DEMOCRATIC AND POPULAR REPUBLIC OF ALGERIA
Ministry of Higher Education and Scientific Research
Mohamed Khider University – BISKRA
Faculty of Exact Sciences, Natural Sciences and Life Sciences

Department of Computer Science

Order N°: ...Startup: . . . /M2/2025

DISSERTATION

Presented to obtain the academic master's degree in

Computer Science

Autonomous Drone System for Palm Tree Pollination

By:

Elbey Mohamed Achraf (RTIC)
Rezig Slimane (IVA)
Belkahla Oussama Abderaouf (IA)

Framed by:

Pr. Kahloul Laid
Dr.Ouaar Hanane

Discussed on 16/06/2025 before the jury composed of:

Djalila Belkebir	...	President
Kahloul Laid	...	Supervisor
Belaala Abir	...	Examiner

Academic year: 2024 / 2025

بسم الله الرحمن الرحيم

Acknowledgements

All praise is due to Allah, Lord of the Worlds, and peace and blessings be upon the most honorable of Messengers.

We, the members of this project team, would like to begin by expressing our deepest gratitude to Allah Almighty, who granted us the strength, patience, and perseverance to complete this work.

We are profoundly thankful to our supervisors, **Prof. Laid Kahloul** and **Dr.Ouaar Hanane**, whose guidance, constructive feedback, and continuous encouragement were instrumental throughout the development of this project. Their expertise and mentorship significantly contributed to the quality and direction of our work.

We also extend our sincere thanks to the esteemed members of the evaluation committee for their time, critical insights, and valuable suggestions.

Our appreciation goes as well to the professors and staff of the **Computer Science Department** for their academic support during our years of study.

Special thanks are due to the **Centre de Recherche en Technologies Industrielles (CRTI)** in Algiers for welcoming us during our field visit and for providing us with meaningful technical insights that enriched the practical aspect of our work.

We also wish to thank our families — parents, brothers, and sisters — for their unwavering love, support, and sacrifices. Without their encouragement, this achievement would not have been possible.

To our friends who shared this academic journey with us — thank you for your presence, your motivation, and the unforgettable moments.

Finally, we express our appreciation to all those who contributed, directly or indirectly, to the success of this project. Your support has been truly invaluable.

Dedication

We dedicate this humble work to our beloved parents, whose unconditional love, countless sacrifices, and unwavering support have been the true foundation of our journey. Their faith in us gave us the strength to persevere, and their values continue to guide our path.

To our brothers and sisters, who have always stood by us with encouragement and inspiration. Their presence brought comfort in difficult times and joy in moments of success.

To our cherished friends, who shared with us the challenges and the victories, offering motivation, loyalty, and unforgettable memories along the way.

And finally, to everyone who believed in us, guided us, or supported us throughout this academic journey this achievement is dedicated to you, with heartfelt gratitude and deep appreciation.

Abstract:

The aim of this project is to develop an autonomous drone system designed to automate the pollination of date palms. This system provides a reliable and intelligent solution for farmers, minimizing manual labor and increasing pollination efficiency. Each drone is equipped with a Raspberry Pi 4B, a Pixracer flight controller, a water pump, sensors, and a Pi Camera, enabling real-time detection of palm trees using the YOLOv12 object detection model. The system architecture adopts a star communication topology, where each drone communicates exclusively with a central ground station, using the UDP protocol for general data exchange and the RTP protocol (over UDP) for real-time video transmission. Internally, MAVLink is used for communication between the Raspberry Pi and the Pixracer. This structure ensures streamlined task distribution and monitoring from a single control point. Through the integration of onboard AI processing and autonomous flight, the system is capable of performing precise pollination tasks in unstructured environments. By combining these technologies with real-time communication, this experimental prototype aims to enhance pollination reliability, reduce resource consumption, and promote sustainable agricultural practices, especially in regions dependent on date production.

Key words: Autonomous drone, smart agriculture, palm tree pollination, YOLOv12, UAV, UDP protocol, RTP, MAVLink, real-time detection, precision farming, IoT in agriculture.

Résumé :

Ce projet a pour objectif de développer un système autonome basé sur des drones afin d'automatiser la pollinisation des palmiers dattiers. Ce système constitue une solution fiable et intelligente pour les agriculteurs, en réduisant la main-d'œuvre manuelle et en augmentant l'efficacité de la pollinisation. Chaque drone est équipé d'un Raspberry Pi 4B, d'un contrôleur de vol Pixracer, d'une pompe à eau, de capteurs, et d'une caméra Pi, permettant la détection en temps réel des palmiers à l'aide du modèle de détection YOLOv12. L'architecture du système adopte une topologie de communication en étoile, dans laquelle chaque drone communique exclusivement avec une station au sol centrale, en utilisant le protocole UDP pour l'échange général de données et le protocole RTP (basé sur UDP) pour la transmission vidéo en temps réel. En interne, le protocole MAVLink est utilisé pour assurer la communication entre le Raspberry Pi et le Pixracer. Cette architecture permet une répartition fluide des tâches et une surveillance centralisée à partir d'un seul point de contrôle. Grâce à l'intégration d'un traitement IA embarqué et du vol autonome, le système est capable d'effectuer des opérations de pollinisation précises dans des environnements non structurés. En combinant ces technologies avec une communication en temps réel, ce prototype expérimental vise à améliorer la fiabilité des processus de pollinisation, à réduire la consommation de ressources et à promouvoir des pratiques agricoles durables, notamment dans les régions dépendantes de la production de dattes.

