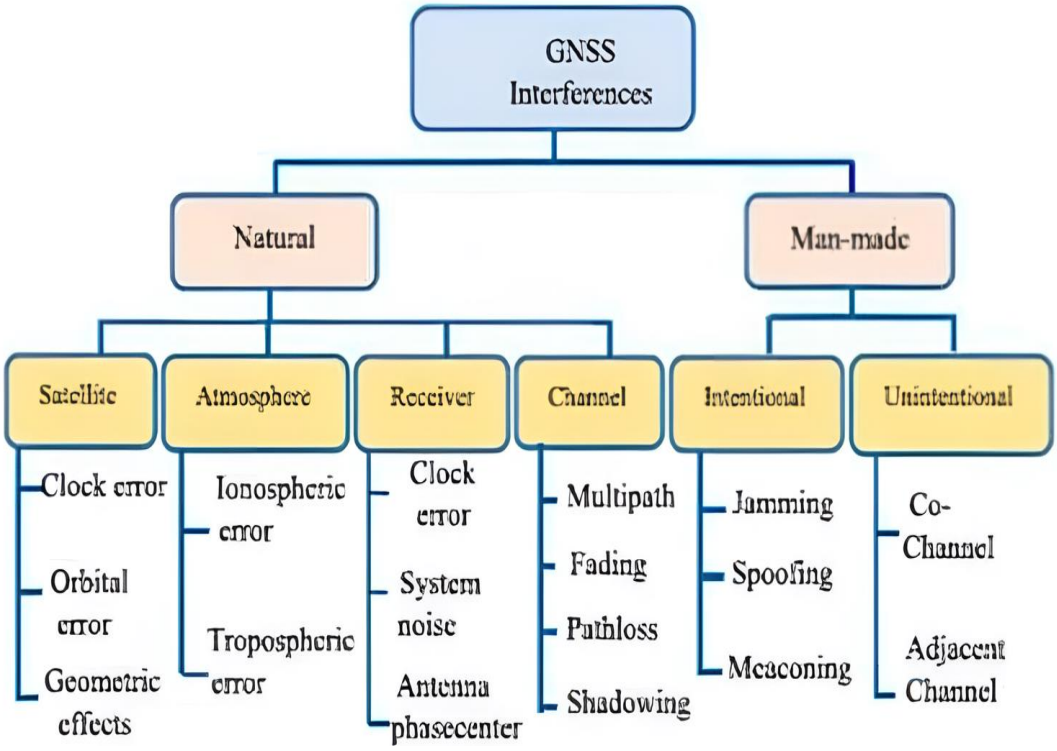


Figure 5. 8 GNSS interferences

Performance of SS in presence of interference:

1) Narrow band interference.

The narrowband noise is spread by the multiplication with the PN sequence pn, of the receiver.

The power density of the noise is reduced with respect to the despread data signal. Only 1/Gp of the original noise power is left in the information baseband (Rs). Spreading and dispreading enables a bandwidth trade for processing gain against narrow band interfering signals. Narrowband interference would disable conventional narrowband receivers.

The essence behind the interference rejection capability of a spread spectrum system: the useful signal (data) gets multiplied twice by the PN sequence, but the interference signal gets multiplied only once.

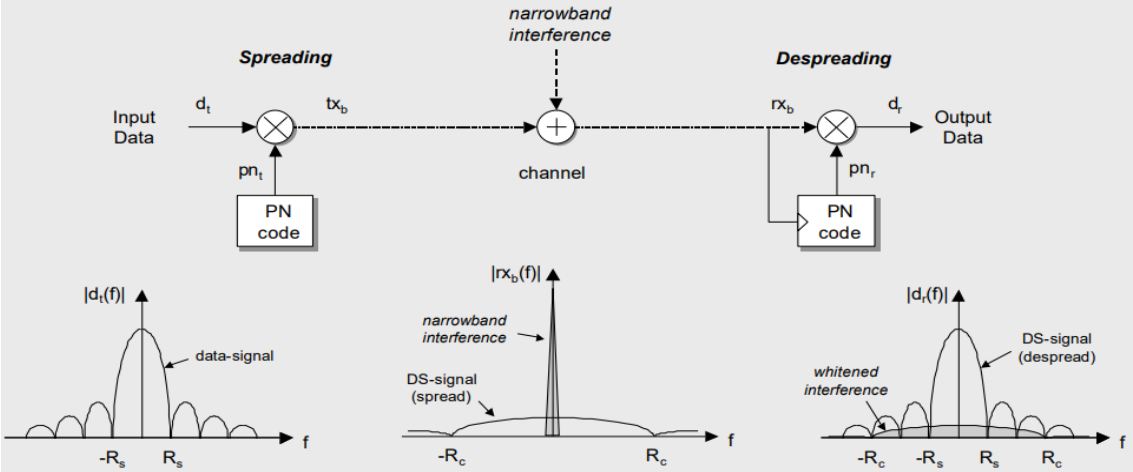


Figure 5. 9 Narrowband interference

2) Wideband interference.

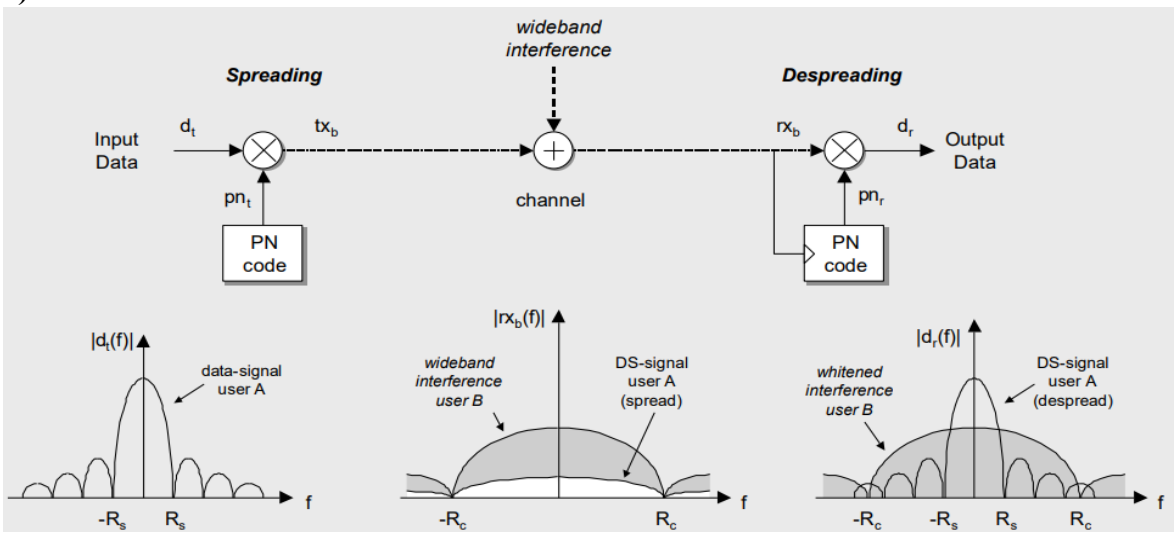


Figure 5. 10 Wideband interference

Multiplication of the received signal with the PN sequence of the receiver gives a selective de-spread of the data signal (smaller bandwidth, higher power density). The interference signal is uncorrelated with the PN sequence and is spread.

Origin of wideband noise:

- Multiple Spread Spectrum users: multiple access mechanism.
- Gaussian Noise: There is no increase in SNR with spread spectrum. The larger channel bandwidth (Rc instead of Rs) increases the received noise power with Gp:

Ninfo = N0.BWinfo → NSS = N0.BWss = Ninfo.Gp

The spread spectrum signal has a lower power density than the directly transmitted signal.

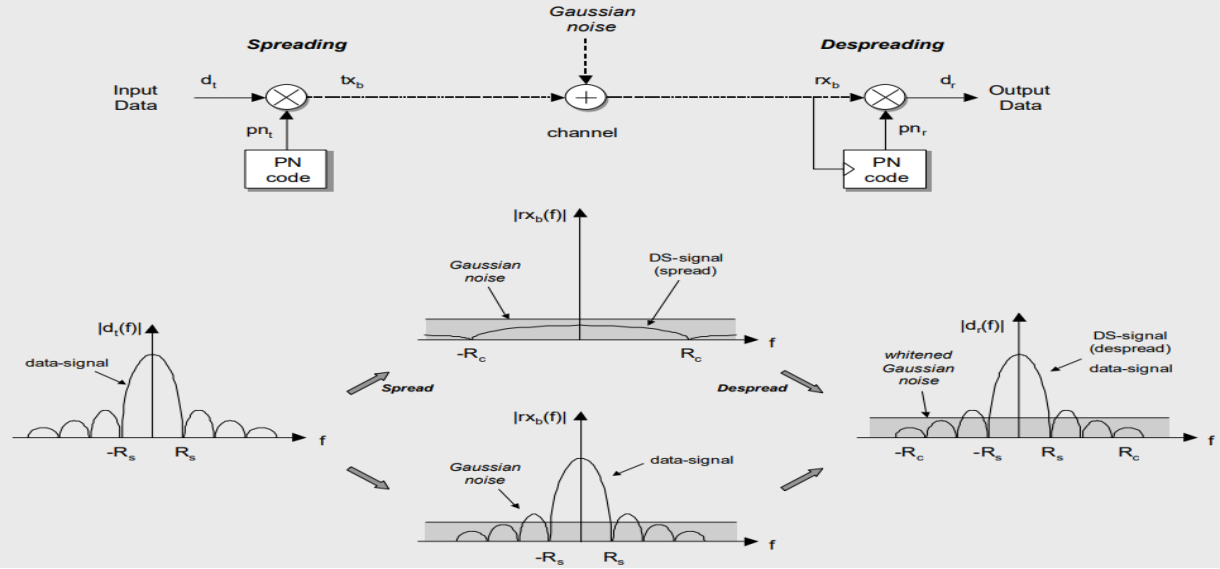


Figure 5. 11 Gaussian noise

i. UAVS and drones' communication:

1) Drones depend mainly on GNSS for positioning and navigation.

2) It also depends on RF communication link between the controller and drone.

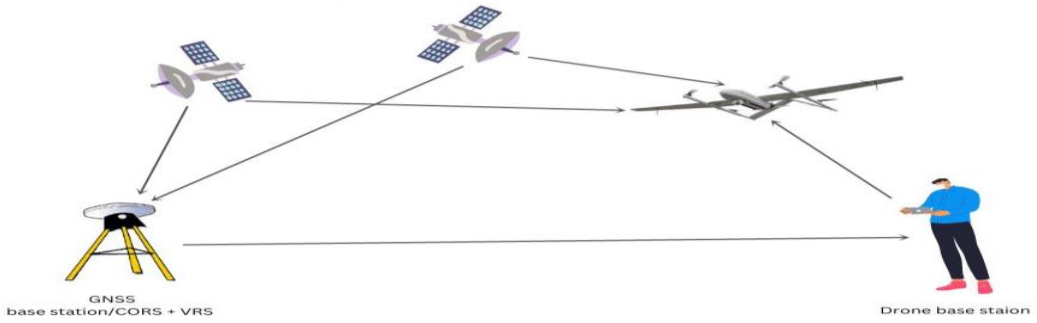


Figure 5. 12 Drones Communication methods

5.5 Performance Parameters

There are two parameters used for measuring the performance of the signal:

Bit Error Rate (BER): The Bit Error Rate (BER) is a measure of the number of bits received incorrectly in a transmission compared to the total number of bits sent. It is used to quantify the quality and performance of a communication system. A lower BER indicates a more reliable and accurate transmission. It is usually expressed as a ratio or percentage,

where:

$$\text{BER} = \frac{\text{Number of bit errors}}{\text{Total number of bits transmitted}}$$

A BER of 0 means perfect transmission, while higher values indicate poor transmission quality.

Jamming-to-Signal Ratio (JSR): The Jamming-to-Signal Ratio (JSR) is the ratio of the power level of jamming signals (undesired interference) to the power level of the legitimate communication signal. It is a measure of how much the jamming signal interferes with the reception of the desired signal. A higher JSR means more interference, leading to greater potential for signal degradation or failure. It can be expressed in decibels (dB):

$$\mathrm{JSR(dB)} = 10\log_{10}(\frac{\mathrm{P_{jam}}}{\mathrm{P_{signal}}})$$

Where $\mathrm{P_{jam}}$ is the power of the jamming signal, and $\mathrm{P_{signal}}$ is the power of the desired signal.

5.6 intentional interference:

There are three types of intentional interference on GNSS:

5.6.1 Jamming techniques

jamming is an intentional transmission of a high-power radio frequency signal equal to or very close to the frequency of the device whose operation is to be prevented. It aims to prevent the receiver from collecting and tracking GNSS signals and navigating using GNSS signals. Jamming occurs due to the transmission of high-power radio frequencies near the L1, L2, and L5 frequency bands on which GNSS receivers operate. Jamming frequencies are intended to overload the receivers to the point that the receivers lose lock on the satellites and have the main effect of rendering the GNSS system ineffective or degraded for users in the jammed area. Because many devices transmit on frequencies close to GNSS receivers, it is possible that some of these devices unintentionally interfere with GNSS signals. Jamming is even more problematic than spoofing because GPS jammers are relatively much simpler in comparison to spoofing devices. In addition, they are easier to make than spoofing devices. Even small jammers that fit in the palm of a hand can have a range of several meters. A jammer can block all radio communications on any device operating on radio frequencies within its range and emit radio frequency waves that prevent the target device from establishing or maintaining the connection. Spoofing is more complex since the fake signal's structure should be imitated and like the authentic one.

Types:

1. Suppression Jam ming:

The satellite navigation signal is suppressed by transmitting a jamming signal that has high power in the frequency band of the satellite navigation signal. In addition, the signal-to-noise ratio of the receiver is reduced, and the satellite navigation signal is annihilated with the jamming signal. As a result, receiver positioning accuracy is reduced or unable to work properly.

There are four types of suppression jamming:

1. Barrage jamming.
2. Tone jamming (Continuous wave (CW) jamming).
3. Sweep jamming.
4. Pulse jamming.

5.6.2 Barrage jamming

It is the simplest form of wideband interference.

It is generally defined as a jammer that transmits noise-like energy throughout the portion of the spectrum occupied by the target.

Advantages

- Effective against communication systems operating on multiple frequencies.
- It is harder for the target to filter out or avoid the jamming signal.

Disadvantages

- Energy Consumption: Barrage jamming requires significant power to maintain a strong signal across a wide frequency range.
- Detection Risk: The high-power signal can make the jamming source detectable and vulnerable to countermeasures.

Power equation: $J(t) = \frac{P}{n(t)}$

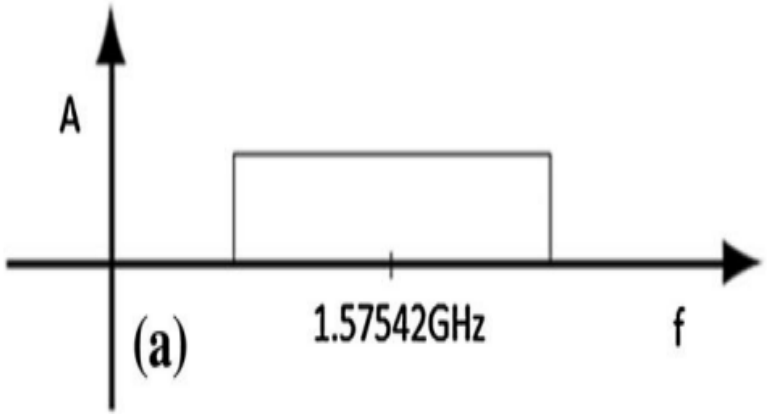


Figure 5. 13 Theoretical spectrum of barrage jamming

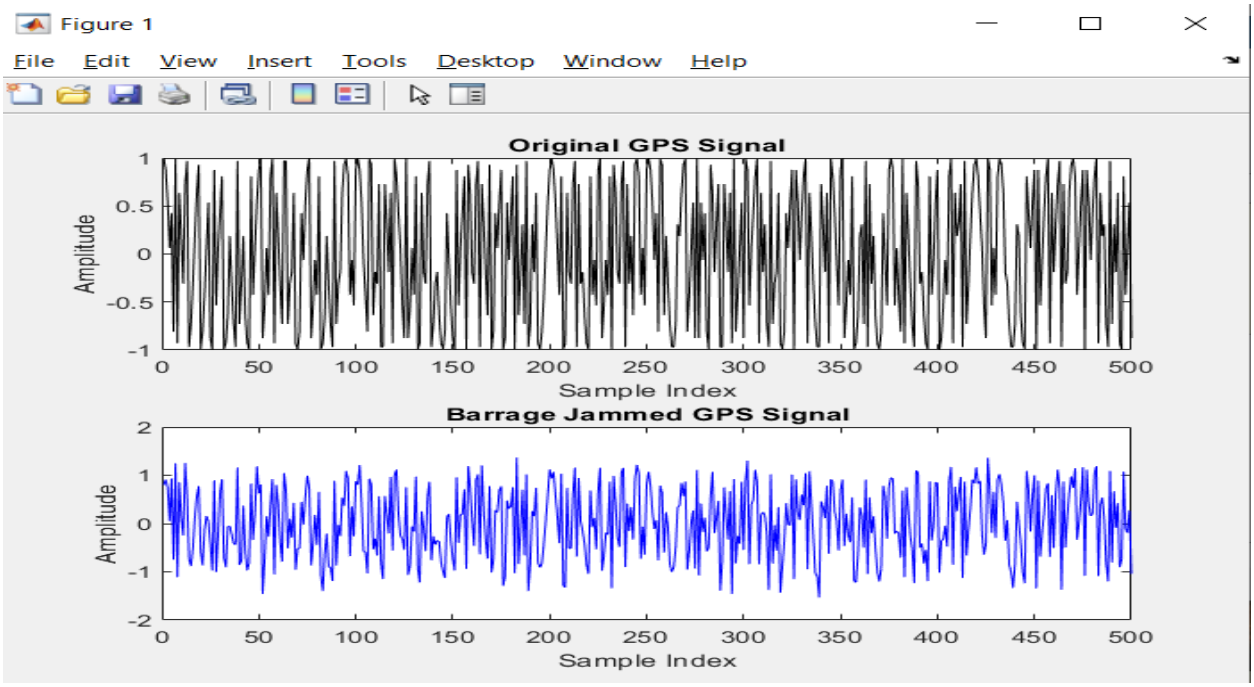


Figure 5. 14 Simulated signal characteristic diagram of Barrage jamming

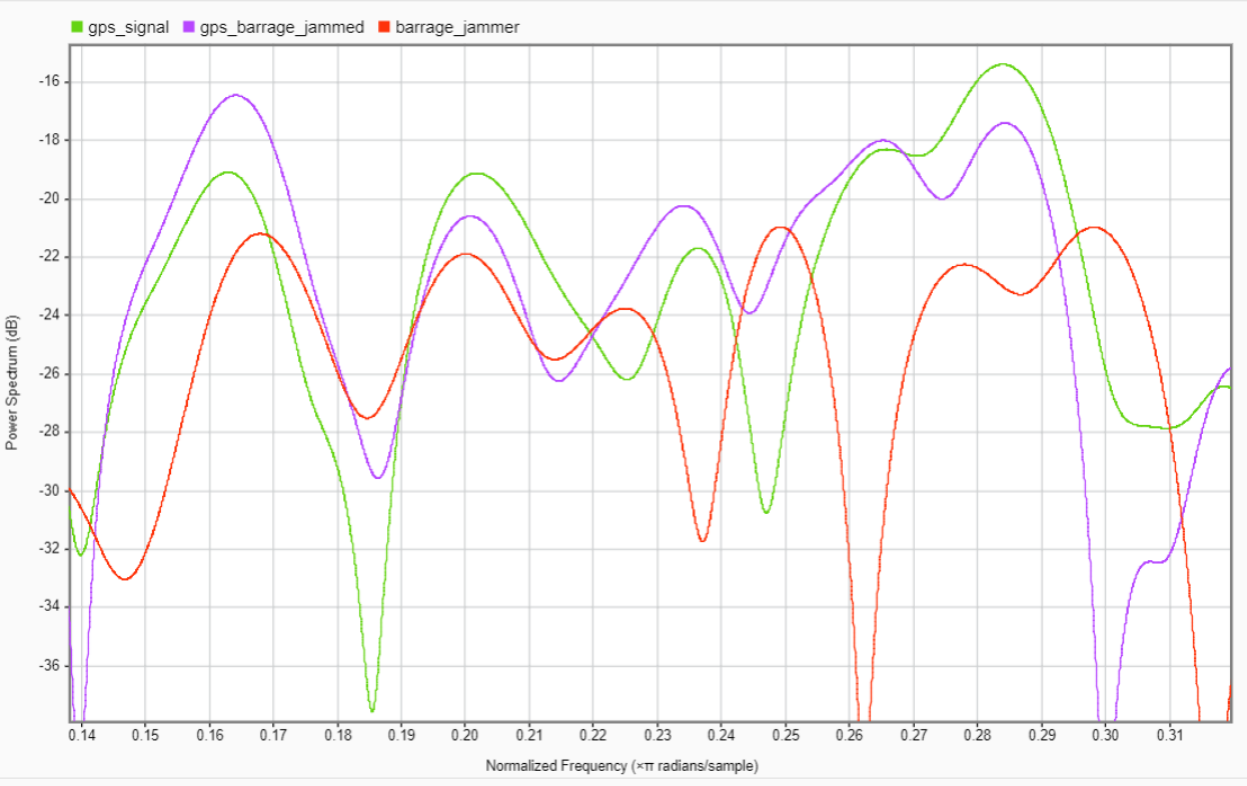


Figure 5. 15 Effect of barrage jamming on GPS L1 signal

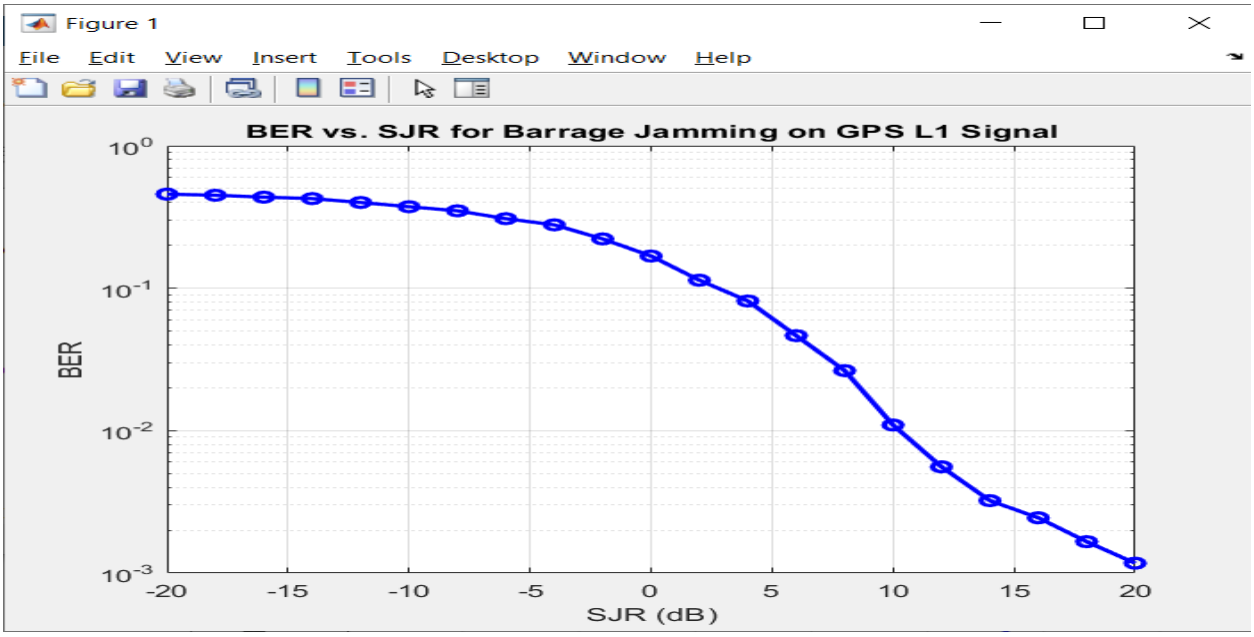


Figure 5. 16 simulated BER vs. SJR for barrage jamming on GPS L1 signal

4.6.3 Tone Jamming (CW)

In this technique, only a sinusoid is transmitted on the same frequency as the carrier of the GPS signal.