Mots-clés : Drone autonome, agriculture intelligente, pollinisation du palmier dattier, YOLOv12, UAV, protocole UDP, RTP, MAVLink, détection en temps réel, agriculture de précision, IoT agricole.

الملخص:

يهدف هذا المشروع إلى تطوير نظام مستقل يعتمد على طائرات بدون طيار لأتمتة عملية تلقيح أشجار النخيل. يوفّر هذا النظام حلاً موثوقًا وذكيًا للمزارعين من خلال تقليل الاعتماد على اليد العاملة وزيادة كفاءة التلقيح. تم تجهيز كل طائرة درون بوحدة Raspberry Pi 4B، ووحدة تحكم في الطيران Pixracer، ومضخة مياه، ومستشعرات، وكاميرا Pi، مما يتيح الكشف في الوقت الحقيقي لأشجار النخيل باستخدام نموذج YOLOv12 لاكتشاف الأجسام. تعتمد بنية النظام على طوبولوجيا اتصالات نجمية، حيث تتواصل كل طائرة بشكل حصري مع محطة أرضية مركزية باستخدام بروتوكول UDP لتبادل البيانات العامة، وبروتوكول RTP (المعتمد على UDP) لنقل الفيديو في الوقت الحقيقي. داخليًا، يُستخدم بروتوكول MAVLink لضمان الاتصال بين Raspberry Pi ووحدة Pixracer. تتيح هذه البنية توزيعًا سلسًا للمهام ومراقبة مركزية من نقطة تحكم واحدة. وبفضل دمج المعالجة الذكية المدمجة والطيران الذاتي، يمكن للنظام تنفيذ عمليات تلقيح دقيقة في بيئات غير مهيكلة. من خلال الجمع بين هذه التقنيات والاتصال اللحظي، يهدف هذا النموذج التجريبي إلى تحسين موثوقية التلقيح، وتقليل استهلاك الموارد، وتعزيز الممارسات الزراعية المستدامة، لا سيما في المناطق التي تعتمد على إنتاج التمور.

الكلمات المفتاحية : درون مستقل، الزراعة الذكية، تلقيح النخيل، YOLOv12، الطائرات بدون طيار، بروتوكول UDP، RTP، MAVLink، الكشف في الوقت الحقيقي، الزراعة الدقيقة، إنترنت الأشياء الزراعي.

Contents

List of Figures

List of Tables

General Introduction

Growing food demand, workforce limitations, and the pressing need for sustainable techniques present major challenges for modern agriculture. Pollination is one of the most essential processes for agricultural productivity, particularly for crops like date palms, which are economically vital in arid regions such as the Middle East and North Africa. However, conventional pollination techniques whether manual or natural are becoming increasingly unreliable due to rising costs, labor shortages, and the global decline in natural pollinators.

In response to these challenges, smart agriculture has emerged through the integration of modern technologies such as Unmanned Aerial Vehicles (UAVs), the Internet of Things (IoT), and Artificial Intelligence (AI). This technological synergy has enabled the development of autonomous systems that optimize resources, reduce operational costs, and promote environmental sustainability.

In this context, the proposed project presents an autonomous drone-based pollination system for palm trees. Each drone is equipped with a `Raspberry Pi`, a `Pixracer` flight controller, a camera, various sensors, and a water pump. The system uses the `YOLOv12` object detection model to identify palm trees in real time during flight. Internally, the MAVLink protocol enables communication between the Raspberry Pi and the flight controller, while external communication with the ground station is handled through the UDP protocol using a star topology. This architecture ensures distributed local intelligence with centralized coordination.

However, this work raises several important research questions:

- How can conventional pollination methods be improved to overcome challenges such as labor intensity, inconsistency, and environmental inefficiency?
- What role can UAVs and artificial intelligence play in enabling real-time, autonomous, and precise pollination?
- How can a communication architecture be designed to support coordination among multiple drones while preserving local autonomy?

These questions guide the design and development of our proposed system, which aims to provide an intelligent and scalable solution for pollination in the context of precision agriculture.

The main objective of this project is to design, implement, and evaluate a multi-drone system capable of carrying out accurate and efficient pollination operations in unstructured environments. The system aims to improve pollination efficiency, reduce manual labor, and encourage the adoption of smart and sustainable agricultural technologies.

To achieve this, the project introduces a novel prototype for autonomous date palm pollination using a coordinated multi-drone architecture. Unlike previous approaches that relied on centralized AI processing or single-drone tasks, this system employs local real-time image analysis on each drone using the YOLOv12 model, allowing each unit to independently detect palm trees and activate the pollination mechanism. Coordination among drones is handled via a lightweight UDP-based communication structure with a central ground station, while internal control within each drone is managed through the MAVLink protocol. This dual-layer communication ensures an effective balance between local autonomy and centralized supervision.