This means that the interference is created only at the central frequency and not over the whole bandwidth.

Advantages

- Simple to implement compared to more complex jamming techniques.
- Highly effective against narrowband systems or systems that rely on specific frequencies.

Disadvantages

- May only affect a specific frequency, leaving other frequencies or systems unaffected.
- It can be easily detected, as it creates a distinct spike in the frequency domain.

Power equation: $J(t) = A\cos(2\pi F_c t)$

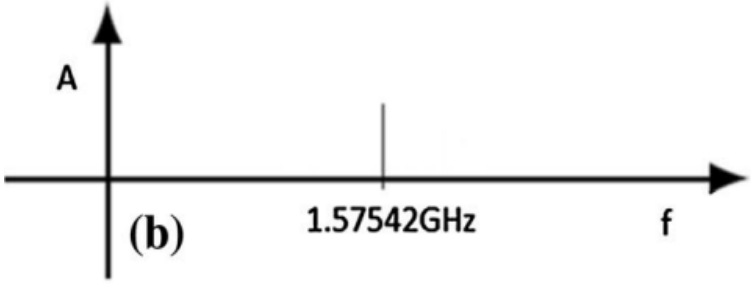


Figure 5. 17 Theoretical spectrum of Tone jamming

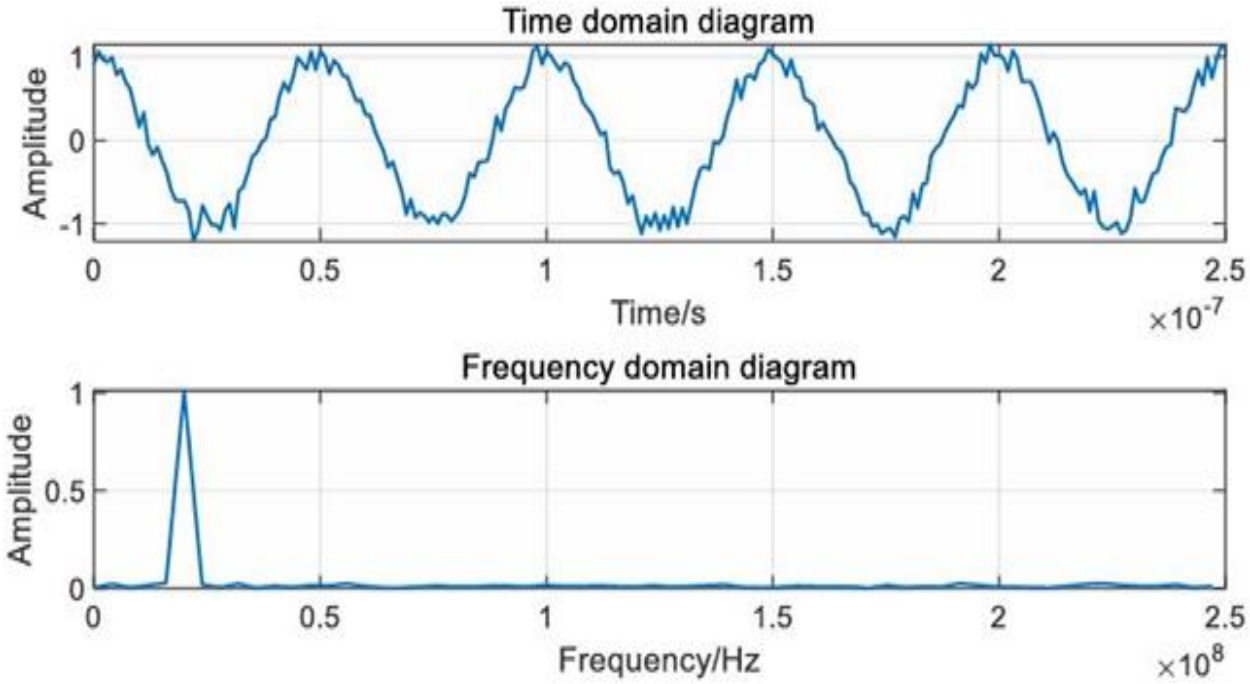


Figure 5. 18 The signal characteristic diagram of Tone jamming

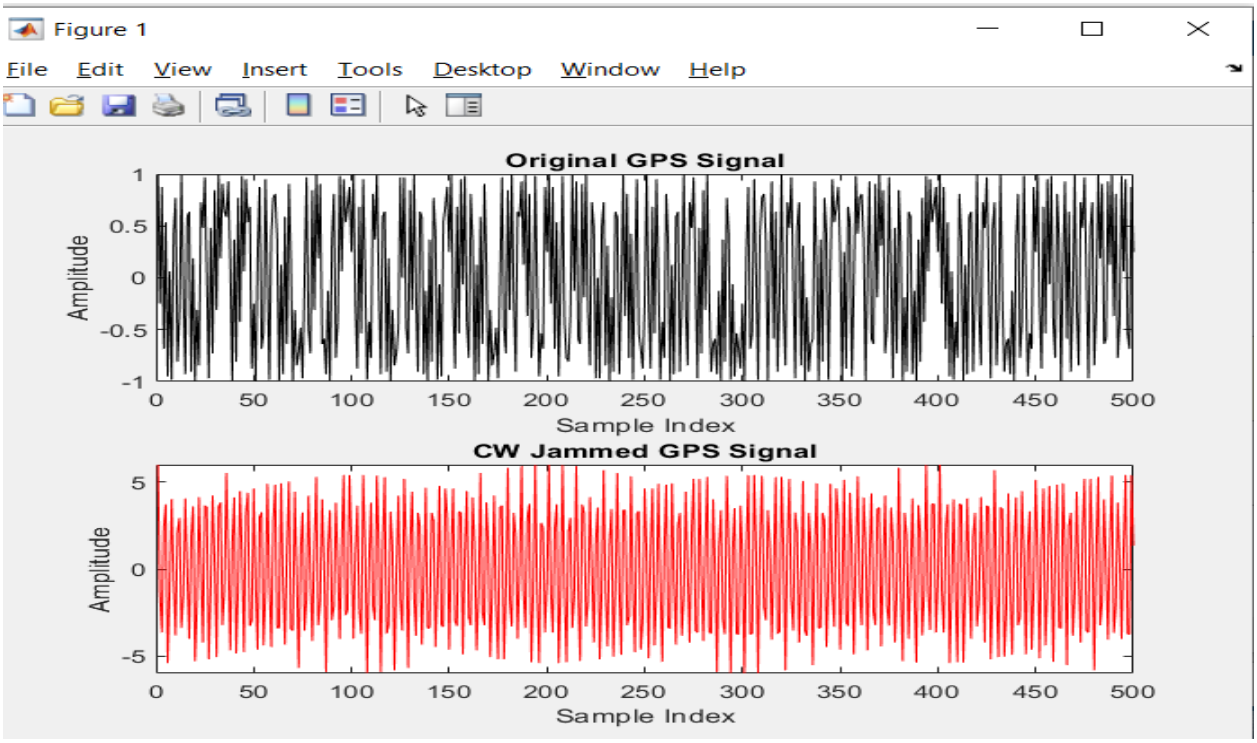


Figure 5. 19 simulated signal characteristic diagram of Tone jamming

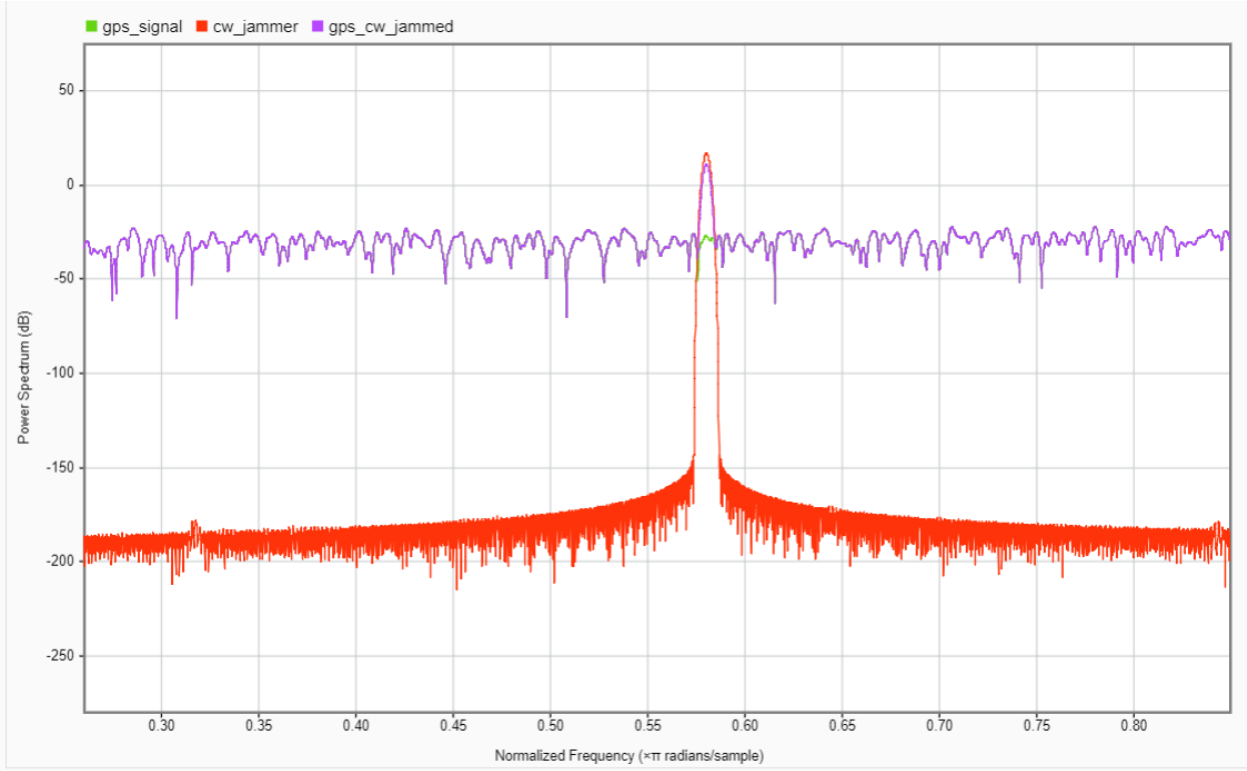


Figure 5. 20 Effect of Tone jamming on GPS L1 signal

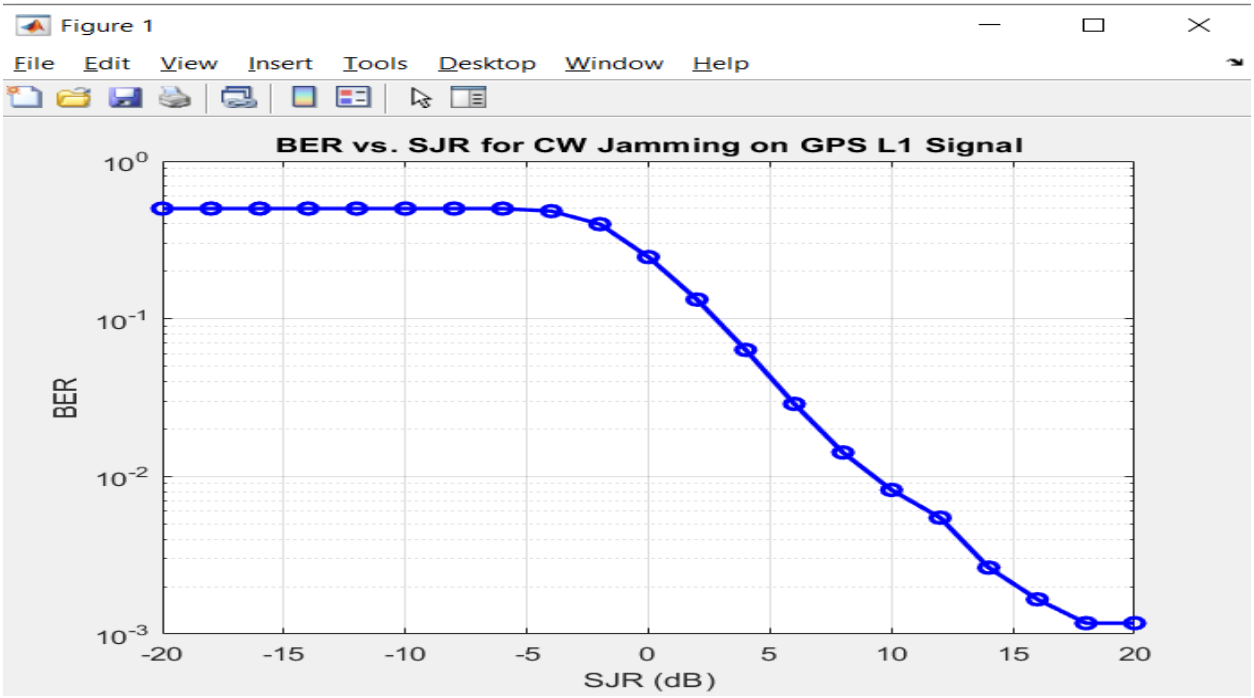


Figure 5. 21 simulated BER vs. SJR for Tone jamming on GPS L1 signal

5.7.4 Sweep jamming

It is a technique that tries to replicate a behavior very similar to Barrage Jamming because it operates over the whole frequency band.

It is a modification of spot jamming where the jammer's full power is shifted from one frequency to another.

Advantages

- It can simultaneously disrupt multiple communication channels or systems operating at different frequencies within the sweeping range.
- Changing frequency of the jamming signal makes it harder for the target to implement countermeasures like frequency hopping or filtering.

Disadvantages

- Less focus than pulse jamming, potentially reducing overall effectiveness.