Specifically, the system integrates an embedded AI model within a Raspberry Pi onboard each drone, which collaborates with the Pixracer flight controller to carry out real-time pollination actions. A star-topology network supports centralized mission planning and monitoring, while each drone independently processes video input, detects trees, and performs pollination. This prototype has been validated through real-world deployment and sets the stage for future developments in smart agriculture systems, including scalable multi-drone coordination and autonomous path planning.

The organization of this dissertation is as follows:

The first chapter discusses the limitations of traditional pollination methods and presents the concept of smart agriculture. It also reviews related work and highlights the increasing relevance of UAVs in modern farming.

The second chapter focuses on the technological foundation of the system, describing the components used, such as object detection algorithms, control strategies, and communication protocols that enable coordination between drones and the ground station.

The third chapter covers the design of the proposed solution and the overall system architecture. It includes diagrams that illustrate the interactions between hardware and software components and the structure of the system.

The fourth chapter presents the implementation of the prototype and its validation through real-world testing. It provides performance analysis, screenshots of the operational system, and a description of the development tools and frameworks used.

The final chapter summarizes our contributions and outlines future research directions in smart agriculture using autonomous drone systems.

Chapter 1

Smart Agriculture and Drones

1.1 Introduction

The agricultural industry is confronted with significant obstacles, particularly when it comes to boosting output while maintaining sustainability. This chapter discusses the emergence of intelligent agricultural systems that utilize advanced technology such as automation, drones, and artificial intelligence (AI) to enhance contemporary farming practices. However, many farmers still rely on traditional techniques because they are not aware of these smart tools. This chapter examines the development of smart agriculture, with a focus on how technology can improve crop yields and control economic and environmental effects.

In addition, it examines conventional pollination methods, including manual and natural methods, and emphasizes the increasing difficulties brought on by the loss of natural pollinators.The chapter presents the concept of smart pollination and focuses on the use of new technologies for date palm pollination.The concept of drones is also explored by outlining their types, parts, and essential technical characteristics that make them beneficial in agriculture.

Overall, The goal of the chapter is to provide a better understanding of how drones can enhance pollination, lower labor costs, and promote sustainable farming. It also highlights this project's scientific contribution and reviews recent related work.

1.2 Smart Agriculture and its Importance

This section outlines the principles, technological advancements, and significance of Smart Agriculture in achieving sustainable and efficient farming.

1.2.1 Smart Agriculture (SA)

Smart agriculture is a modern agricultural paradigm that leverages advanced technologies to improve sustainability, productivity, and operational efficiency [2]. It integrates big data, cloud computing, artificial intelligence (AI), sensor systems, and the Internet of Things (IoT) to optimize agricultural operations [57, 56]. Through the use of smart sensors, real-time monitoring of environmental parameters is made possible [48], enabling decisions based on data in areas like crop health analysis, pest control, and precision irrigation[57, 56].

Smart agriculture optimizes inputs like water and fertilizer and reduces resource waste to increase yields while fostering environmental sustainability [2, 29]. Technologies such as agricultural robots and Unmanned Aerial Vehicles (UAVs) have been employed to automate tasks and increase field efficiency [48, 57]. This digital transformation allows farmers to adopt precision agriculture techniques that result in higher output and lower costs.

Furthermore, smart agriculture fosters resilience in the face of climatic unpredictability and rising population needs by marking a significant shift toward more intelligent and sustainable food systems [7]. Thus, Smart Agriculture serves as a cornerstone for the future of sustainable and data-driven farming.

Figure 1.1 shows : Architecture for Smart Agriculture.

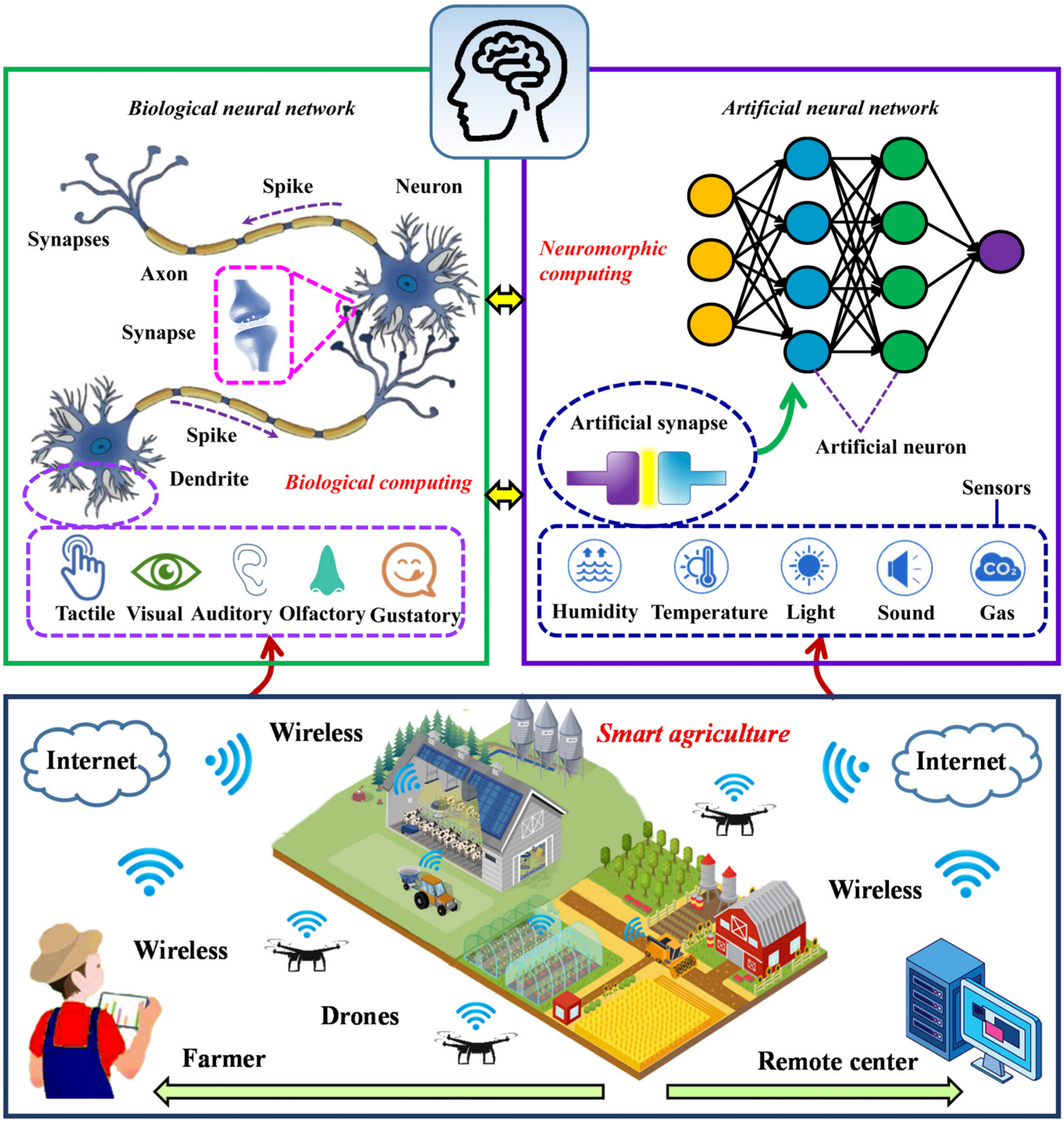


Figure 1.1. Architecture for Smart Agriculture [50]

1.2.2 Evolution of Agriculture

Historically, making of food on farmed areas for human survival and animal reproduction was linked to old agricultural practices [71]. In order to obtain the food required for the existence, humans have been cultivating land and rearing animals since ancient times. As seen in the illustration, this practice known as agriculture has undergone a slow and long-term evolution, progressing from Agriculture 1.0 to 4.0.

The evolution of agriculture can be divided into four major phases. Agriculture 1.0 corresponds to the traditional era, relying on human and animal power, with limited productivity. Agriculture 2.0, marked by the introduction of steam engines and chemicals, has led to a significant improvement in productivity, but at the cost of serious environmental impacts. Agriculture 3.0, resulting from advances in computer science and robotics, has introduced intelligent machines, reducing the excessive use of chemicals and improving the precision of farming practices. Finally, the current Agriculture

4.0 integrates advanced technologies such as the Internet of Things, big data, artificial intelligence, and wireless sensor networks to sustainably optimize agricultural systems [81].

Figure 1.2 shows : Development of agriculture throughout history.