Power equation: $J(t) = A\,\mathbf{COS}(2\pi(f_c + f_{sweep}t)t)$

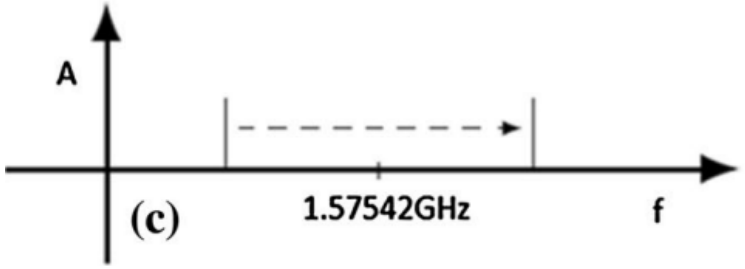


Figure 5. 22 Theoretical spectrum of Sweep jamming

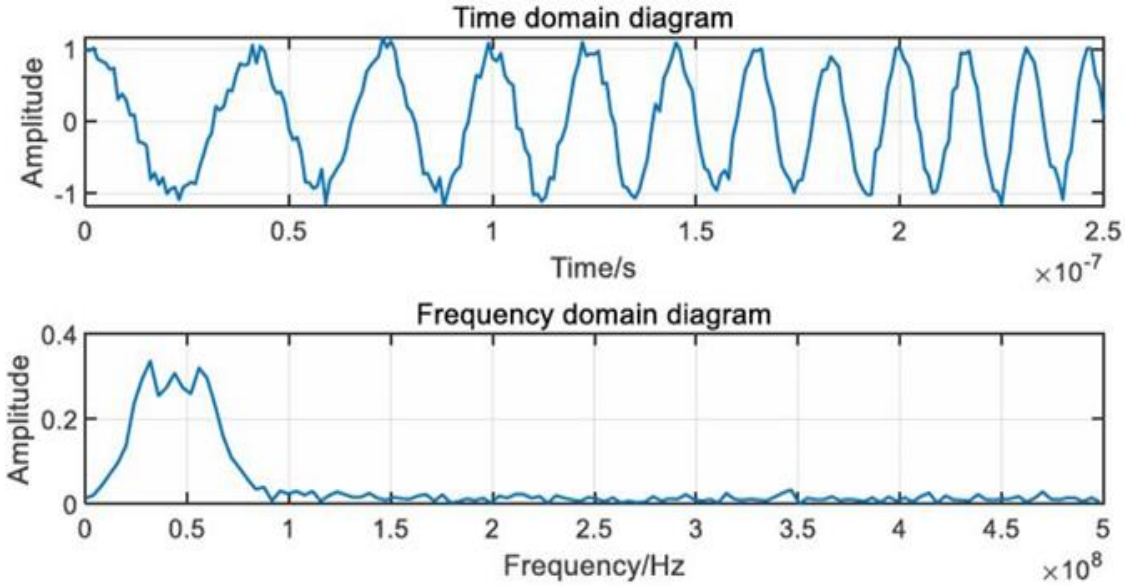


Figure 5. 23 The signal characteristic diagram of sweep jamming

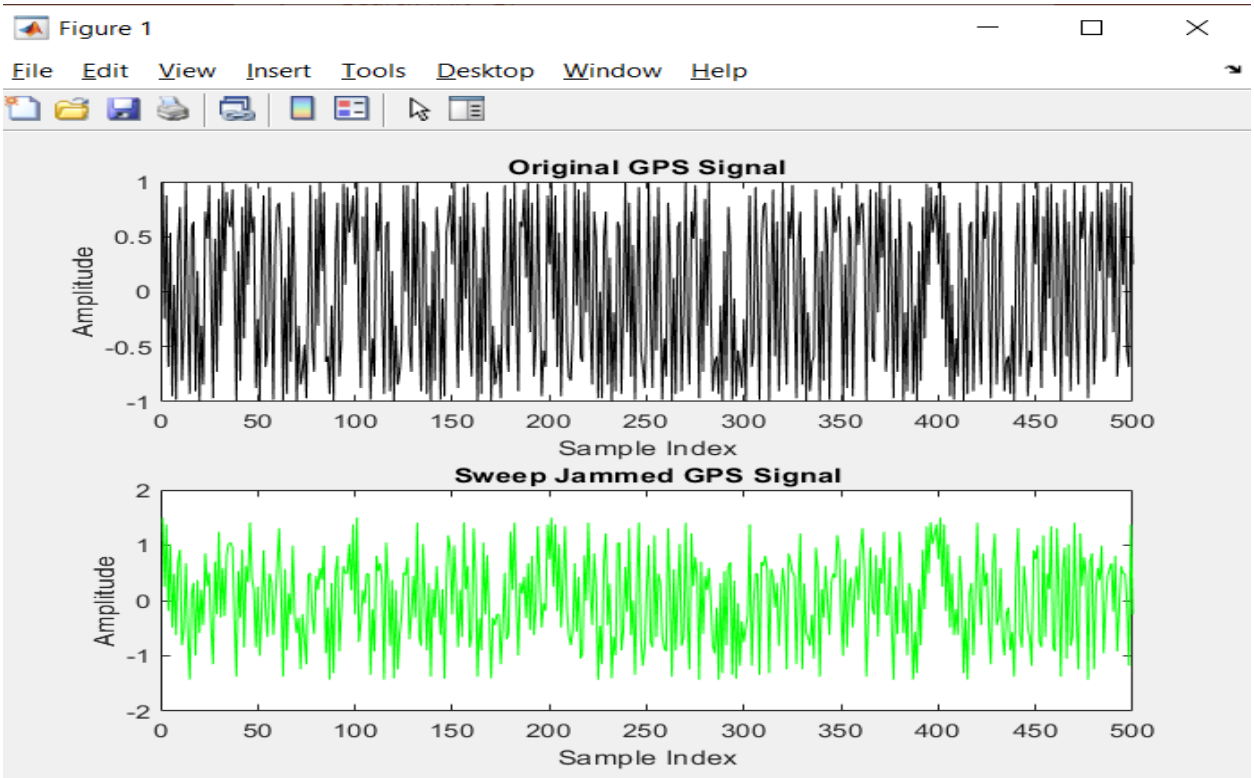


Figure 5. 24 simulated signal characteristic diagram of sweep jamming

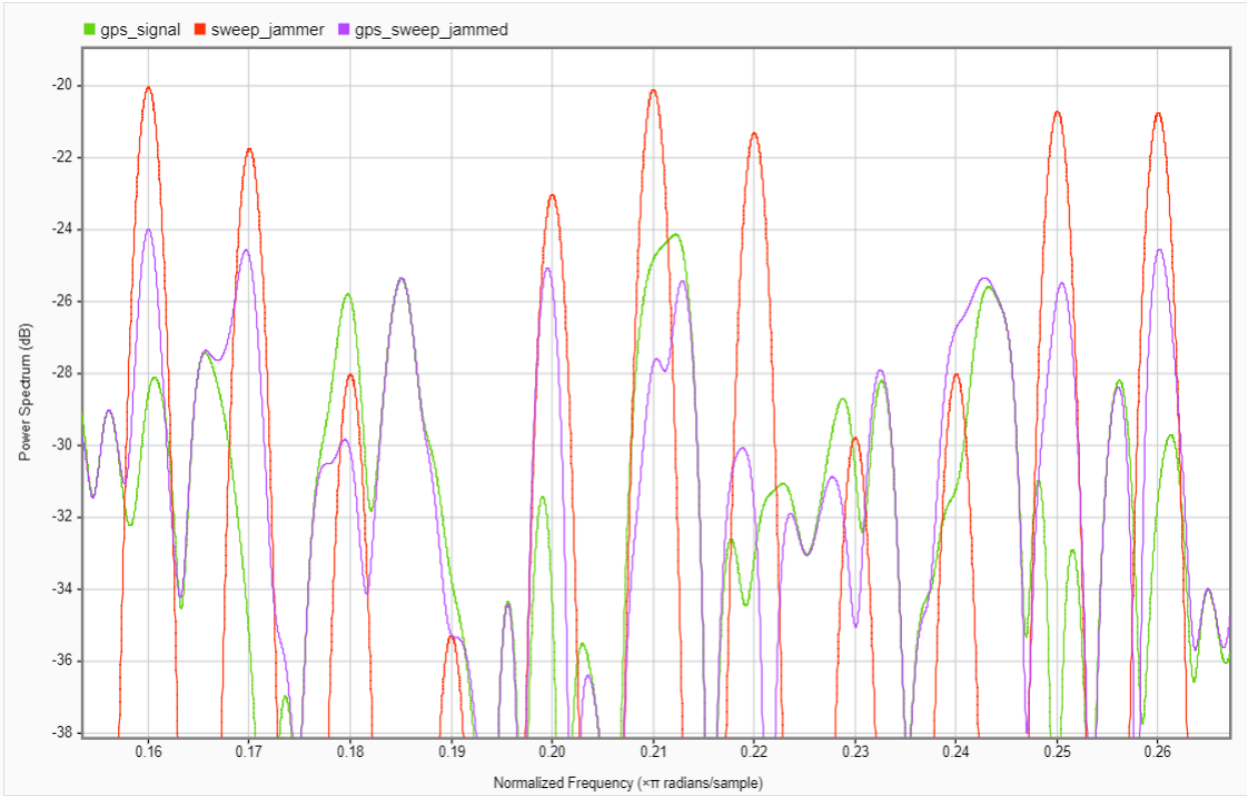


Figure 5. 25 Effect of sweep jamming on GPS L1 signal

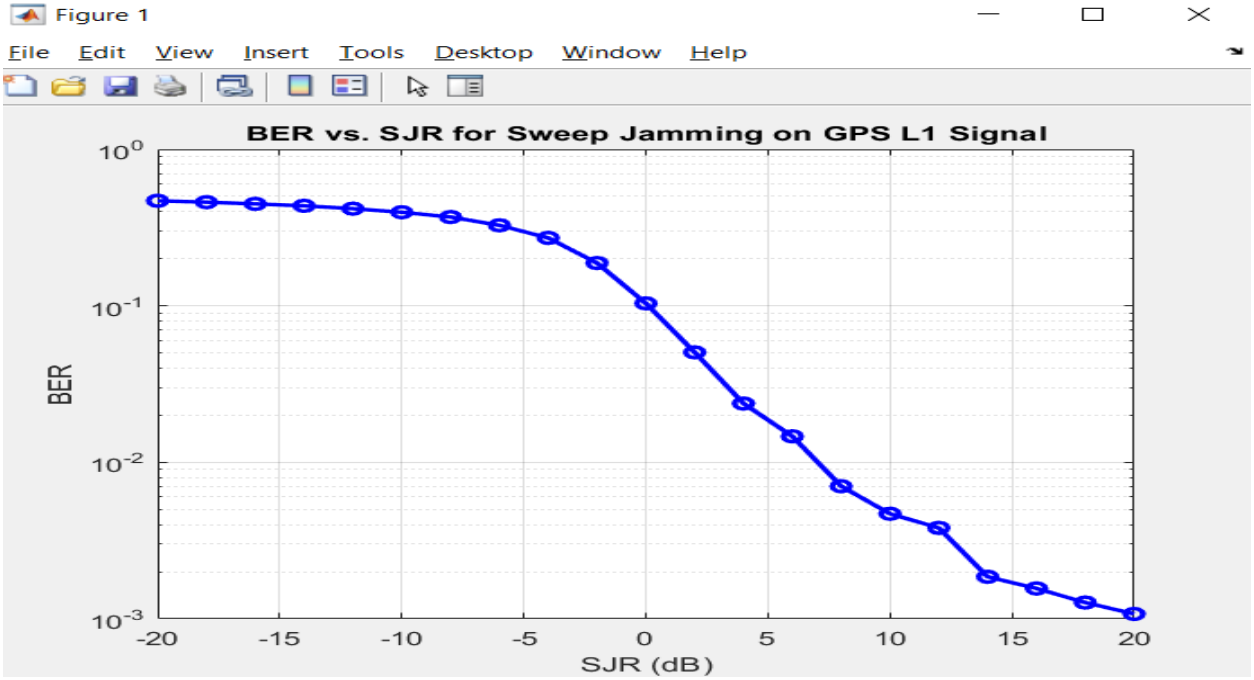


Figure 5. 26 simulated BER vs. SJR for Sweep jamming on GPS L1 signal

5.7.5 Pulse jamming

It refers to a form of electronic warfare where short bursts or "pulses" of electromagnetic energy are transmitted to disrupt or interfere with the operation of a specific target.

Advantages

- It can be highly effective against specific communication systems.

Disadvantages

- Ineffective against multiple or frequency-hopping systems.
- Requires precise knowledge of target signal characteristics.

Power equation: $J(t) = A\cos(2\pi f_c\, t) * s(t))$

A: is the amplitude of the pulse jamming signal,
fc: is the GPS L1 carrier frequency,
s(t): is the ideal rectangular wave signal.

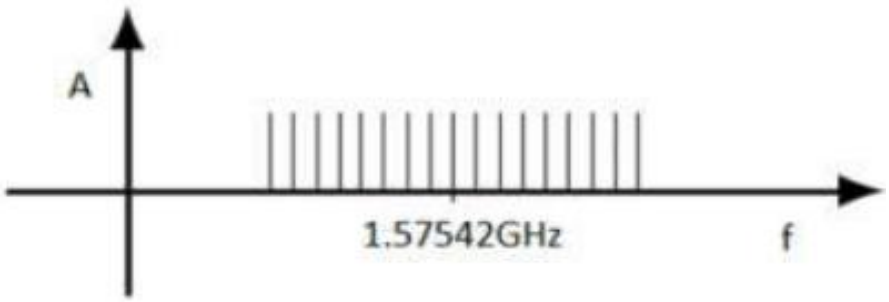


Figure 5. 27 Theoretical spectrum of Pulse jamming

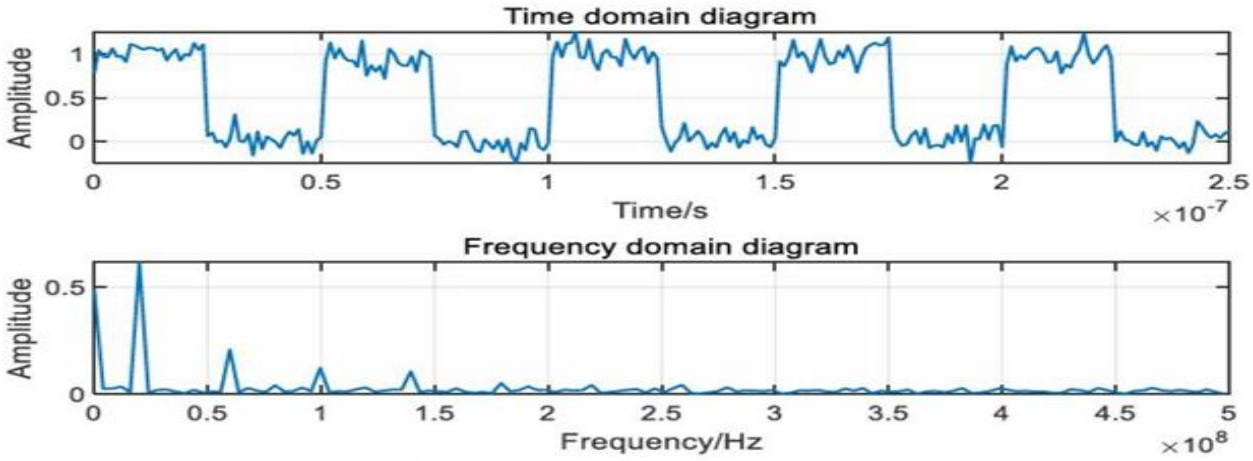


Figure 5. 28 The signal characteristic diagram of Pulse jamming

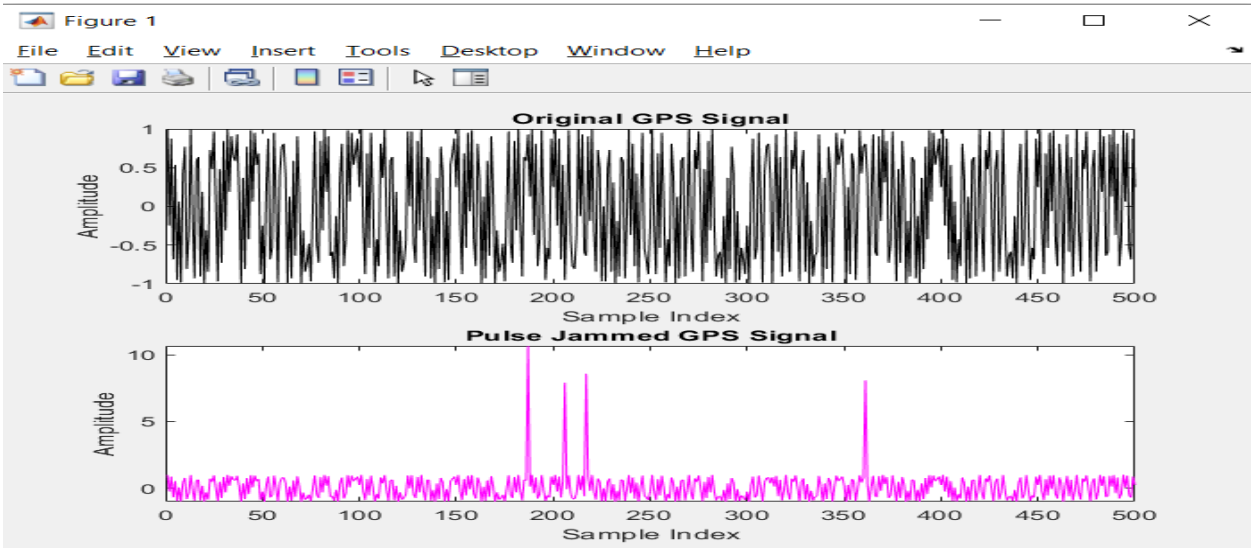


Figure 5. 29 simulated signal characteristic diagram of Pulse jamming

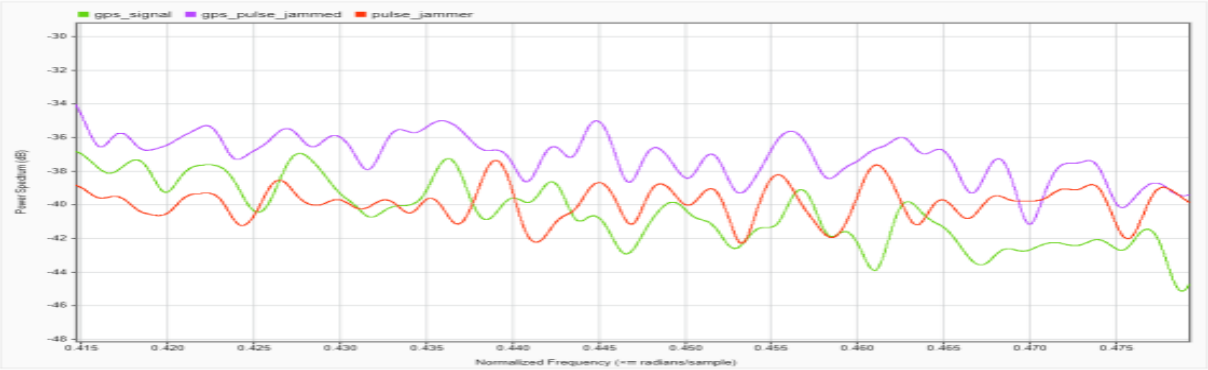


Figure 5. 30 Effect of Pulse jamming on GPS L1 signal

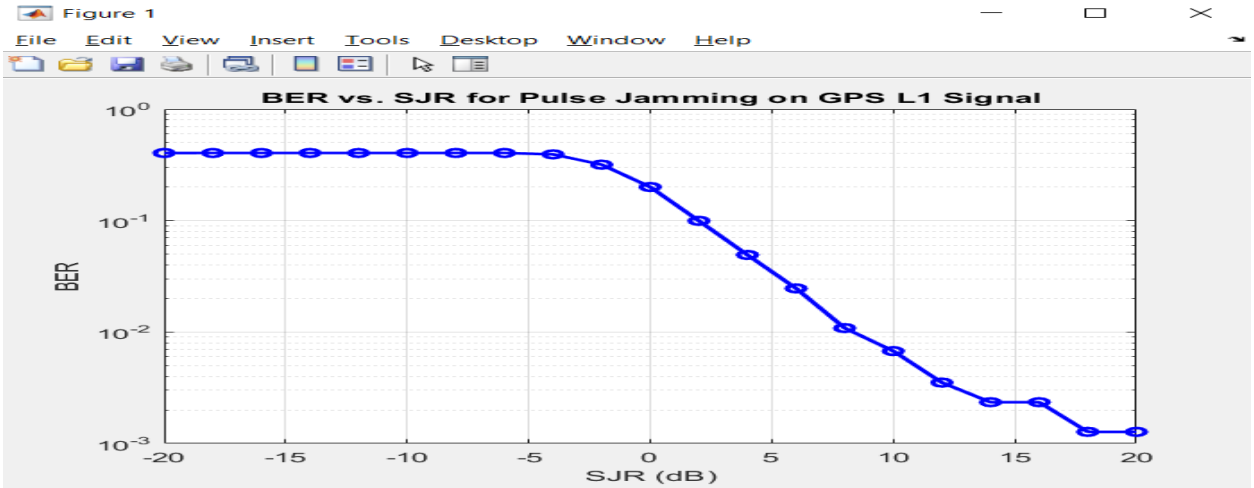


Figure 5. 31 simulated BER vs. SJR for Pulse jamming on GPS L1 signal

Different jamming techniques:

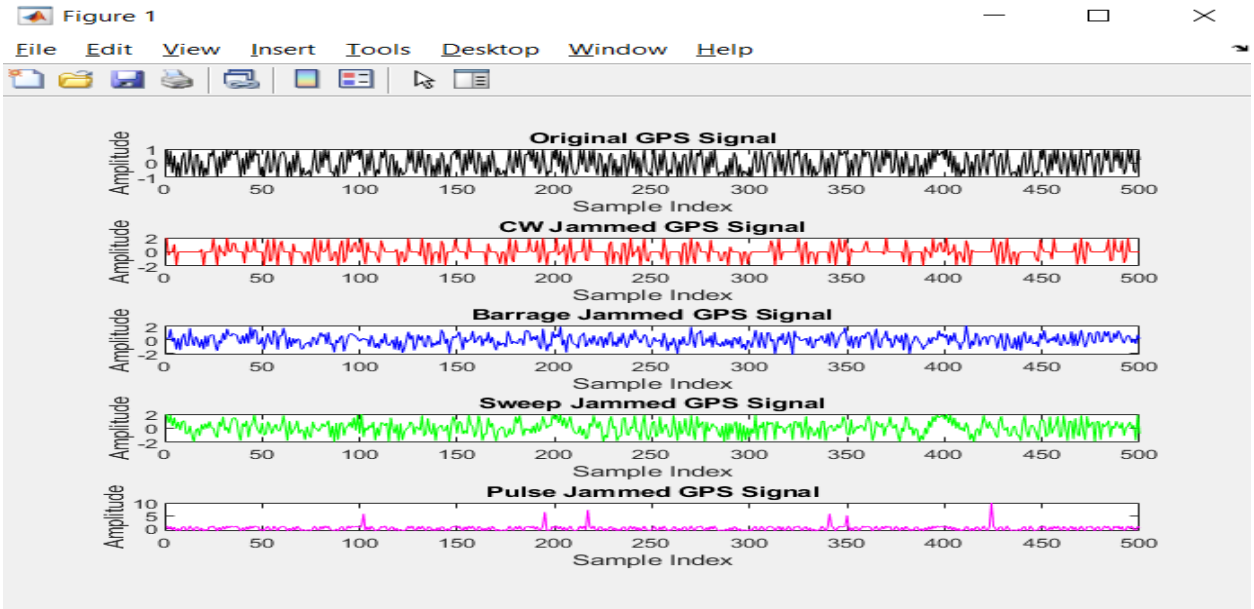


Figure 5. 32 Simulated signal characteristic diagram of different jamming techniques

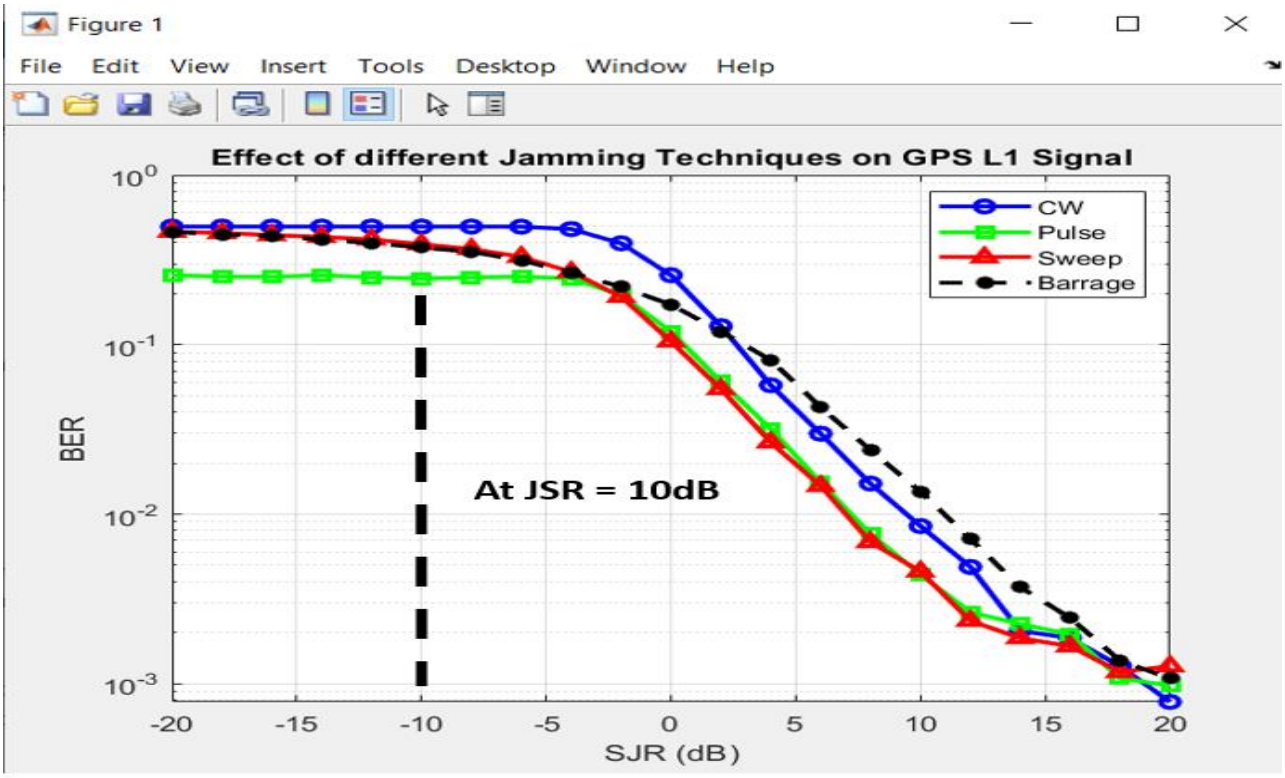


Figure 5. 33 simulated BER vs. SJR for different jamming techniques on GPS L1 signal

- **Comparison between Different jamming techniques:**

Type of Jamming	Advantages	Disadvantages
Pulse Jamming	- Can disrupt signals at specific frequencies. - Efficient for short bursts of interference. - Can be highly effective against specific radar or communication systems.	- Limited by the ability to identify the target signal's pulse repetition interval. - May require sophisticated timing for effectiveness. - May be easily countered by advanced systems with frequency agility.
Sweep Jamming	- Covers a broad range of frequencies. - Can target systems with broad bandwidth. - Effective against frequency-hopping systems.	- Less focused than pulse jamming, potentially reducing overall effectiveness. - Can be energy-inefficient. - May result in detection of jamming sources due to the broad spectrum of interference.
Tone Jamming	- Simple to implement. - Can effectively block narrowband signals. - Good for interfering with communication channels at specific frequencies.	- Not effective against systems with frequency agility or wideband signals. - Can be easily identified and localized due to its continuous nature. - Limited effectiveness if the target system uses sophisticated filtering or hopping techniques.
Barrage Jamming	- Covers a wide frequency spectrum, making it harder for the target to adapt. - Effective against wideband signals and multi-channel systems. - Harder to locate due to its broad-spectrum nature.	- Requires significant power to cover a wide range of frequencies. - Can cause substantial interference to non-target systems. - Less precise, as it indiscriminately jams frequencies within the target range.