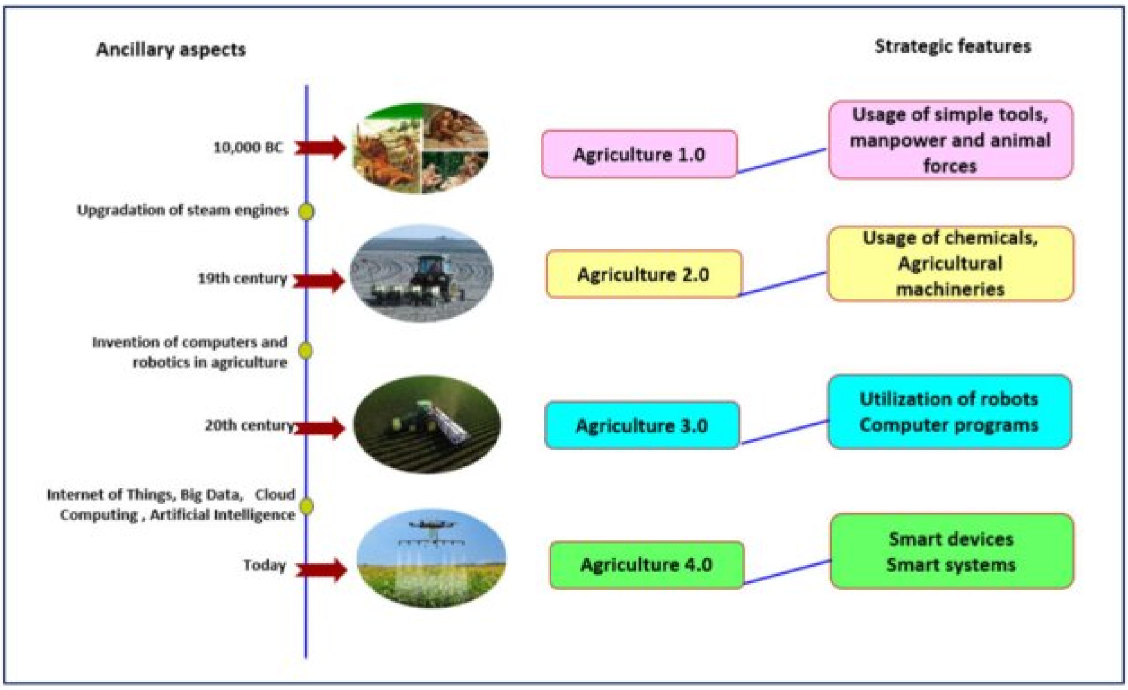


Figure 1.2. Development of agriculture throughout history [18]

1.2.3 Role of Technology in Enhancing Crop Production

Technology plays a key role in enhancing crop production through smart tools like Precision Agriculture, IoT, data analytics, and remote monitoring. Wireless networks and sensors help control irrigation, lighting, and ventilation systems efficiently, leading to improved crop quality and up to 25% water savings [68].

The integration of smart and precision farming allows for data-driven decisions using drones and ground vehicles for accurate land monitoring [12]. Technologies such as GPS, sensors, and cloud systems enable precise input application based on soil needs, increasing yield and reducing pollution [31].

Machine learning enhances the prediction of pests and diseases and improves planting and harvesting schedules [12, 31]. It also supports early detection and targeted pest control, better harvest timing, and reduced waste [31].

Overall, technology is transforming traditional agriculture into smart, data-driven systems that boost productivity and promote environmental and social sustainability [68, 12, 31].

Figure 1.3 shows :Applications of Smart Technologies in Crop Monitoring and Pest Management.

Figure 1.3. Applications of Smart Technologies in Crop Monitoring and Pest Management [31]

1.2.4 Environmental and Economic Impact of Smart Agriculture

Modern farming practices have complex and multifaceted impacts on both the environment and the economy. On one hand, mechanization and the use of agrochemicals have significantly boosted agricultural productivity and improved global food security. These advancements have played a key role in reducing poverty by fostering economic growth and supporting rural development [72].

However, the ecological consequences of these practices are becoming increasingly evident. The widespread use of chemical fertilizers and pesticides has led to several serious environmental concerns, including soil degradation, water pollution, and loss of biodiversity [34]. Moreover, excessive groundwater extraction for irrigation often driven by modern agricultural techniques has caused severe water stress in many regions of the world.

1.2.5 Importance of Smart Agriculture

Smart agriculture plays a crucial role in transforming the traditional farming sector into a data-driven and sustainable system. Its significance can be summarized in the following key points:

- **Increased Agricultural Productivity:** By leveraging real-time information and predictive analytics, farmers can significantly enhance crop yields and quality. [78]
- **Efficient Resource Management:** Smart agriculture allows optimal use of water, fertilizers, and pesticides, reducing waste and minimizing environmental damage. [84]
- **Real-Time Environmental Response:** Sensors and monitoring systems provide

accurate, up-to-date information, allowing for rapid adjustments in farming operations based on soil and weather conditions. [42]

- **Cost Reduction:** Automation and intelligent control reduce labor dependency and operational costs, making farming more economically viable. [10]
- **Environmental Sustainability:** Smart agriculture supports sustainable practices by reducing harmful inputs and conserving natural resources. [30]
- **Climate Change Adaptation:** With climate-smart techniques, farmers can better adapt to weather variability, drought, and rising temperatures. [23]

1.3 Pollination Practices and Challenges

1.3.1 Natural Pollination Methods (Wind, Insects, Birds)

Natural pollination occurs without human intervention and relies mainly on environmental agents such as wind, insects, and birds.

- Wind pollination (anemophily) is common in crops like wheat, rice, and corn, where pollen grains are light and easily carried by air.
- Insect pollination (entomophily), especially by bees, is vital for fruit trees, vegetables, and oil-producing crops. Bees are considered the most efficient pollinators due to their hairy bodies and behavior.
- Bird pollination (ornithophily) occurs in specific regions, especially tropical areas, where birds like hummingbirds transfer pollen as they feed on nectar.

These natural agents maintain ecosystem balance and support the reproduction of more than 75 % of flowering plant species [43, 26].

1.3.2 Manual Pollination and Human Intervention

In cases where natural pollination is insufficient or unreliable, manual pollination techniques are used. These methods involve human intervention to transfer pollen from male to female flowers using tools such as brushes or cotton swabs.