2. Deception jamming:

works in a way that the source generates a deception signal which is similar to the real satellite navigation signal. The deception signal has slightly higher power than the real signal. There is one more way to perform such an attack: the source repeats the real satellite or navigation signal in space. With this attack, the attacker achieves that the receiver “picks up” wrong information about the time and location and sends it as such. There are two types of deception jamming: Generated deception jamming is a jamming method in which the attacker generates and transmits the deception signal. The generated signal has the same structure as the real navigation signal. It gradually replaces the real signal in the tracking loop under the signal control strategy and power advantage. After that, it controls the tracking loop to achieve the purpose of deception.

• Repeater deception jamming adds a certain time delay based on receiving the real satellite navigation signal. After that, it repeats the signal through power adjustment to make the satellite navigation receiver receive the repeater signal, and thus a false signal is transmitted.

Deception Type	Advantage	Disadvantage	Scope of Applications
Generated deception Jamming [50–53]	High flexibility and controllable parameters	Unable to deceive military code signal	Civilian signal deception
Repeater deception Jamming [56,57]	Can deceive military code signals	Low success rate of deception	Military signal deception

Figure 5. 34 Comparison of deception methods with different signal generation modes

spoofing:

Spoofing is the provision of GNSS-like signals locally transmitted and coded to fool the receiver. A device sends a movement analogous to the satellite signal, but which has higher power, so the GNSS receives the false signal instead of the real one; in such a manner, it calculates an incorrect position or an incorrect time variable. There is a type of spoofing called a ‘carry-off attack’ that is caused by broadcasted signals synchronized with genuine signals observed by the target receiver. The power of the counterfeit signals is gradually increased, so that the GNSS receiver tracks the false signals which can be manipulated by giving a different location to the provided by genuine signals; to simplify, spoofing causes the receiver to lie being this attack more complex than a jamming attack.

meaconing:

Meaconing is the interception and rebroadcast of navigation signals. These signals are rebroadcast on the received frequency, typically with power higher than the original signal, to confuse enemy navigation. Consequently, aircraft or ground stations are given inaccurate bearings. Meaconing is more of a concern to personnel in navigation ratings than to radio operators. However, communications transmitters are often used to transmit navigation signals. Since communications personnel operate the transmitters, they must know how to deal with any communications problems resulting from meaconing. Successful meaconing can cause aircraft to be lured into "hot" (ambush-ready) landing zones or enemy airspace, ships to be diverted from their intended routes, bombers to expend ordnance on false targets, or ground stations to receive inaccurate bearings or position locations.

5.8 ADF4351 PLL Frequency-Synthesizer Module

The ADF4351 module is the core of the project's RF-signal-generation stage. By integrating a high-performance fractional-/integer-N phase-locked loop (PLL) and a wideband voltage-controlled oscillator (VCO) on a single chip, it produces stable output frequencies from 35 MHz up to 4.4 GHz under digital control. This frequency agility is fundamental to tasks such as drone-link disruption, laboratory signal generation, and general wireless-system testing.

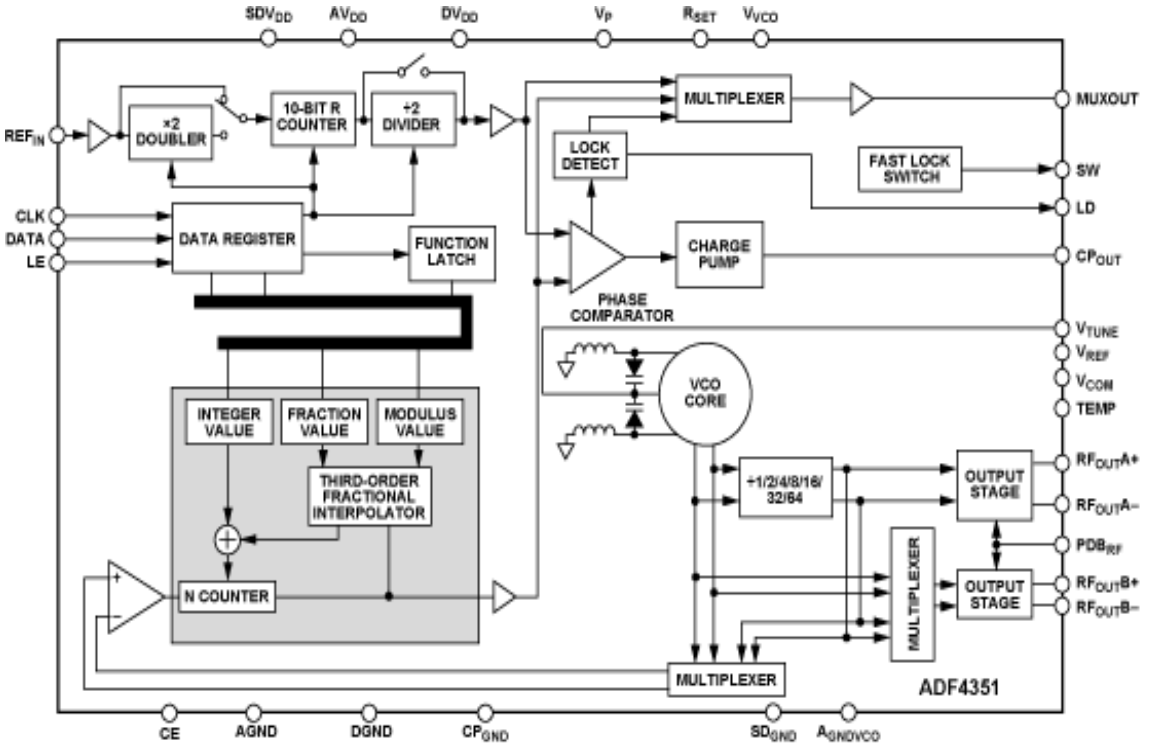


Figure 5. 35 Functional block diagram

5.8.1 Functional Overview

- Wide tuning span (35 MHz – 4.4 GHz) enables both narrow-band and wide-band operation.
- Fractional-/Integer-N PLL architecture delivers fine resolution with low phase noise.
- SPI,(Serial Peripheral Interface),(CLK, DATA, LE) allows real-time frequency updates via an external micro-controller (e.g., STM32 or Arduino).
- Lock-detect & fast-lock signals ensure the synthesizer is fully locked before RF power is applied, improving reliability.
- Dual differential outputs (RFoutA ±) simplify routing to downstream amplifiers or mixers

5.8.2 Key Electrical Characteristics

The ADF4351 synthesizer module exhibits high-performance electrical specifications that enable precise, and reliable RF signal generation. Understanding these characteristics is crucial for effective integration into systems such as signal jammers.

The table below summarizes the key electrical parameters of the module along with their technical implications and relevance to the project.

Parameter	Typical Value / Range	Description and Relevance
Output Frequency Range	35 MHz to 4.4 GHz	Wide tuning range covers multiple RF bands (VHF, UHF, L, S). Enables multi-band jamming and testing.
Reference Input (EXT_REF)	10 MHz (external)	Stable clock input required for frequency synthesis. Determines output stability and accuracy.
Frequency Resolution	< 1 kHz (fractional-N mode)	Fine tuning steps allow precise frequency selection, essential for targeting narrow-band signals.
Phase Noise (1 kHz offset @ 1 GHz)	~ −110 dBc/Hz	Low phase noise results in cleaner signals, reducing interference and improving jamming effectiveness.
Supply Voltage	3.3 V to 5 V (typ. 3.3 V)	Requires a clean and stable power supply. Low-noise LDOs used to avoid RF degradation.
Lock Time	< 50 µs (fast-lock enabled)	Fast switching between frequencies allows real-time hopping, critical for counter-drone systems.
Output Power Level	Programmable: −4 to +5 dBm	Adjustable output level suits various downstream RF circuits like amplifiers and filters.
Temperature Range	−10°C to +55°C	Proven stable operation during lab tests across typical ambient conditions.
Harmonic Suppression	Improved via internal filtering	Reduces unwanted spectral components. External filtering is used for enhanced spectral purity.

5.8.3 Module Pin Configuration

The ADF4351 module includes several input/output pins and connectors that facilitate power delivery, digital control, and RF signal output. Understanding the function of each pin is essential for proper integration with microcontrollers, reference sources, and RF components.
The following table summarizes the key pins and connectors used in the project:

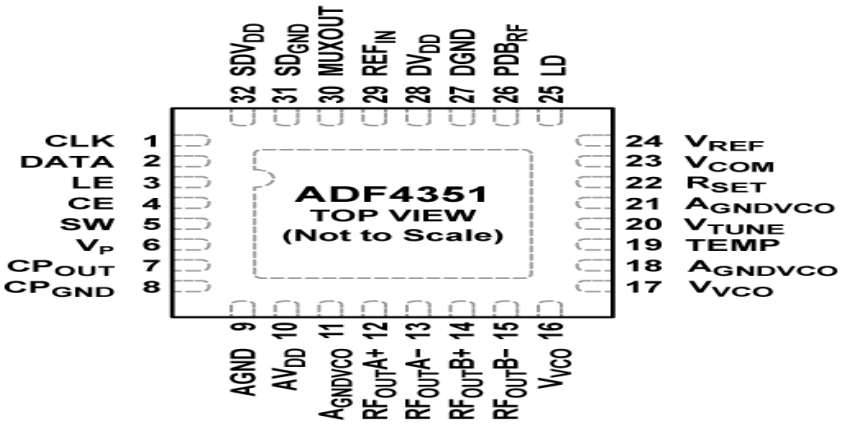


Figure 5. 36 Pin configuration

Pin / Connector	Function
VCC	Main power supply input (3.3V to 5V). A low-noise voltage source is recommended.
GND	Electrical ground. Connects to system ground for proper reference.
CLK (Clock Input)	SPI clock signal. Used to synchronize data transmission during programming.
DATA (Serial Data)	SPI data line. Carries configuration data to the module from the microcontroller.
LE (Latch Enable)	SPI latch enables pin. Data is latched into the ADF4351 registers on rising edge.
CE (Chip Enable)	Used to enable or disable the chip. High = Enabled, Low = Powered down.
MUXOUT	Multipurpose output. Can be configured to show lock-detect status, R counter, etc.
EXT_REF	External reference clock input. Accepts 10 MHz (or other) signal from OCXO or crystal oscillator.
RFoutA+ / RFoutA−	Differential RF output signals. Provide the synthesized frequency output. Connected to RF path.
OSC (Optional)	Pads available for mounting an onboard crystal oscillator (not used in this project).

5.8.4 Integration in the System

The ADF4351 module was successfully integrated into the system, with careful consideration given to control and power regulation.

• Control Interface : The module is controlled by an Arduino Uno via an SPI interface. A dedicated routine program the six configuration registers of the ADF4351 during system startup and dynamically updates them whenever a new jamming frequency is required.

• Power Conditioning : A low-noise 3.3 V low-dropout (LDO) regulator with an output noise of 40 µVrms is used to power the analog section of the synthesizer. The digital I/O lines are designed to be 3.3 V-tolerant for compatibility with the MCU.

5.8.5 Contribution to Project Objectives

The ADF4351 synthesizer played a pivotal role in supporting the project's technical goals by offering the following advantages:

• Enabled agile, frequency-hopping jamming, effectively countering multi-band commercial drones.

• Facilitated laboratory calibration of receiver front ends without the need for external signal generators.

• Allowed rapid waveform prototyping and flexibility through firmware updates instead of costly

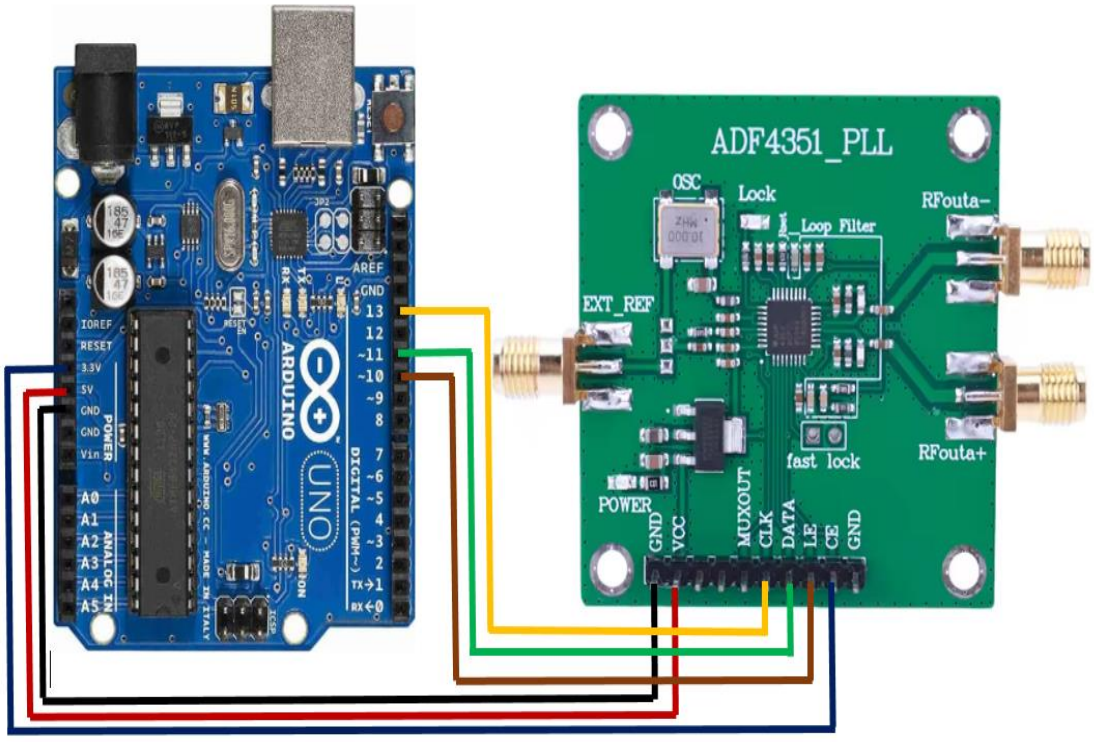


Figure 5. 37 ADF4351 to Arduino Uno connection

GND → GND
VCC → 5V
CLK → PIN 13
DATA → PIN 11
LE → PIN 10
CE → 3.3V

5.8.6 Programming ADF4351 via SPI (1575 MHz)

1575 §

```
#include <SPI.h>

// Define LE pin
const int LE = 10;

void setup() {
  Serial.begin(9600);
  SPI.begin();
  SPI.beginTransaction(SPISettings(5000000, MSBFIRST, SPI_MODE0));

  pinMode(LE, OUTPUT);
  digitalWrite(LE, LOW);

  delay(100);

  // Register values for 1575 MHz
  uint32_t regs[] = {
    0x00580005,   // R5
    0x0095003C,   // R4
    0x000004B3,   // R3
    0x00004E42,   // R2
    0x08000011 ,   // R1
    0x013B0000     // R0 (last to latch)
  };
```

```
  // Write registers R5 to R0
  for (int i = 0; i < 6; i++) {
    writeADF4351(regs[i]);
    delay(10);
  }

void loop() {
  // Nothing to do in loop
}

void writeADF4351(uint32_t reg) {
  digitalWrite(LE, LOW);

  // Send 4 bytes MSB first
  SPI.transfer((reg >> 24) & 0xFF);
  SPI.transfer((reg >> 16) & 0xFF);
  SPI.transfer((reg >> 8) & 0xFF);
  SPI.transfer(reg & 0xFF);

  digitalWrite(LE, HIGH);
  delayMicroseconds(1);
  digitalWrite(LE, LOW);
```

5.8.6.1 Code explanation

- **Libraries Used**:
  - SPI.h enables SPI communication between the Arduino and the ADF4351 frequency synthesizer.
- **Pin Definition**:
  - LE (Latch Enable) pin is defined as digital pin 10 for SPI communication control.
- **Setup Function**:
  - Initialize serial communication at 9600 bps for debugging.
  - Starts SPI communication with:
    - Clock speed: 5 MHz
    - Data order: MSB first
    - SPI mode: 0
  - Sets LE as an output and initializes it to LOW.
  - Introduce a small delay to stabilize the system.
- **ADF4351 Register Configuration**:
  - An array regs[] holds six 32-bit register values corresponding to R5 through R0.
  - These values are pre-calculated by ADF435x software tool to configure the ADF4351 to output a frequency of **1575 MHz**.
  - Registers are written in descending order (R5 to R0), as required by the ADF4351.
- **Function writeADF4351()**:
  - Sends a 32-bit register value to the ADF4351 via SPI in 4 bytes (MSB first).
  - Toggle the LE pin to latch the data into the ADF4351.
  - Includes a short delay to ensure proper timing.
- **Outcome**:
  - Once the setup is complete, the ADF4351 is configured to output a frequency of **1575 MHz**.

F (MHz)	R0	R1	R2	R3	R4	R5
1176	00758008	08008029	00004E42	000004B3	0095003C	00580005
1207	00788010	08008029	00004E42	000004B3	0095003C	00580005
1227	007A8010	08008029	00004E42	000004B3	0095003C	00580005
1278	00D68018	08008029	00004E42	000004B3	00A5003C	00580005
1575	013B0000	08000011	00004E42	000004B3	0095003C	00580005

Figure 5. 38 The pre-calculated registers values for each frequency using ADF435x software tool.