Manual pollination is especially common in:

- Greenhouses or controlled environments.
- Crops with low natural pollination rates.
- Areas with declining pollinator populations.

While manual pollination increases control and precision, it is labor-intensive, time-consuming, and costly, especially for large-scale farms[52, 17].

1.3.3 Challenges and Limitations of Traditional Pollination

Traditional pollination faces multiple technical and environmental challenges, including:

- Climate change, which impacts the timing of flowering and pollinator pastime.

- Pesticide use, which reduces pollinator health and populace.
- Habitat destruction, leading to fewer nesting and feeding sites.
- Diseases and parasites like Varroa mites affecting bee colonies.

Moreover, relying on manual pollination is unsustainable for large farms and not always feasible in rural areas lacking labor resources[63, 37].

1.3.4 Decline of Natural Pollinators and Its Impact on Agriculture

The ongoing decline of herbal pollinators, together with bees, butterflies, and different insects, represents a essential task to worldwide agricultural systems. This phenomenon threatens each crop productivity and ecosystem stability via several interconnected mechanisms.

Key Factors Contributing to Pollinator Decline

Multiple anthropogenic pressures are driving pollinator losses:

- Chemical-intensive agriculture, particularly the widespread use of systemic pesticides
- Habitat fragmentation and degradation from land-use changes
- Climate disruptions affecting seasonal synchrony between plants and pollinators
- Emerging diseases and parasites impacting pollinator health

Agricultural Implications

The reduction in pollinator populations has direct consequences for food production:

- Decreased yields in pollinator-dependent crops
- Reduced genetic diversity in plant populations
- Compromised food security for human populations
- Economic losses throughout agricultural value chains [58]

1.3.5 Smart Pollination

The decline of pollinators especially bees has become a global concern. The causes include:

- Intensive farming.
- Pesticides and herbicides.
- Diseases, parasites, and climate disruptions.

This decline directly threatens food production. Crops like almonds, apples, blueberries, and melons depend heavily on pollinators. The economic value of pollination services is estimated at $235–$577 billion USD annually.

Without sufficient pollinators, yields decrease, quality drops, and food security is at risk [25, 28].

1.3.6 Date Palm Pollination

Pollination in date palm is defined as the transfer of pollen grains from the male palm tree to the female, either naturally via wind or insects, or through manual or mechanical methods [64, 4]. Since natural pollination is insufficient to achieve economically viable production, artificial pollination is typically used by manually placing fresh male flower strands into the female date palm flowers [64]. This traditional method is effective in improving fruit set rates, but it requires considerable labor and is costly.

Figure 1.4 shows : the main manual methods used for pollinating date palms.

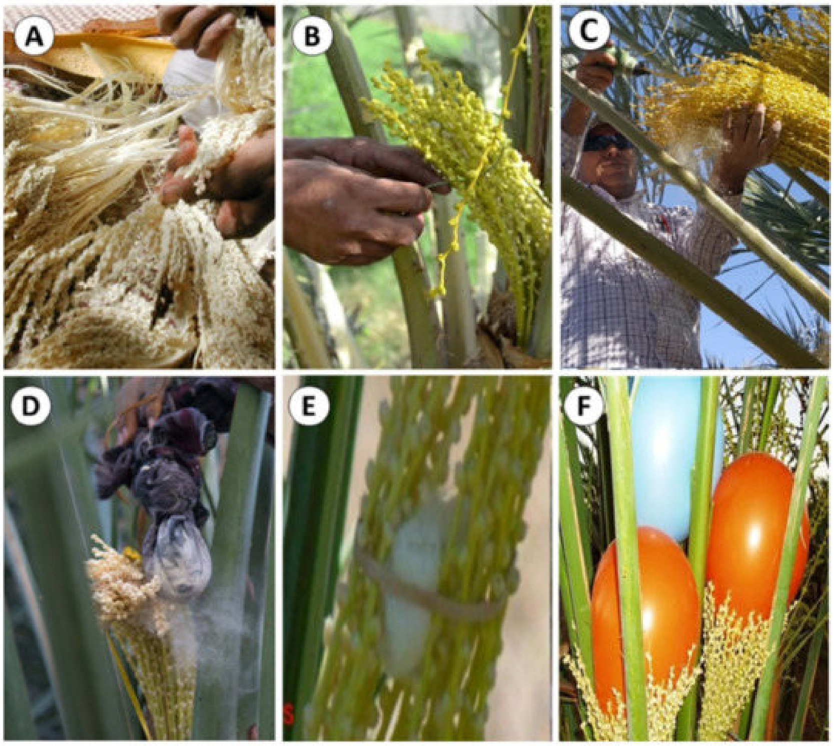


Figure 1.4. Manual pollination techniques in date palms: (A) Separation of male flower strands, (B) Insertion into female inflorescence, (C) Squeeze bulb dusting, (D) Use of cloth to apply pollen, (E) Embedded cotton soaked with pollen, and (F) Balloons filled with pollen for gradual release [66].