5.8.7 Frequency Hopping with ADF4351 via SPI

Frequency_Hopping

```
#include <SPI.h>

// Define ADF4351 LE pin
const int LE = 10;

// ADF4351 frequency register values for 1176MHz, 1207MHz, 1227MHz, 1278MHz and 1575 MHz
// These values should be calculated based on your ADF4351 reference frequency and configuration
// You can use ADI's ADF4351 Software to get these register values

const uint32_t freq1176[6] = {
    0x00580005,  // R5
    0x95003C,  // R4
    0x000004B3,  // R3
    0x4E42,  // R2
    0x8008029,  // R1
    0x758008   // R0 (1176 MHz)
};

const uint32_t freq1207[6] = {
    0x00580005,  // R5
    0x95003C,  // R4
    0x000004B3,  // R3
    0x00004E42,  // R2
    0x8008029 ,  // R1
    0x788010    // R0 (1207 MHz)
};
```

```
const uint32_t freq1227[6] = {
    0x00580005,  // R5
    0x0095003C,  // R4
    0x000004B3,  // R3
    0x00004E42,  // R2
    0x8008029 ,  // R1
    0x7A8010    // R0(1227 MHz)
};

const uint32_t freq1278[6] = {
  0x00580000,  // R5
  0xA5003C,  // R4
  0x00004B3,  // R3
  0x000004E42,  // R2
  0x8008029,  // R1
  0xD68018   // R0 (1278 MHz)
};

const uint32_t freq1575[6] = {
0x00580005,  // R5
    0x0095003C,  // R4
    0x000004B3,  // R3
    0x00004E42,  // R2
    0x08000011 ,  // R1
    0x013B0000    // R0 (1575 MHz)
};
```

```
void sendADF4351Registers(const uint32_t* regValues) {
  for (int i = 5; i >= 0; i--) {
    digitalWrite(LE, LOW);
    SPI.transfer((regValues[i] >> 24) & 0xFF);
    SPI.transfer((regValues[i] >> 16) & 0xFF);
    SPI.transfer((regValues[i] >> 8) & 0xFF);
    SPI.transfer(regValues[i] & 0xFF);
    digitalWrite(LE, HIGH);
    delayMicroseconds(20); // Small delay to latch data
  }
}

void setup() {
  pinMode(LE, OUTPUT);
  digitalWrite(LE, HIGH);

  SPI.begin();
  SPI.beginTransaction(SPISettings(1000000, MSBFIRST, SPI_MODE0));

  // Start at 1575 MHz
  sendADF4351Registers(freq1575);
}
```

```
  // Start at 1575 MHz
  sendADF4351Registers(freq1575);
}

void loop() {
  // Hop to 1176 MHz
  sendADF4351Registers(freq1176);
  delay(500);   // Hold for 500 ms

    // Hop to 1207 MHz
  sendADF4351Registers(freq1207);
  delay(500);   // Hold for 500 ms

    // Hop to 1227 MHz
  sendADF4351Registers(freq1227);
  delay(500);   // Hold for 500 ms

    // Hop to 1278 MHz
  sendADF4351Registers(freq1278);
  delay(500);   // Hold for 500 ms

  // Hop to 1575 MHz
  sendADF4351Registers(freq1575);
  delay(500);   // Hold for 500 ms
}
```

5.8.7.1 Code Explanation

- **Libraries Used**:
  - SPI.h is included to facilitate SPI communication with the ADF4351 frequency synthesizer.
- **Pin Configuration**:
  - LE (Latch Enable) is set as digital pin 10 and configured as an output to control register latching.
- **Register Arrays for Multiple Frequencies**:
  - Defines six 32-bit registers for each target frequency:
    - **1176 MHz**
    - **1207 MHz**
    - **1227 MHz**
    - **1278 MHz**
    - **1575 MHz**
  - Each array (freqXXXX[]) holds values for registers R5 to R0.
  - These values are precomputed using Analog Devices' ADF4351 software tools based on a specific reference frequency.
- **Function sendADF4351Registers()**:
  - Sends a complete set of six registers (R5 to R0) to the ADF4351 via SPI.
  - Transfers the data in 4 bytes per register, MSB first.
  - Latches each register value by toggling the LE pin with a short delay.
- **setup() Function**:
  - Initializes the LE pin as output and sets it HIGH initially.
  - Begins SPI communication with:
    - Speed: 1 MHz
    - Data order: MSB first
    - Mode: 0
  - Programs the ADF4351 to **1575 MHz** as the initial frequency.
- **loop() Function – Frequency Hopping**:
  - Continuously cycles through all predefined frequencies in this sequence:
    - 1176 MHz → 1207 MHz → 1227 MHz → 1278 MHz → 1575 MHz
  - Holds each frequency for **500 milliseconds** before hopping to the next.

5.9 Antenna Design

The system consists of Single helical antenna which covers the frequency range from 1.1 to 1.7 GHZ.

5.9.1 purpose of antenna in the system

The antenna is a critical component in the system, primarily responsible for transmitting jamming signals to disrupt communication between the drone and its controller. It is designed to operate effectively within the frequency range used by most commercial drones.

In the proposed system, the antenna's role is to radiate electromagnetic energy across targeted frequency bands to interfere with both control signals and navigation aids of unmanned aerial vehicles (UAVs). By doing so, it denies the drone access to essential communication channels, potentially forcing it to return to land, or lose stability depending on its fail-safe configuration.

5.9.2 Type of Antenna Used

A circularly polarized helical antenna was selected due to its ability to cover a wide frequency range and provide consistent performance regardless of the drone's antenna polarization. Circular polarization is particularly beneficial for jamming applications because it increases the likelihood of interfering with signals regardless of their orientation.

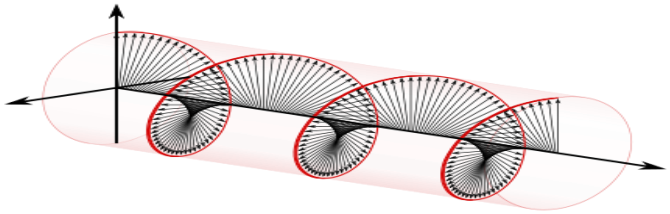


Figure 5. 39 circular polarization

In axial mode operation, the helical antenna produces a directional beam along its axis with relatively high gain and stable radiation characteristics across the operating band. This is particularly important in drone jamming scenarios, where the target is typically at a distance and may be moving dynamically in three-dimensional space. The directional nature of the helical antenna helps concentrate the radiated power toward the target, improving the effective jamming range without significantly increasing the transmitter power.

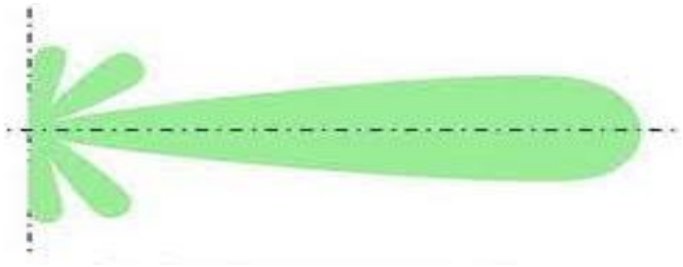


Figure 5. 40 Radiation pattern of helical antenna in axial mode

Additionally, the helical antenna supports a wide bandwidth, which is essential for disrupting multiple communication and navigation signals used by commercial drones. These include GPS (around 1.575 GHz), ISM bands like 2.4 GHz and 5.8 GHz, and other proprietary control frequencies. By designing the antenna to operate efficiently across these frequencies, the system can achieve broad-spectrum jamming coverage with a single radiating element.

The physical construction of the helical antenna is also advantageous. It consists of a conductive wire wound into a helix shape over a dielectric support, with a ground plane at the base. This simple structure is easy to fabricate and customize for different frequency bands by adjusting parameters such as the helix diameter, number of turns, and spacing between turns.

5.9.3 Operating Frequency Band

The antenna is designed to operate within the frequency band of 1.1 GHz to 1.7 GHz, which was carefully chosen to target communication and navigation systems commonly used by commercial and consumer drones. This range includes several critical frequency bands:

- Global Positioning System (GPS): The GPS L1 band operates at 1.575 GHz, a key frequency used by drones for navigation and positioning.
- Telemetry and Control Links: Some drones use lower-frequency bands within this range (e.g., 1.2 GHz or 1.3 GHz) for telemetry or custom control systems.

The selected frequency band ensures that the antenna can effectively radiate jamming signals over all relevant channels without the need for multiple narrowband antennas. A wideband design is essential for counter-drone applications, as drone manufacturers often use various frequency allocations depending on region, brand, or operating mode.
The wide operational bandwidth ensures compatibility with a broad set of UAV models and technologies, improving the effectiveness of the jamming system.

5.9.4 Helical Antenna Parameters and Design

The design of the helical antenna for this system is based on the principles of Kraus' Equation, which is instrumental in determining the key design parameters necessary to achieve the required performance. The helical antenna is chosen for its ability to provide circular polarization, which is essential for the jamming system to disrupt drone communications effectively, regardless of the orientation of the drone’s antenna.

5.9.4.1 Kraus' Equation in Antenna Design

Kraus' Equation is widely used for designing helical antennas, as it relates the antenna's physical characteristics (such as diameter, number of turns, and pitch angle) to its gain, radiation resistance, and polarization. The formula for circularly polarized helical antennas in the axial mode is given by:

$$G = 15N \cdot \frac{S^2}{\lambda} \cdot (\frac{C}{\lambda})$$

where G represents the gain, n is the number of turns, D is the diameter of the helix, λ is the wavelength, and α is the pitch angle. This equation was used to calculate and fine-tune the key design parameters to ensure the antenna operated effectively within the required frequency range of 1.1 GHz to 1.7 GHz.

To operate a helical antenna in axial mode, design parameters must be carefully evaluated and adjusted through simulation. While the wire diameter has minimal impact on performance, the pitch angle should ideally fall between 12° and 14° for optimal results.

The pitch angle of a helical antenna is the angle between the turn of the wire and the base (horizontal) plane. If the wire loops are close together, the angle is small. If they spread out, the angle is bigger. This angle helps control how the antenna sends out signals.

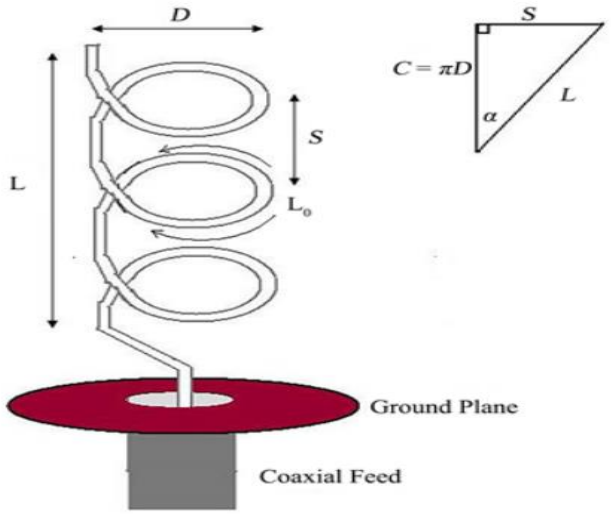


Figure 5. 41 Radiation pattern of helical antenna in axial mode

To find the parameters we use the following equations by selecting N , D, α then substitute in these equations:

1. To find the center wavelength of the range:

$$F_c = \frac{F_1 + F_2}{2} , \lambda = \frac{c}{F_c}$$

After selecting N ,D , α find:

2. Find S: spacing between turns: $S = 0.23\pi D$
3. Find C: Coil circumference: $C = \pi D$
4. To check also: $\alpha = \tan^{-1}\frac{s}{c}$
5. Then H: height of helix: $H = S * N$

The ground plate's size and shape significantly influence directive gain, requiring a minimum diameter of 0.75λ. Both the number of turns and the spacing between the helix and the ground plate can affect the antenna's gain and return loss.

5.9.4.2 Ground plane Design

The ground plane is an essential part of the helical antenna, as it reflects the radiated waves and helps form a directional beam, improving the antenna's overall gain and performance. In this project, a parabolic ground plane was chosen instead of the more common flat circular plate. The parabolic shape offers enhanced directivity and focuses the radiated energy more efficiently in the forward direction, which is highly beneficial for jamming applications where strong, focused signals are needed. The use of a parabolic reflector also helps reduce back radiation and improves the antenna's front-to-back ratio

To draw parabolic shape, we use general equation of curve :

$$w(t) = At^2$$

is a parabolic equation, representing a quadratic relationship between the variable t and the output W(t). This type of equation is typical of a parabola that opens upward (if A>0) or downward (if A<0).
In this form, W(t) denotes the vertical displacement (or height) of the curve at a horizontal distance t from the center, while A is a constant that controls the curvature or steepness of the parabola.

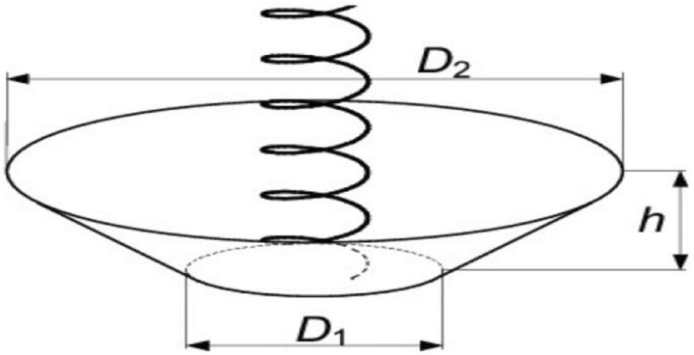


Figure 5. 42 Helix with parabolic ground plane

5.9.5 Simulation and Optimization

To ensure the helical antenna met the required performance specifications, detailed simulation and optimization were conducted using CST Studio Suite, a powerful electromagnetic simulation tool. CST was used to model the antenna structure and analyze key parameters such as return loss (S11), gain, radiation pattern, and axial ratio over the target frequency band of 1.1 GHz to 1.7 GHz. The simulation allowed precise tuning of critical design elements, including the helix diameter, number of turns, pitch angle, and the dimensions and shape of the ground plane. During the optimization process, each parameter

was varied systematically to study its impact on the antenna's overall performance. Particular attention was given to achieving stable circular polarization and maximizing gain. The parabolic ground plane was also modeled in CST, and its effect on beam direction and front-to-back ratio was evaluated in detail. The simulation results guided the refinement of the antenna design, ensuring it delivered directional radiation with high efficiency, making it well-suited for jamming drone communication signals. The successful optimization using CST confirmed the reliability of the final antenna configuration for practical implementation in the system.

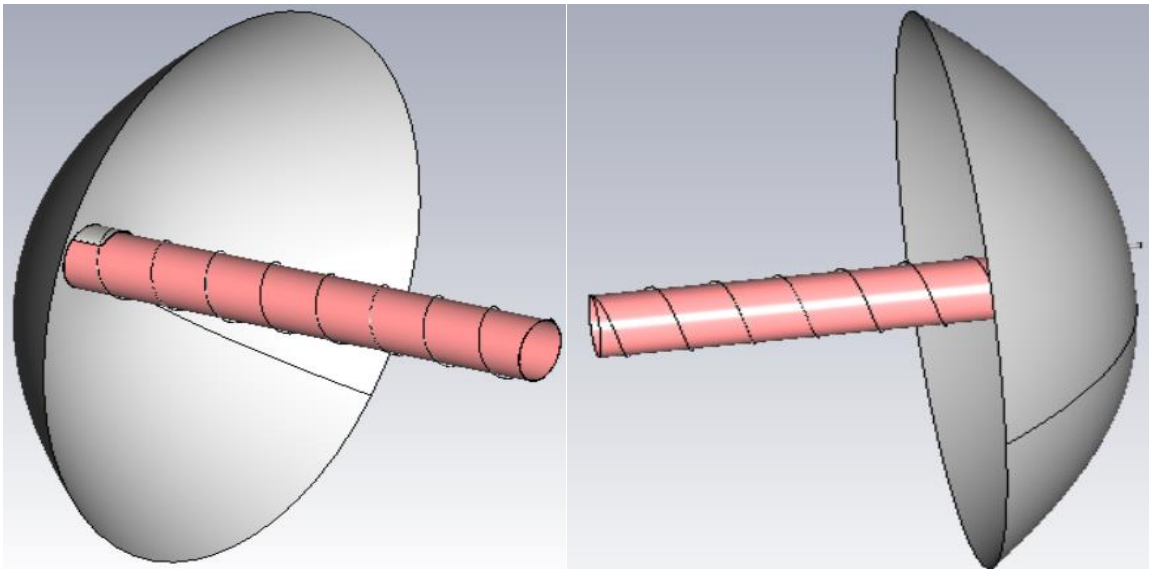


Figure 5. 43 simulation of helical antenna using CST

Parameter List

Name	Expression	Value	Description
Linear_Spiral_cst_torrus_ra	= R	27.0563403256222	Coil's major radius
Linear_Spiral_cst_torrus_ri	= R	27.0563403256222	Coil's minor radius
Linear_Spiral_cst_torrus_phi	= pA	13	Coil's segmentation angle
R	= P/2/pi	27.0563403256222	Helix Radius
Linear_Spiral_cst_torrus_N	= N-1	8	Coil's Nr of turns
H	= N*P*tan(pA*pi/180)	353.228332422112	Helix Height
Linear_Spiral_cst_torrus_h	= 346.30228668834502	346.30228668834502	Coil's height
Rp	= 200	200	
P	= 170	170	Perimeter of the Helix
RG	= 150	150	Ground Disc Radius
sA	= 50	50	
LF	= 15	15	Length of Feed Line
pA	= 13	13	Pitch Angle
LA	= 12	12	Length of Stup Above First Turn
LB	= 11	11	Length of Stub Below First Turn
LC	= 10	10	SMA Length
LS	= 10	10	Stup Height
N	= 9	9	Number of Turns
Ro	= 4.3/2	2.15	Outer Conductor Radius of SMA
D	= 2*Ri	0.9	Wire Diameter
Er_Filler	= 2	2	Dielectric Constant of the SMA
Er_Radom	= 1.5	1.5	Radom Dielectric Constant
Ri	= 0.9/2	0.45	Inner Conductor Radius of the SMA
TR	= 0.5	0.5	Radom Tube Thickness

Figure 5. 44 Simulation parameters list

5.9.5.1 Simulation Results

1. S11: S11, also known as the reflection coefficient, is a key parameter in antenna design that indicates how much power is reflected from the antenna input port. It is expressed in decibels (dB) and represents the ratio of the reflected power to the incident power. In simpler terms, S11 shows how well the antenna is matched to the transmission line or source. A lower S11 value (more negative) means better matching and less power reflected, which translates to more efficient radiation. For example, an S11 of –10 dB means that only 10% of the power is reflected and 90% is transmitted into the antenna. In practical antenna design, an S11 value of –10 dB or lower across the operating frequency range is typically considered acceptable.

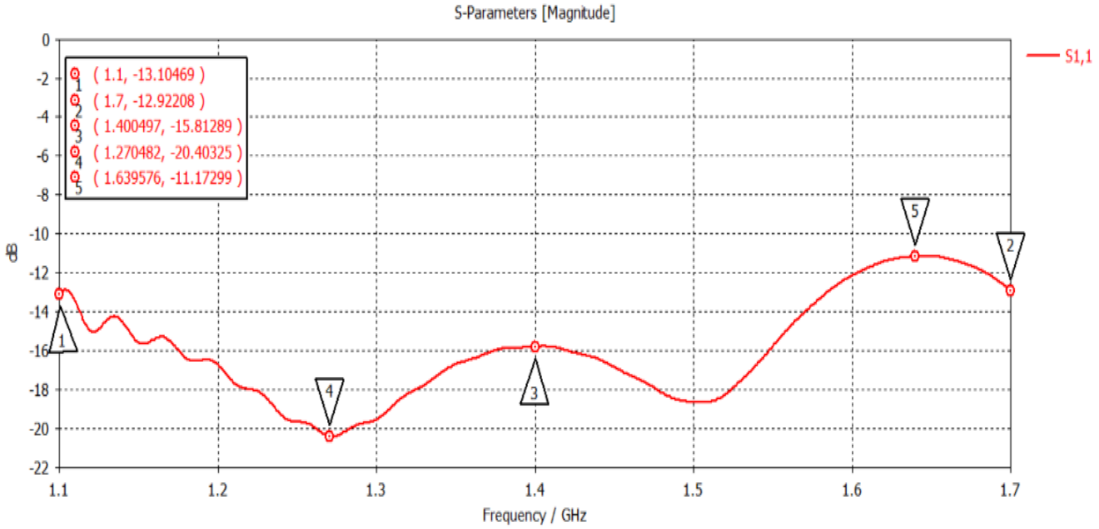


Figure 5. 45 S11 graph

Our design s11 lies in range from -20.4 to -11.17 dB, which is practical due to the reflected power less than 10%.

2. Voltage Standing Wave Ratio (VSWR) is a measure of how efficiently radio-frequency power is transmitted from a power source, through a transmission line, into a load—such as an antenna. It indicates the degree of impedance matching between the antenna and the transmission line. A perfect match results in all the power being transmitted, while a mismatch causes some power to be reflected. VSWR is calculated from the reflection coefficient (S11) and is always a positive number greater than or equal to 1. A VSWR of 1:1 represents an ideal match, meaning no power is reflected.
   In practical antenna design, a VSWR of 2:1 or lower is generally considered acceptable, corresponding to an S11 value of –9.5 dB or better. Lower VSWR values indicate better performance and more efficient radiation from the antenna.

   We achieved VSWR in range from 1.38 to 1.58.

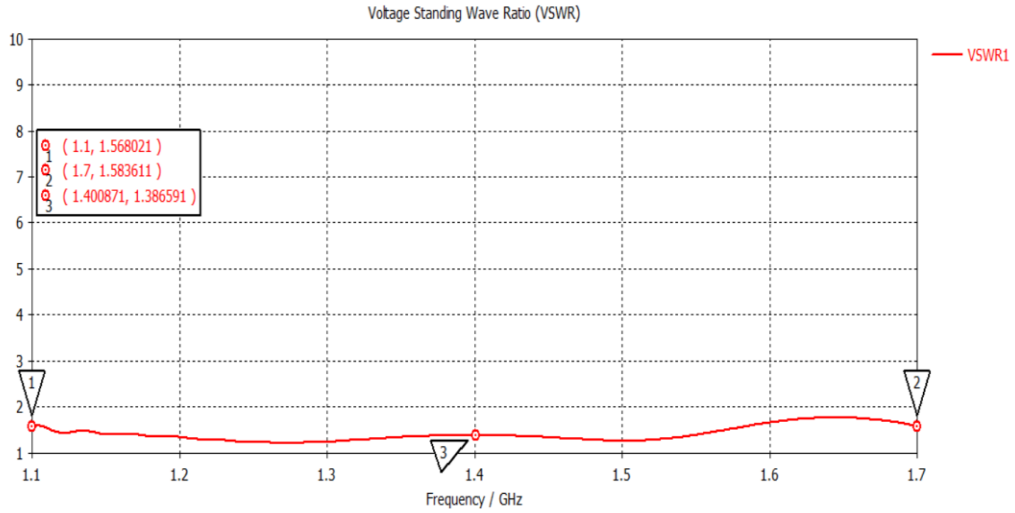


Figure 5. 46 VSWR graph

3. Gain: Realized Gain is a critical antenna performance parameter that represents the actual gain of the antenna while accounting for all losses, including mismatch losses due to impedance differences between the antenna and the transmission line. It provides a more realistic measure of how much power the antenna effectively radiates in a specific direction, considering both its efficiency and the power lost due to reflections (related to S11), realized gain combines both factors and incorporates return loss. It is typically expressed in decibels (dBi) and is often used in practical system-level analysis because it reflects the true radiated power that contributes to communication or jamming effectiveness. dBi stands for decibels relative to an isotropic radiator. It is a unit used to express an antenna's gain; specifically, how much power the antenna radiates in a particular direction compared to an ideal isotropic antenna that radiates equally in all directions. An antenna with a gain of 0 dBi radiates the same amount of power in all directions (like the isotropic reference). A positive dBi value means the antenna focuses more power on a specific direction, which is beneficial for directional applications such as jamming or point-to-point communication. For example, a gain of **10 dBi** means the antenna radiates **10 decibels more** power in its main direction than an isotropic antenna would. This makes dBi a standard and widely used metric for evaluating the effectiveness of antenna designs.
   In simulations, realized gain is especially important for evaluating antenna performance under real-world conditions, and it is often used when designing antennas for applications like drone jamming, where effective power delivery in a specific direction is critical.