In contrast, alternative mechanical techniques have emerged, relying on mixing pollen with filler materials such as wheat flour or cornstarch, or using spray-based systems. These approaches reduce pollen consumption and improve labor efficiency while maintaining production quality [64, 4]. Some studies have shown that mechanical pollination can yield results comparable to manual pollination in terms of fruit set percentage, yield, and fruit quality [4].

Figure 1.5 presents common mechanical pollination systems used in date palm farming.

Figure 1.5. Mechanical pollination techniques in date palms: (A) Hand pollination using liquid suspension, (B) High-pressure liquid pollination using a tractor-mounted sprayer, (C) Inflorescence before and after liquid pollination, (D) Manual pollination machine with dry pollen, (E) Hand-operated dry pollen applicator, (F) Hydraulic pollination machine mounted on a tractor, (G) Motorized duster for dry pollen, (H) Pressurized sprayer system, (I) Electric pollinator operated from the ground, (J) Robotic arm prototype for automated pollination using AI [66].

1.4 Unmanned Aerial Vehicle (UAV)

This section provides a concise overview of UAVs, detailing their types, components, and core technologies.

1.4.1 Definition of a Unmanned Aerial Vehicle (UAV)

A drone, or Unmanned Aerial Vehicle (UAV), is an aircraft that operates without a human pilot onboard. It can be remotely controlled or fly autonomously using onboard systems. Originally developed for military use, drones are now widely adopted in various civil and commercial fields, including aerial photography, delivery, agriculture, security, and surveying [76].

1.4.2 Types of Unmanned Aerial Vehicles (UAVs)

The classification of Unmanned Aerial Vehicles (UAVs) provides a structured framework for understanding and deploying drones across various domains. Classification parameters include size, design, altitude, power source, operational autonomy, and intended application. The following summarizes the main types:

- **By Size:**
  - **Nano:** Less than 250 mm; designed for close-range surveillance (e.g., Black Hornet Nano).
  - **Micro:** 250–500 mm; used for lightweight outdoor operations with basic cameras.
  - **Mini:** 0.5–2 m; suitable for agriculture and search-and-rescue missions.
  - **Small:** 2–5 m; deployed for industrial-scale inspections.
  - **Tactical:** 5–10 m; employed in military reconnaissance and battlefield support.
  - **Strike:** >10 m; used in long-range surveillance and precision strike missions (e.g., MQ-9 Reaper).
- **By Aerodynamic Design:**
  - **Fixed-Wing:** High endurance and efficient for mapping and surveillance (e.g., Parrot Disco).

    Figure 1.6 shows a fixed-wing drone.

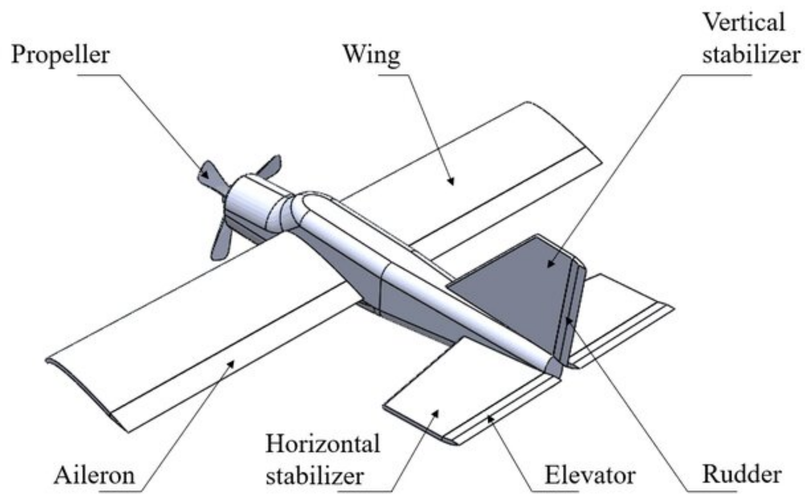


Figure 1.6. Fixed-wing drone structure [16].

- **Rotary-Wing:** Provides hovering and agility for inspections and photography (e.g., DJI Phantom).

  Figure 1.7 shows a multirotor drone.

Figure 1.7. Multirotor drone structure [44].

- **Hybrid:** Combines VTOL capability with long-range flight (e.g., WingtraOne).

  Figure 1.8 shows a hybrid drone with VTOL capability.