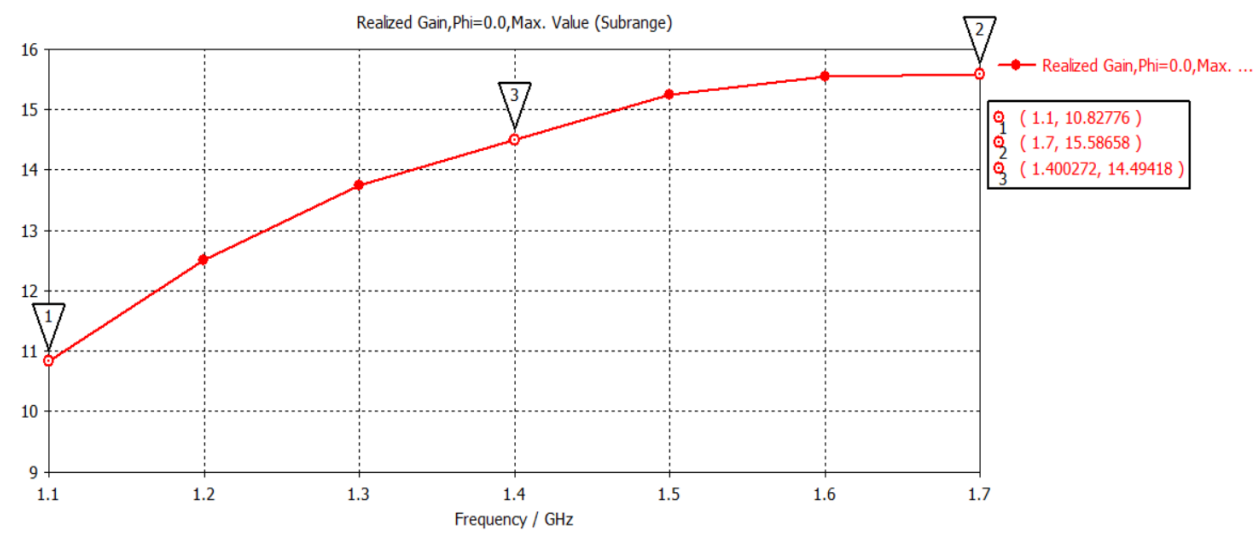


Figure 5. 47 Gain graph

our antenna`s gain in range from 10.827 to 15.586 dB.

4. Far-field: The far-field region is the area far enough from the antenna where electromagnetic waves travel in straight lines and behave like plane waves. In this region, the antenna's radiation becomes stable and predictable, making it the ideal zone to measure key performance characteristics such as gain, radiation pattern, and polarization.
   In simulation software like CST Studio Suite, far-field results are visualized using 1D, 2D, or 3D radiation pattern graphs. These graphs show how the antenna radiates power in different directions in space:
   - **1**D plots typically represent the radiation pattern along a single axis or direction. These are useful for examining the intensity of radiation in a single plane or direction and can help identify main lobe characteristics, null points, and the direction of maximum radiation.

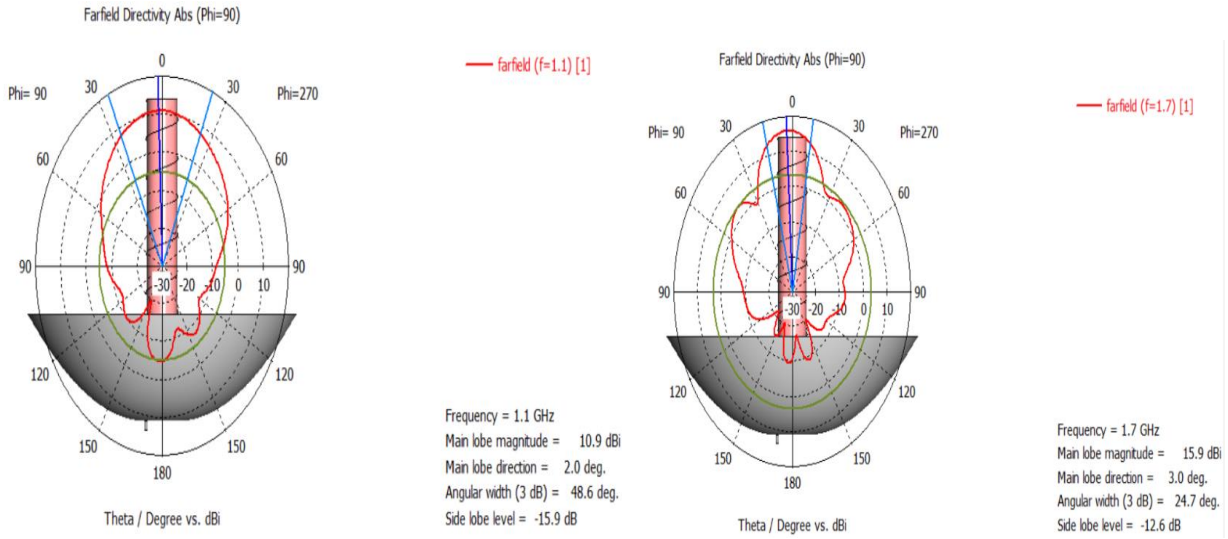


Figure 5. 48 1D Far-field at (1.1 , 1.7 GHZ)

- 2D polar plots display power distribution in a specific plane (e.g., horizontal or vertical).
- 3D radiation patterns provide a complete spatial view of how energy is radiated.

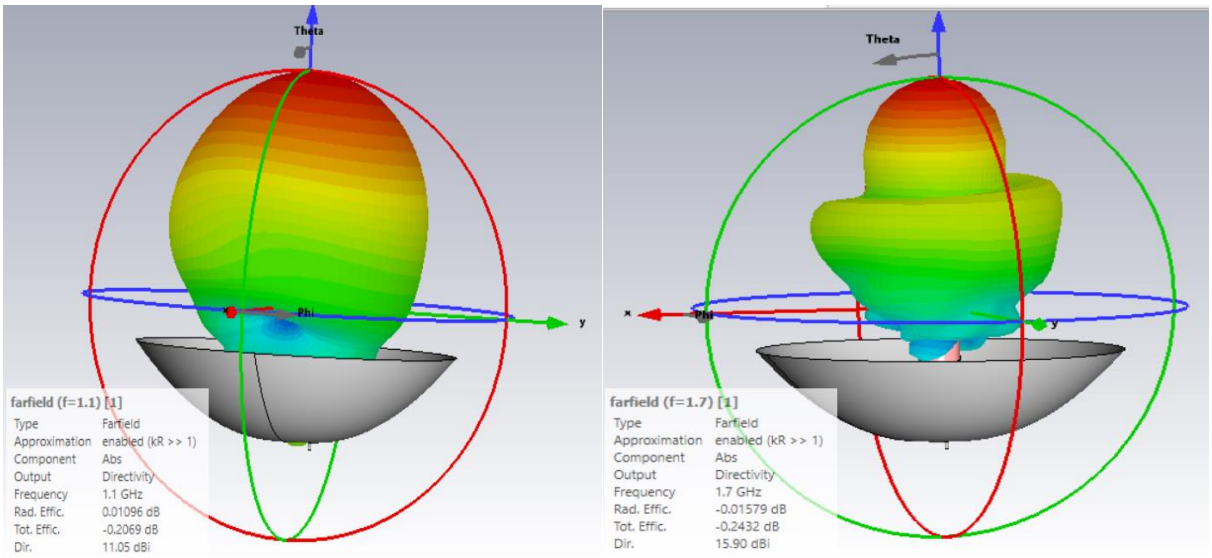


Figure 5. 49 3D Far-field at (1.1 , 1.7 GHZ)

From these plots, we can observe important properties such as the main lobe direction, beamwidth, and side lobe levels. These characteristics help evaluate how focused the antenna's radiation is, which is especially important in jamming applications where the goal is to direct energy toward a specific target—like a drone communication link.

5.9.6 Fabricated Antenna

Figure 5. 50 Fabricated Antenna

5.9.1 The used materials

There are 3 main materials used in the fabricated process:

- Coil of helix : copper
- Ground plane : Aluminum
- Radom (Di-electric shell) :PLA (Polylactic Acid)

  The PLA radome is often assigned a relative permittivity (εr) of 1.5 because 3D-printed PLA contains air gaps that lower its effective permittivity. While solid PLA has a higher εr (around 2.5–3), the printed structure's mix of plastic and air brings it closer to 1.5. This value gives more accurate results in simulations and helps ensure good antenna performance.
- Used SMA connector(SubMiniature version A)
  An SMA connector is used in helical antennas because it provides a reliable, high-frequency connection between the antenna and the rest of the RF system. Here's why it's commonly chosen:

1. High-Frequency Performance: SMA connectors work well up to several GHz, making them ideal for helical antennas used in wireless, GPS, and drone systems.
2. Low Loss: They have low signal loss, which is important to maintain antenna efficiency.
3. Compact Size: SMA connectors are small, saving space in compact antenna setups.
4. Durability: They are mechanically strong and offer repeatable performance.
5. Easy Integration: Widely available and easy to solder or mount on PCBs or antenna bases.

Chapter 6

Results

6.1 synthesizer

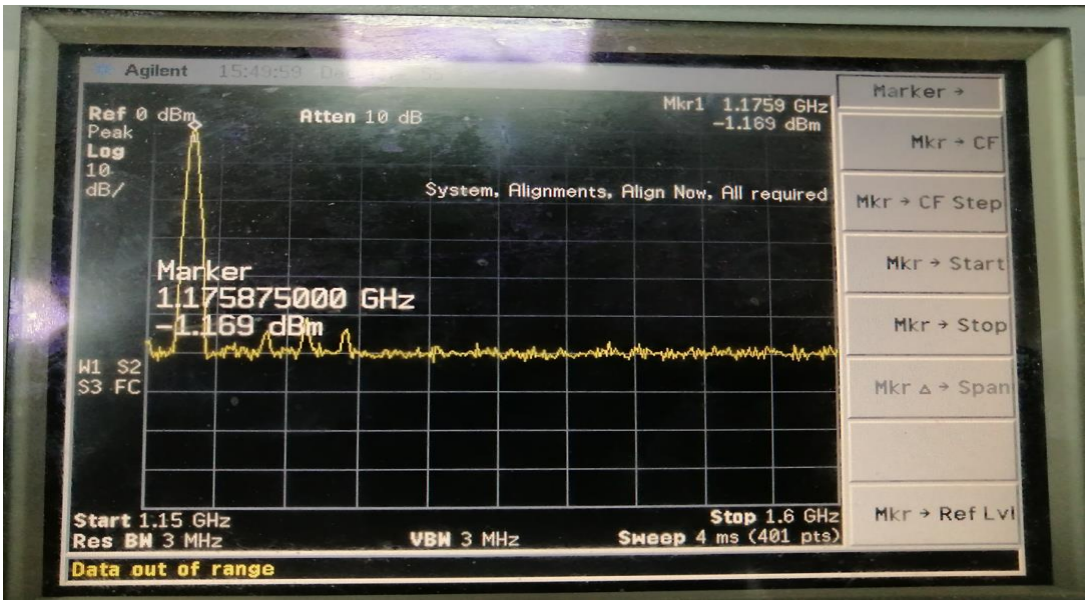


Figure 6. 1 CW signal at 1176.45 MHz

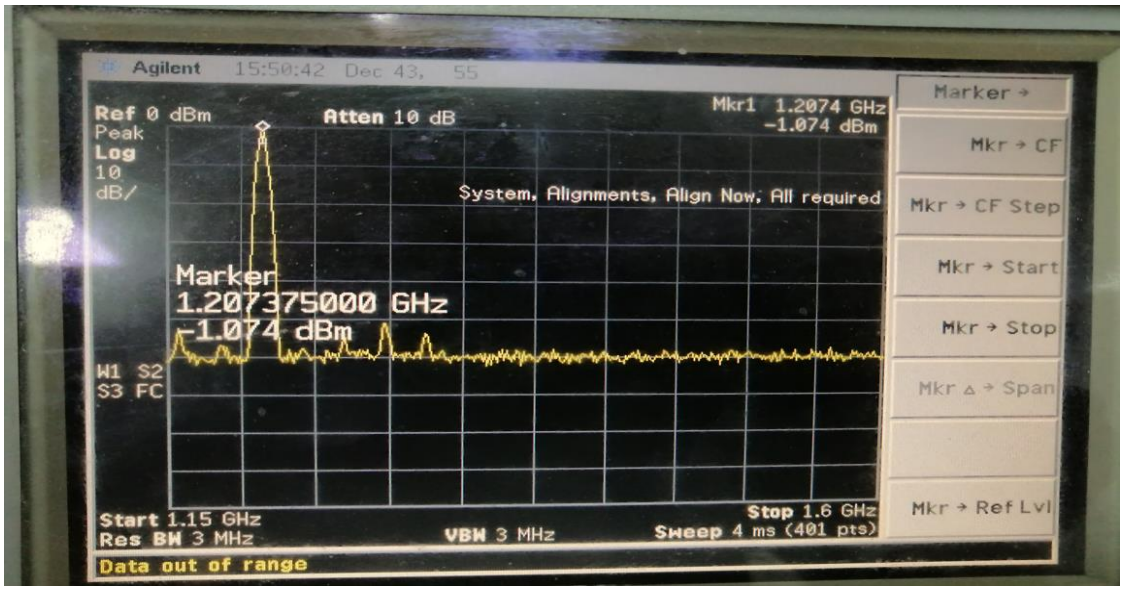


Figure 6. 2 CW signal at 1207.14 MHz

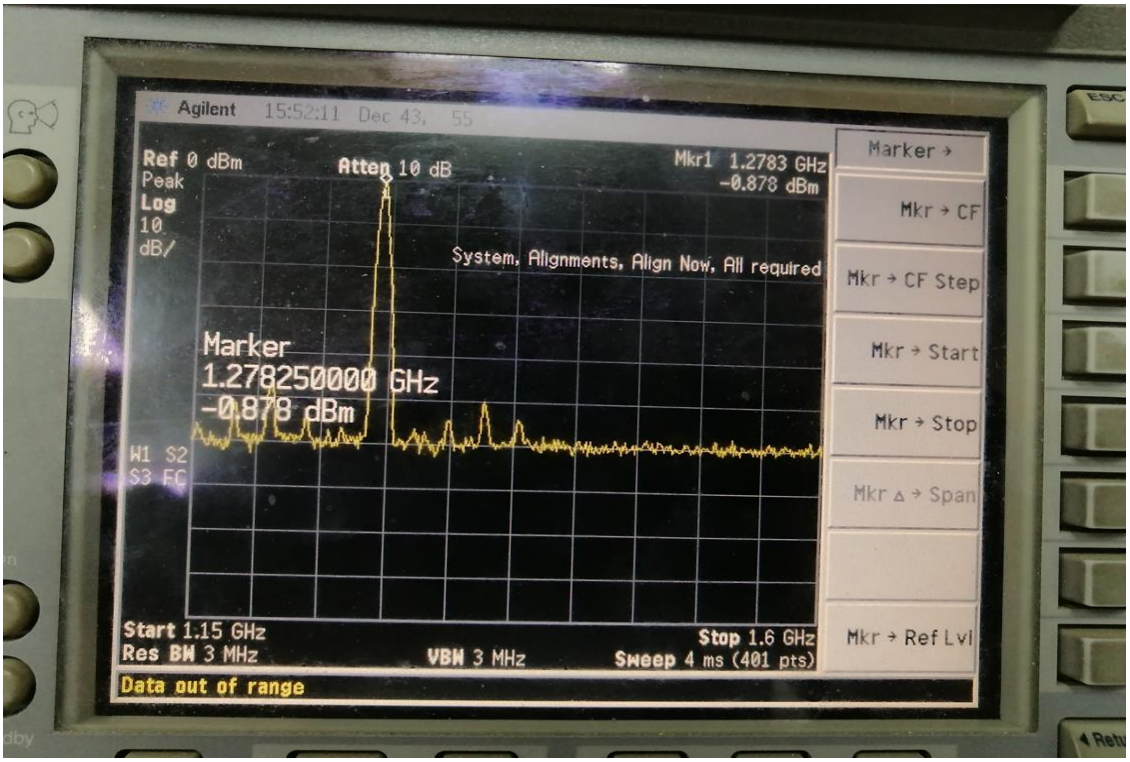


Figure 6. 3 CW signal at 1227.6 MHz

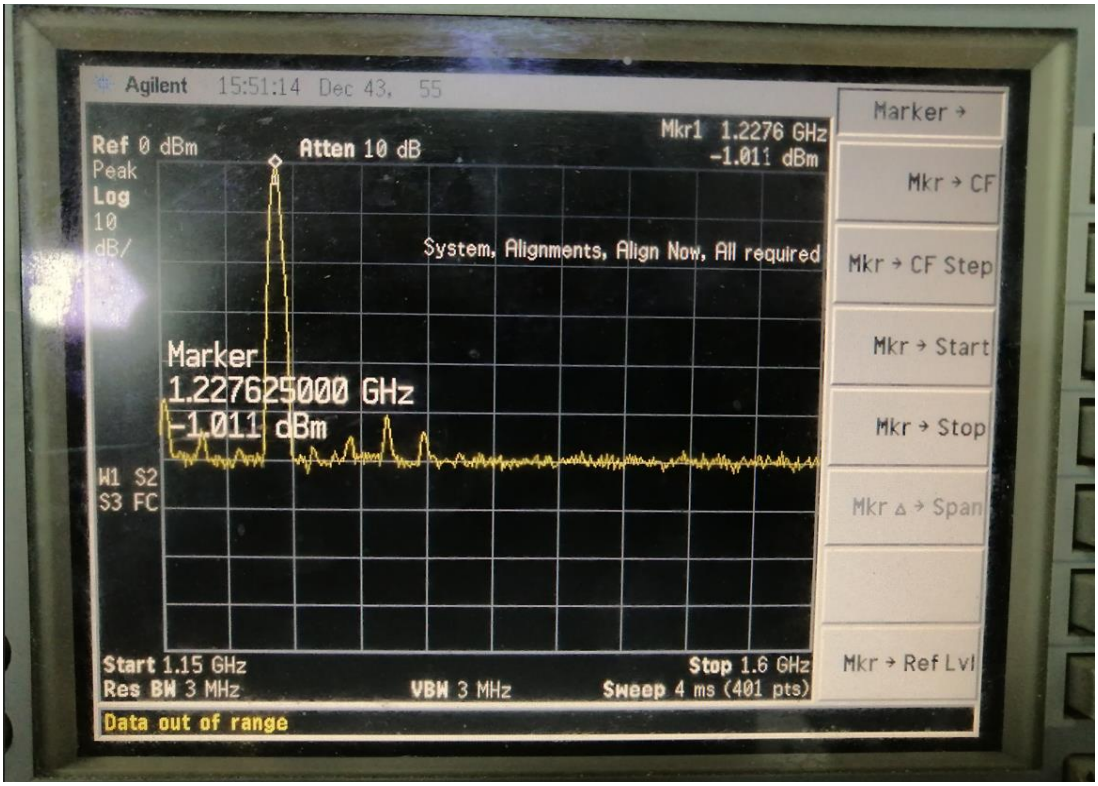


Figure 6. 4 CW signal at 1278.75 MHz

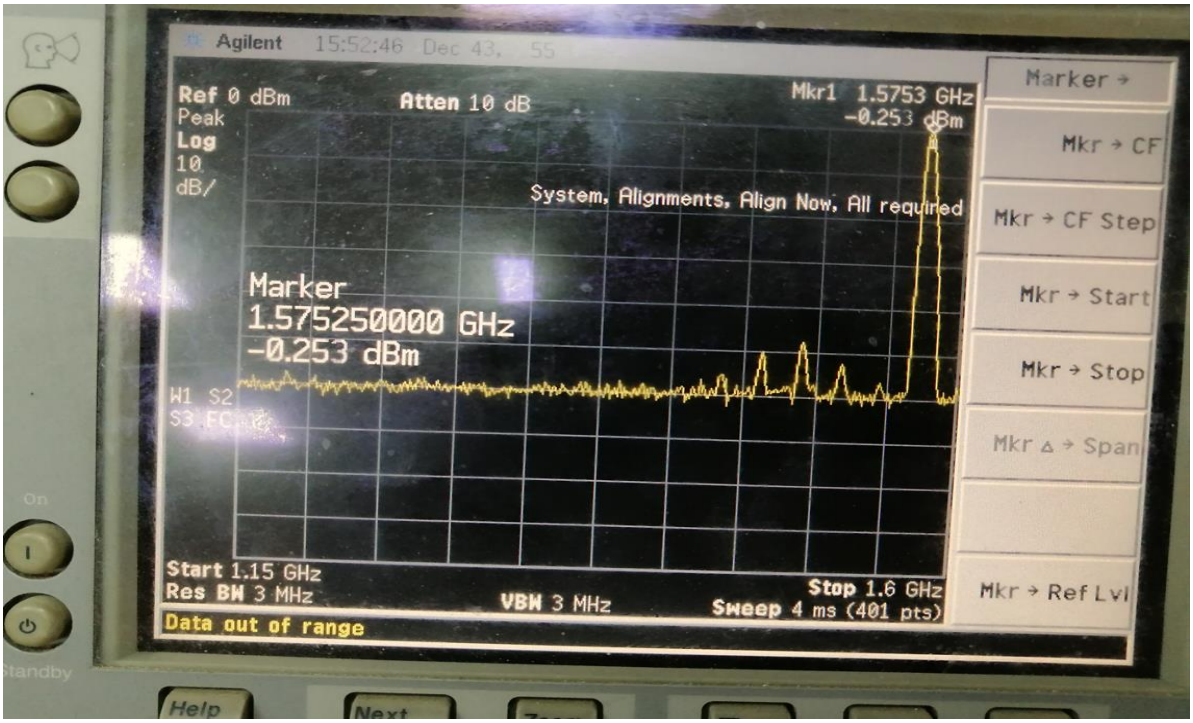


Figure 6. 5 CW signal at 1575.42 MHz

Comparing result before and after applying different jamming techniques.

1. GNSS statues before applying jamming techniques

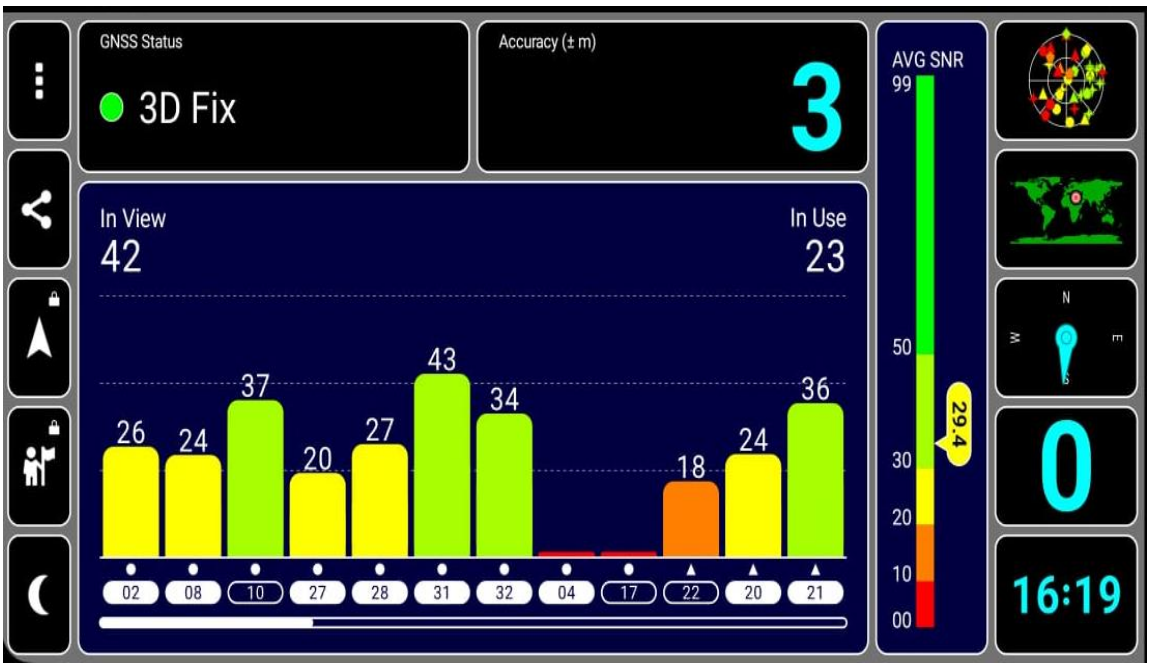


Figure 6. 6 GNSS statues

2. Tone jamming (CW)

Figure 6. 7 CW @ 1575 MHz

3. Sweep jamming

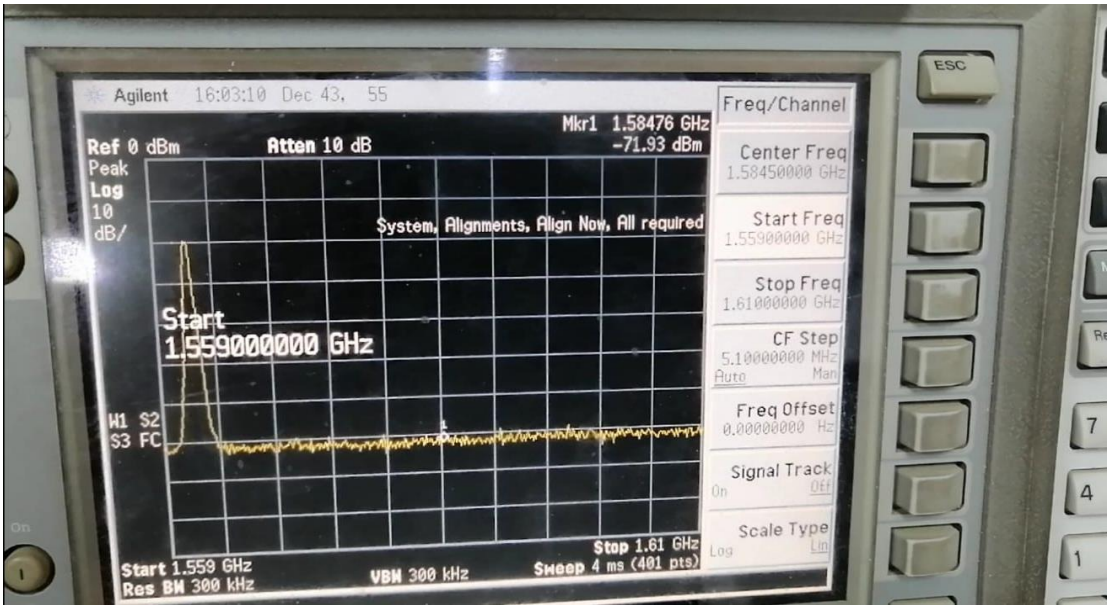


Figure 6. 8 Sweep Start

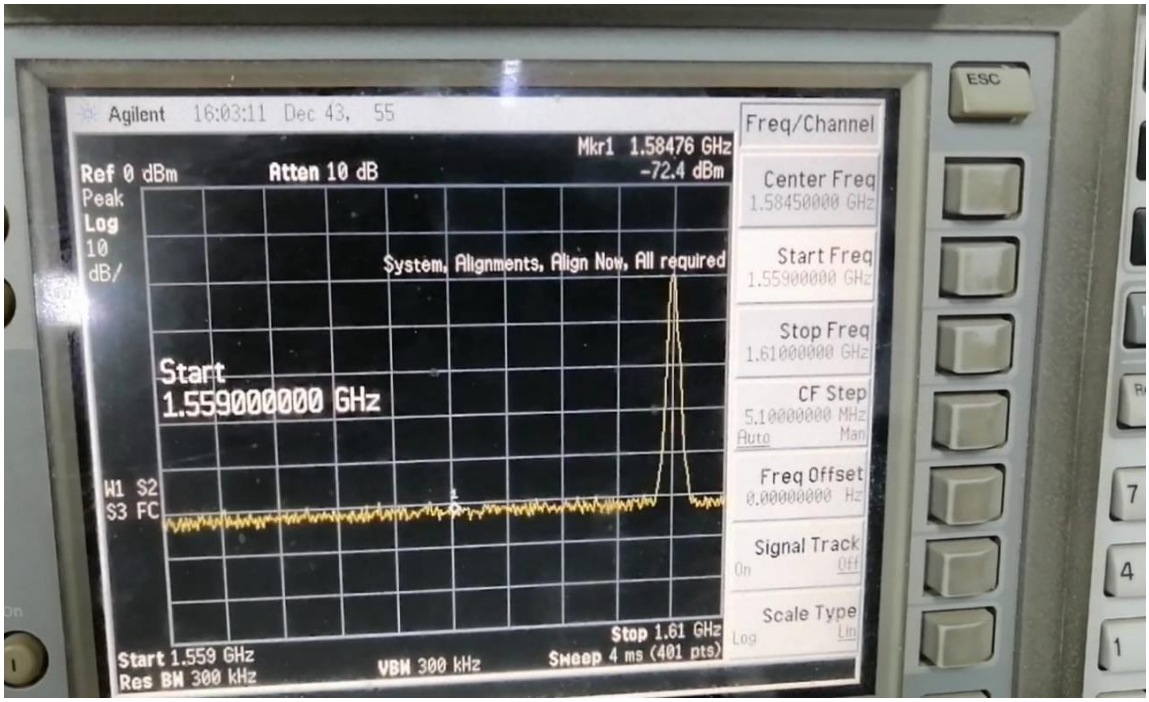


Figure 6. 9 Sweep End

4. Pulse jamming

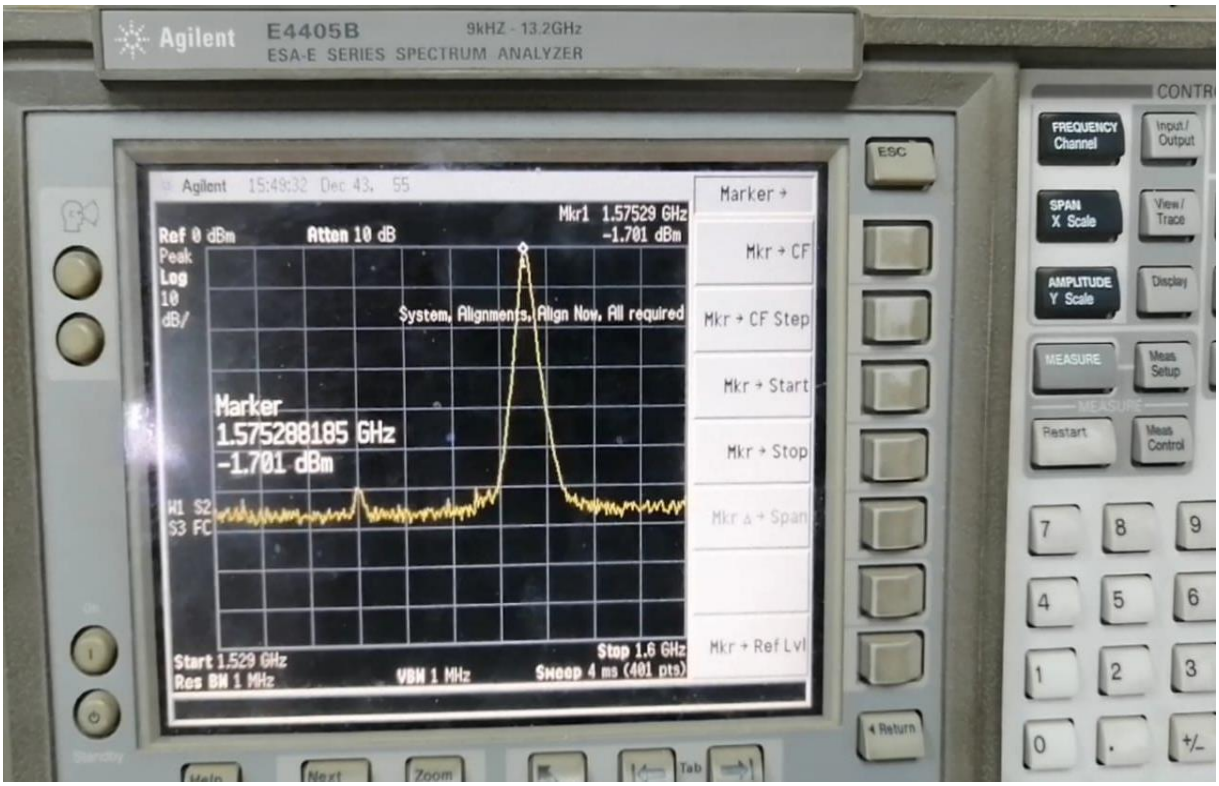


Figure 6. 10 Pulse is On

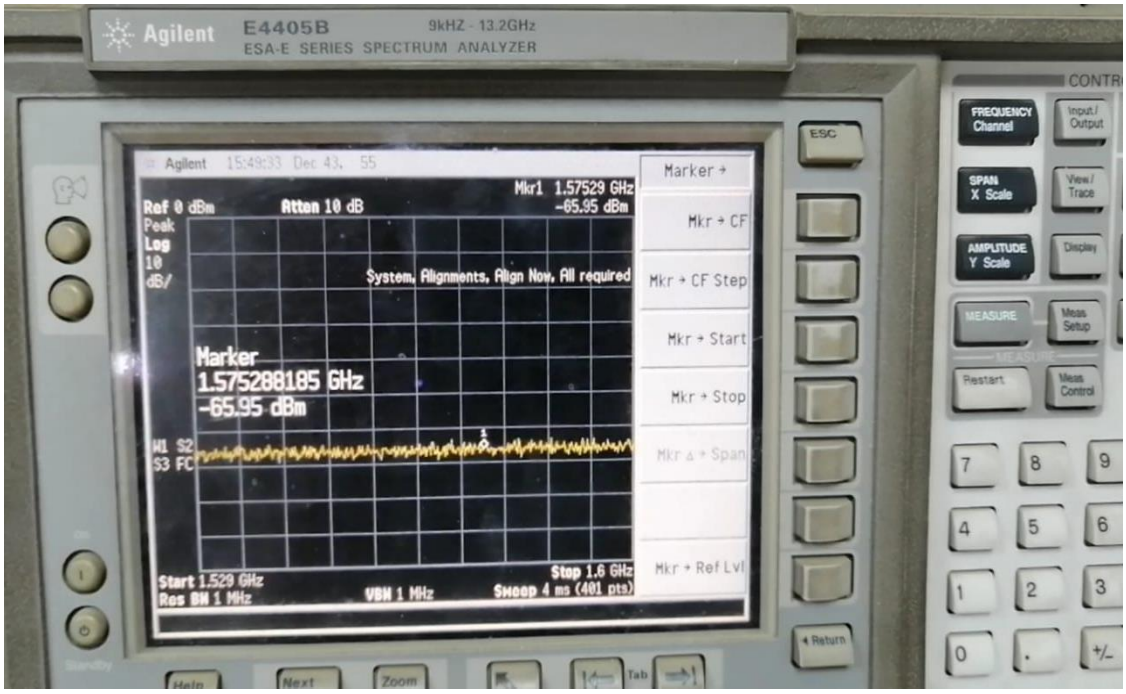


Figure 6. 11 Pulse is OFF

6.1.1 GPS Test Mobile Application

This app displays information from your device's GPS receiver and other sensors. It shows the position and receives signal strength of each satellite within view, as well as your current location and time.

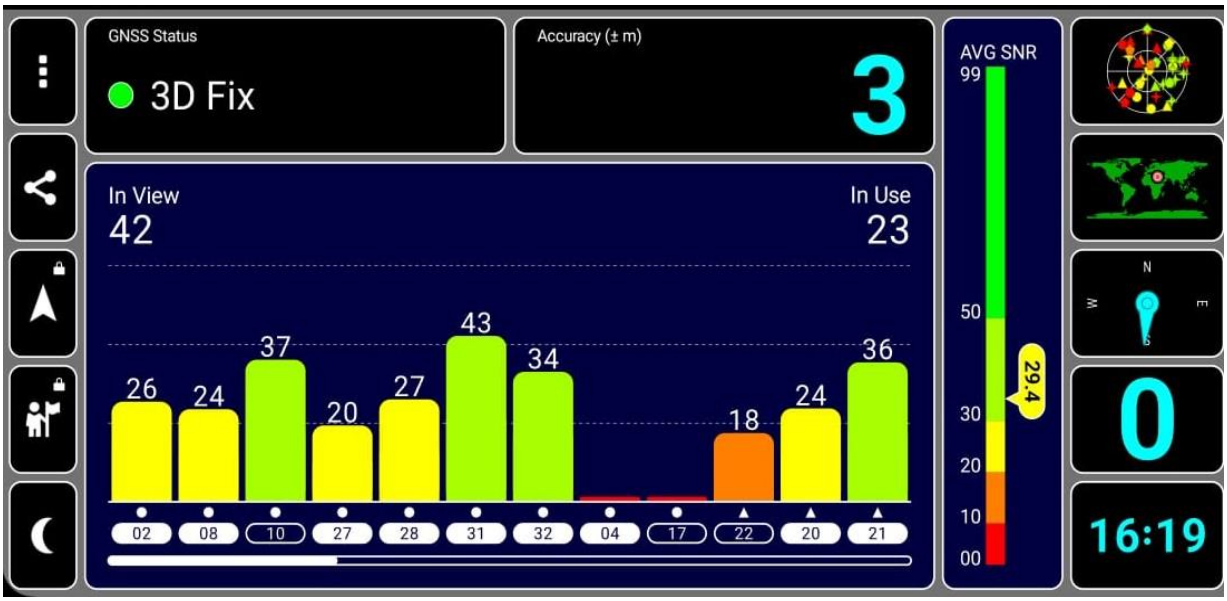


Figure 6. 12 Main Application Interface

This view displays satellite signal information. The GPS Status field shows the state of the GPS receiver. You can tap on this field to display a help dialog, showing the current state of the GPS signal. This dialog has a location services button to turn the GPS on or off. GPS Status can have one of the following values: -

- GPS Error with GPS could be permissions.
- GPS None Your device has no GPS.
- GPS Off the GPS is turned off.
- No Fix the GPS has no location fix.
- 2D Fix the GPS has a partial location fix.
- 3D Fix the GPS has a full location fix.

The Accuracy field shows the accuracy of the GPS location, the smaller the number the better.

Below this field is the satellite signal graph, it shows a bar for each satellite in view. The number beneath the bar is the satellite number (if the number colors are inverted the satellite is used in calculating the positional fix) and the number above it is the signal to noise ratio in decibels. you can horizontally scroll the graph to show more satellites. In the top corners of the graph are two fields showing how many satellites the GPS can see and how many are being used to calculate your position.

Below the satellite signal graph is a legend for the SNR color. The colors represent signal to noise ratio ranges. The satellite signals can have an SNR value between 0 and 99dB, 0 is worse. Normally the SNR will never get above 50 and anything above 40 is good.

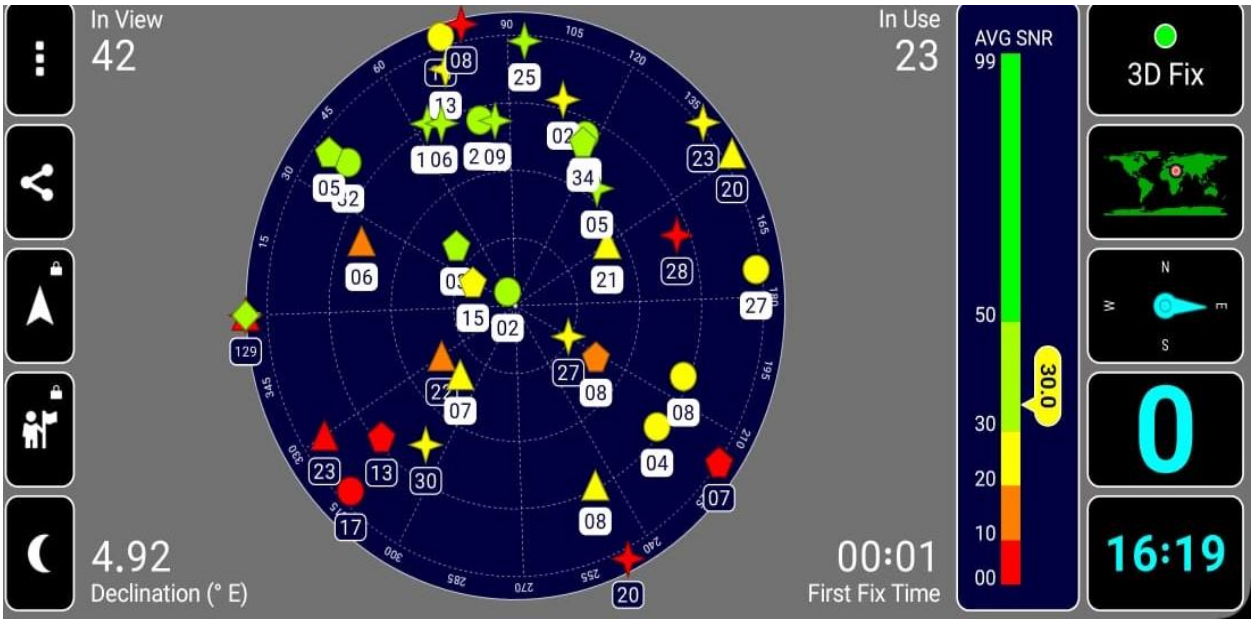


Figure 6. 13 SKY view

This view displays satellite positions. The large circular field on this screen shows a view of the sky overhead (sky view), the outside of the circle represents the horizon, and the center represents the sky directly above your head. The sky view rotates to the correct position using the internal compass. The satellites in view of the GPS receiver are drawn on the sky view if the symbol is round the satellite is a GPS satellite and if the symbol is a triangle the satellite is a GLONASS satellite. Each satellite symbol has the satellite number shown below it and is drawn in a color that represents the satellite signal to noise ratio.

The signal to noise ratios and corresponding color ranges are shown on the SNR legend, below the sky view.

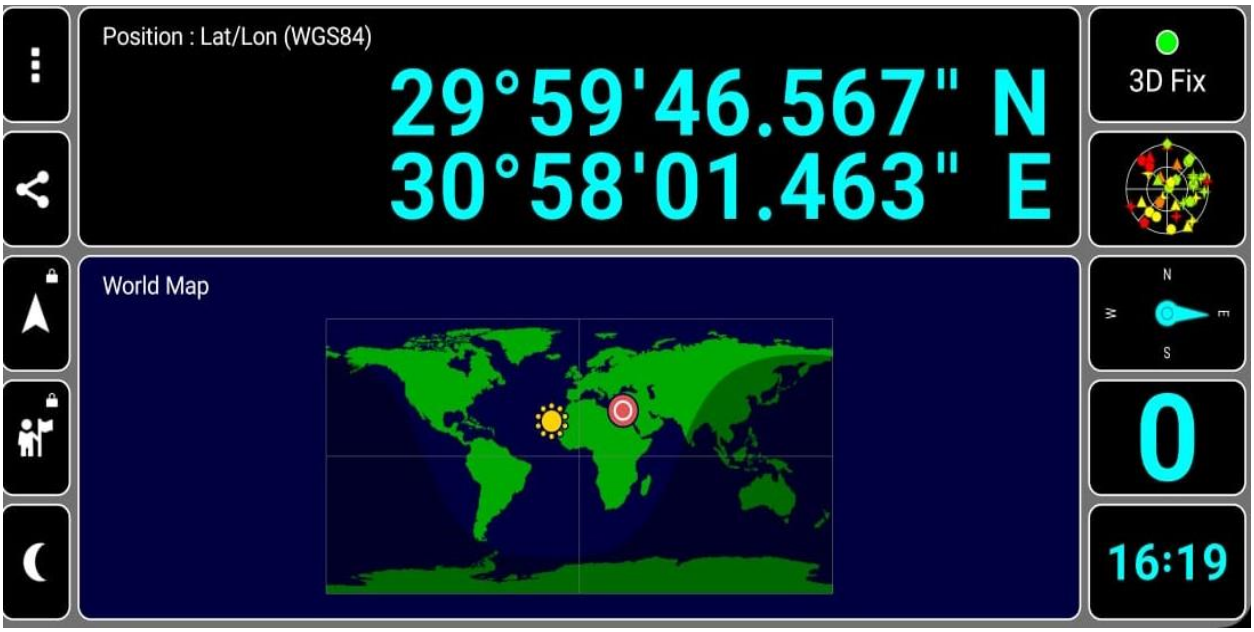


Figure 6. 14 Latitude and longitude and map

This view displays your position on the earth.

The top field shows the coordinates of your current location in the selected coordinate grid. The bottom field is a map of the world, on which your current position is shown by a red marker.

The transition between day and night and the location of the sun relative to your current location is also drawn on the map.

6.2 Antenna

6.2.1 Measurement process

Figure 6. 15 S11 measurement process

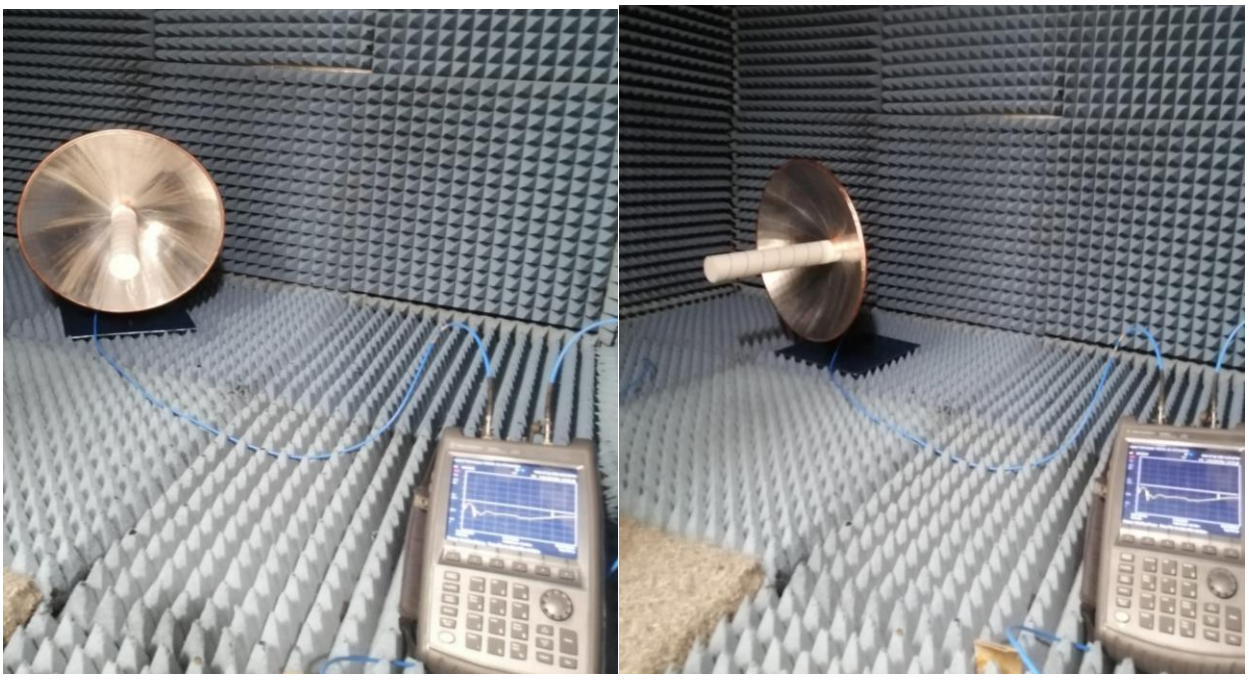


Figure 6. 16 Radiation pattern process

6.2.2 Measurement vs Simulation

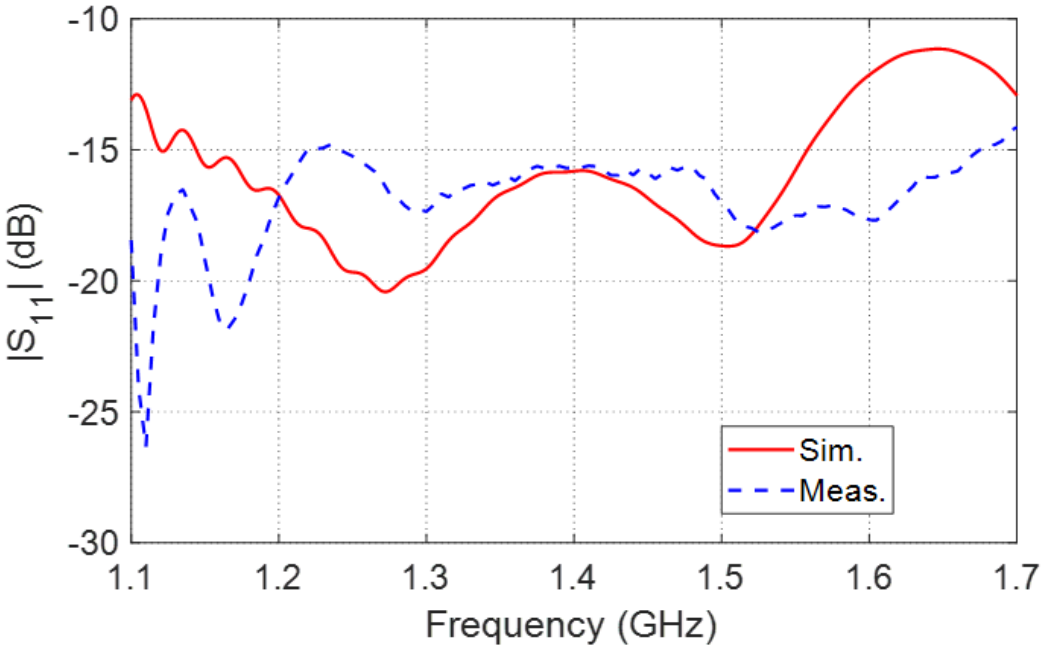


Figure 6. 17 S11

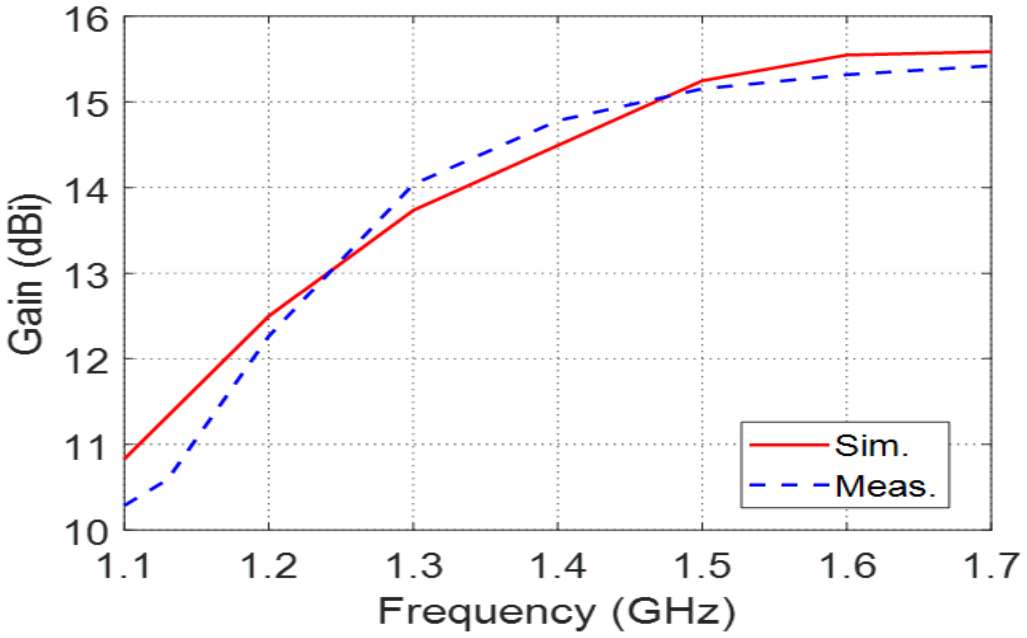
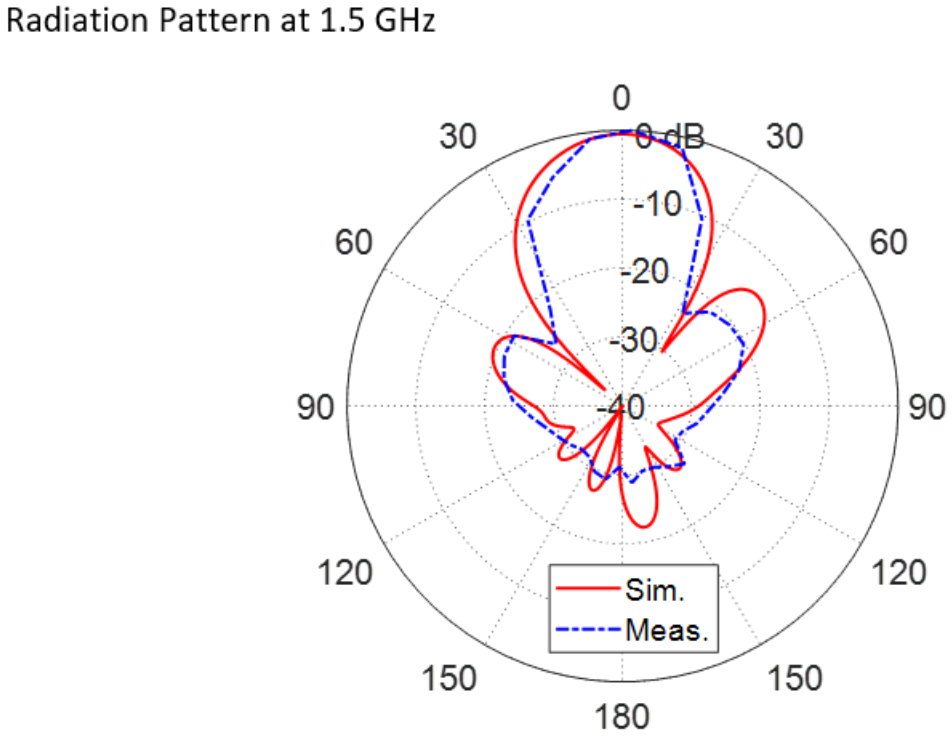
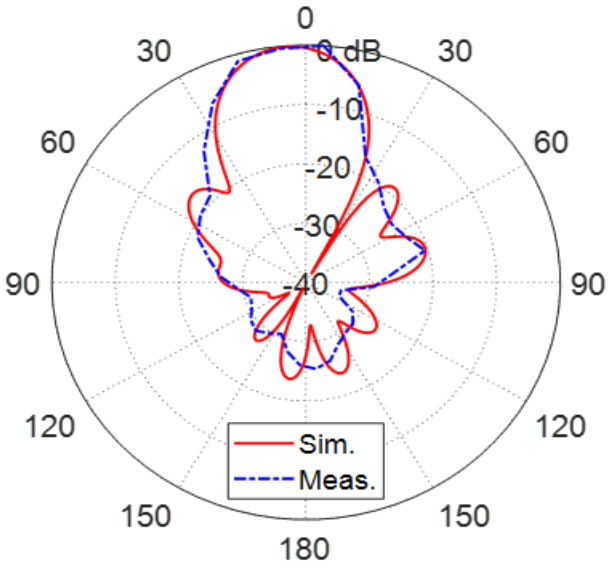


Figure 6. 18 Gain



$\phi = 0^o$

Figure 6. 19 Fairfield @ Phase = 0° (E-plane pattern)



$\phi = 90^o$

Figure 6. 20 Farfield @ Phase = 90° (H-plane pattern)

References

1. Methods for Drone Detection and Jamming. 2020

https://www.eventiotic.com/eventiotic/files/Papers/URL/f07e8f39-5c16-420e-b0e3-5eb5b5ab1ba0.pdf

2. Quantitative Analysis of Deep Learning-Based Object Detection Models 15 May 2024

Quantitative Analysis of Deep Learning-Based Object Detection Models

3. Multiple Real-time object identification using SSD. Multi-Box detection. 23 Feb 2019 https://ieeexplore.ieee.org/document/8862041

4. Drone Detection in Long-range Surveillance Videos 25 Nov 2019

Drone Detection in Long-range Surveillance Videos

5. An Improved SSD Object Detection Algorithm 3 Feb 2020

https://ieeexplore.ieee.org/abstract/document/8978787

6. A Comparative Survey on State-of-the-Art Detectors R-CNN, YOLO, and SSD Book 3 Apr 2021

https://link.springer.com/chapter/10.1007/978-981-33-4443-3_46

7. Drone vs. Bird Detection: Deep Learning Algorithms and Results from a Grand Challenge 16 Apr 2021

https://www.mdpi.com/1424-8220/21/8/2824

8. Fast Drone Detection with Modeling Algorithms 5 Aug 2024

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10623452

9. OSSDY: embedded system-based object surveillance detection system with small drone using deep YOLO 17 May 2021

https://jivp-eurasipjournals.springeropen.com/articles/10.1186/s13640-021-00559-1

10. DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor 21 May 2023

DC-YOLOv8: Small-Size Object Detection Based on Camera Sensor

11. Bird and UAVs Recognition Detection and Tracking Based on Improved YOLOv9-DeepSORT 7 Oct 2024

https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10706910

12. Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano 19 Nov 2024

Edge Computing-Driven Real-Time Drone Detection

13. Lightweight Anti-UAV Detection: Improved YOLOv11 25 Dec 2024

https://www.mdpi.com/2504-446X/9/1/11

14. Deep learning-based strategies for the detection and tracking of drones using several cameras. 24 Jul 2019

https://ipsjcva.springeropen.com/articles/10.1186/s41074-019-0059-x

15. Drone Surveillance Using Detection, Tracking and Classification Techniques 4 Aug 2022

Drone Surveillance Using Detection, Tracking and Classification Techniques

16. Detection and tracking of dim objects in Infrared (IR) Images using SVM 21 May 2016

https://ieeexplore.ieee.org/document/7535265

17. Drone Detection and Tracking in Real-Time by Fusion of Different Sensing Modalities 26 Oct 2022

https://www.mdpi.com/2504-446X/6/11/317

18. Field Test Validations of Vision-based Multi-camera Multi-drone Tracking and 3D Localizing with Concurrent Camera 26 May 2021

https://ieeexplore.ieee.org/document/9435654

19. Towards Reliable Identification and Tracking of Drones Within a Swarm 5 June 2024

https://link.springer.com/article/10.1007/s10846-024-02115-1

20. Design of anti-GPS for reasons of security January 2009

https://www.researchgate.net/publication/228961312

21 A study of GPS jamming and anti-jamming 19-20 December 2009

https://ieeexplore.ieee.org/document/5406988

22. GPS signal jamming and anti-jamming strategy — A theoretical analysis 16-18 December 2016

https://ieeexplore.ieee.org/document/7838933

23. Design of GPS anti-jamming algorithm using adaptive array antenna to mitigate the noise and interference 13 February 2019

https://link.springer.com/article/10.1007/s12652-019-01187-4

24. Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms 6 March 2020

https://link.springer.com/article/10.1007/s11277-020-07212-6#Sec18

25. Methods for Drone Detection and Jamming 2020

https://scholar.google.com/scholar?hl=ar&as_sdt=0%2C5&q=Methods+for+Drone+Detection+and+Jamming&btnG

26. A Survey on the Use of Deep Learning Techniques for UAV Jamming and Deception 23 September 2022

https://www.mdpi.com/2079-9292/11/19/3025

27. Overview of Jamming Technology for Satellite Navigation 22 July 2023

https://www.mdpi.com/2075-1702/11/7/768

28. Smart Jamming: Deep Learning-Based UAVs Neutralization System 02-05 September 2024

https://ieeexplore.ieee.org/document/10722367

29. A Survey on the Use of Deep Learning Techniques for UAV Jamming and Deception 23 September 2022

https://www.mdpi.com/2079-9292/11/19/3025

30. Overview of Jamming Technology for Satellite Navigation 22 July 2023

https://www.mdpi.com/2075-1702/11/7/768

31. Smart Jamming: Deep Learning-Based UAVs Neutralization System 02-05 September 2024

https://ieeexplore.ieee.org/document/10722367

32. A Foundational Framework for an UAV GPS Jamming Simulator 21 November 2024

A Foundational Framework for an UAV GPS Jamming Simulator | IEEE Conference Publication | IEEE Xplore

33. UAV Communication and Navigation Signals Jamming Methods 04 December 2024

UAV Communication and Navigation Signals Jamming Methods | IEEE Conference Publication | IEEE Xplore

34. The Effects of Jamming on Global Positioning System (GPS) Accuracy for Unmanned Aerial Vehicles (UAVs) 04 January 2023

The Effects of Jamming on Global Positioning System (GPS) Accuracy for Unmanned Aerial Vehicles (UAVs) | IEEE Conference Publication | IEEE Xplore

35. Jamming of Spread Spectrum Communications Used in UAV Remote Control systems 2017

36. A detection and identification method based on signal power for different types of Electronic Jamming attacks on GPS signals 21 November 2019

37. Drone Detection with Vision Algorithms 5 September 2024

A detection and identification method based on signal power for different types of Electronic Jamming attacks on GPS signals | IEEE Conference Publication | IEEE Xplore

38. The effect of Electronic Jammers on GPS Signals 21-24 March 2019

The effect of Electronic Jammers on GPS Signals | IEEE Conference Publication | IEEE Xplore

39. Tactical Battlefield Communications Electronic Warfare 2009

40. Digital communication jamming by Cem Sen September 2000

41.DESIGN OF AN INNOVATIVE HELICAL ANTENNA FOR AERIAL PLATFORMS TO ENABLE JAM RESISTANT RECEPTION OF GNSS SIGNALS

https://www.researchgate.net/publication/333727081_Design_of_an_Innovative_Helical_Antenna_for_Aerial_Platforms_to_Enable_Jam_Resistant_Reception_of_GNSS_Signals

42. A MODERN APPROACH FOR DESIGN AND OPTIMIZATION OF HELICAL ANTENNA USED AT TRANSMITTING END OF GPS

http://ael.chungbuk.ac.kr/ael/%EC%A1%B8%EC%97%85%EC%9E%91%ED%92%88/%EC%9C%A4%EC%A2%85%ED%9B%88/%EC%B0%B8%EA%B3%A0%EB%AC%B8%ED%97%8C/rao.pdf

43. Design, Construction and Performance Analysis of Helical Antenna Operating at 5.8ghz

https://www.arcjournals.org/pdfs/ijarps/v2-i11/5.pdf

44. Design and Realization Step Frequency Continuous Wave Generator for Ground Penetrating Radar Using Phase-locked Loop October 2019

https://ieeexplore.ieee.org/document/8985506

45. Audio Based Drone Detection and Identification using Deep Learning

https://www.researchgate.net/publication/332727775_Audio_Based_Drone_Detection_and_Identification_using_Deep_Learning

46. Acoustic-Based Drone Detection Using Neural Networks – A Comprehensive Analysis

https://www.researchgate.net/publication/377884609_Acoustic-Based_Drone_Detection_Using_Neural_Networks_-_A_Comprehensive_Analysis

APPENDIX

All references , videos and project materials are attached in flash memory.