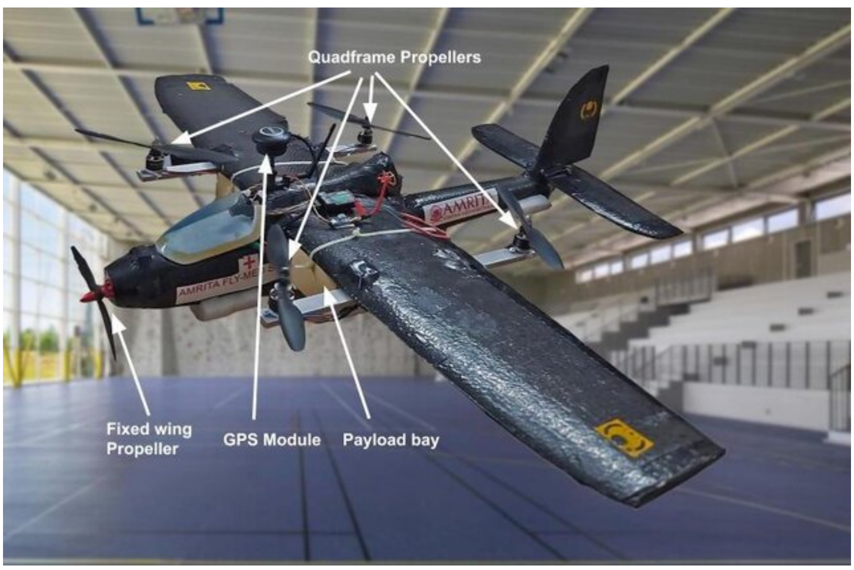


Figure 1.8. Hybrid VTOL drone [55].

- **Flapping-Wing:** Mimics natural flight; used in biomimetic and stealth operations.

  Figure 1.10 shows a Flapping-Wing drone.

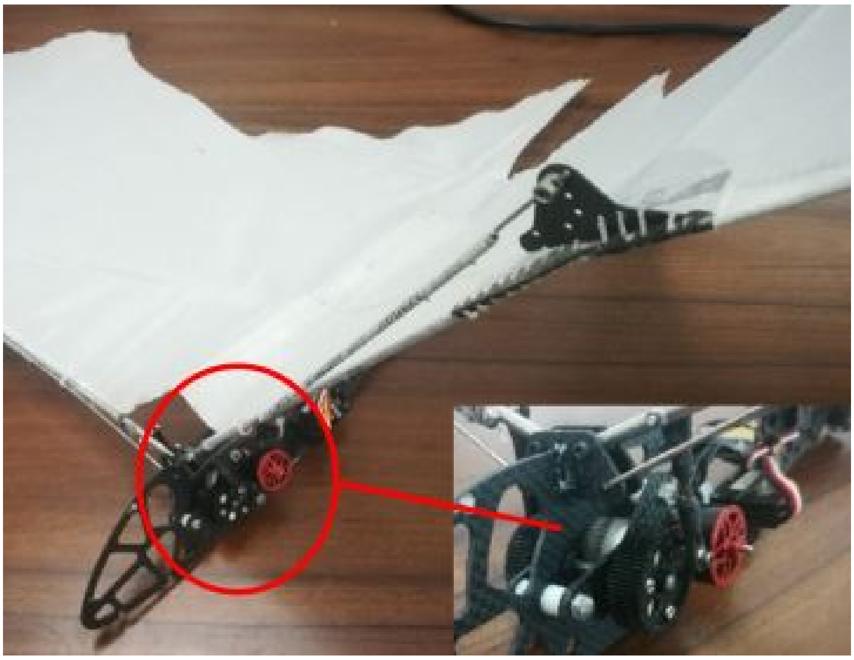


Figure 1.9. Flapping-Wing [11]

Figure 1.10. Flapping-Wing [11]

- **By Operating Altitude:**
  - **High Altitude Platforms (HAP):** Operate above 20 km; used in atmospheric research (e.g., Zephyr S).
  - **Medium Altitude Platforms (MAP):** Between 5–20 km; used in military surveillance (e.g., MQ-9 Reaper).
  - **Low Altitude Platforms (LAP):** Below 5 km; suitable for urban monitoring and delivery missions.
- **By Flight Environment:**
  - **Indoor:** Compact and impact-resistant for indoor navigation (e.g., Tello Drone).

  - **Outdoor:** GPS-enabled and weather-resistant for field operations (e.g., DJI Phantom 4).
  - **Underwater:** Designed for marine exploration (e.g., BlueROV2).
  - **Space:** Adapted for low-gravity environments (e.g., NASA Mars Helicopter).

- **By Power Source:**
  - **Electric (Battery):** Lightweight but limited in range.
  - **Fuel-Based:** Suitable for high-payload or long-endurance missions (e.g., Yamaha RMAX).
  - **Solar-Powered:** For extended flight time in environmental monitoring.
  - **Hydrogen Fuel Cell:** Eco-friendly with high endurance (e.g., Doosan DS30).

- **By Level of Autonomy:**
  - **Manual:** Fully controlled by a human operator.
  - **Semi-Autonomous:** Assisted by onboard autopilot.
  - **Fully Autonomous:** Executes missions independently using AI (e.g., Skydio 2).
  - **Swarm:** Coordinated multi-drone missions (e.g., LOCUST).

- **By Application:**
  - **Military:** Reconnaissance, surveillance, and tactical strikes (e.g., MQ-9 Reaper).
  - **Commercial:** Photography, delivery, mapping (e.g., DJI Inspire 2).
  - **Industrial:** Infrastructure inspection and precision agriculture (e.g., Sense-Fly Albris).
  - **Environmental:** Climate monitoring and wildlife protection (e.g., Zephyr S).
  - **Medical:** Emergency supply delivery (e.g., Zipline).
  - **Recreational:** Hobby flying and aerial videography (e.g., DJI Mini 3).

Figure 1.11 shows Details Classification of Drone.

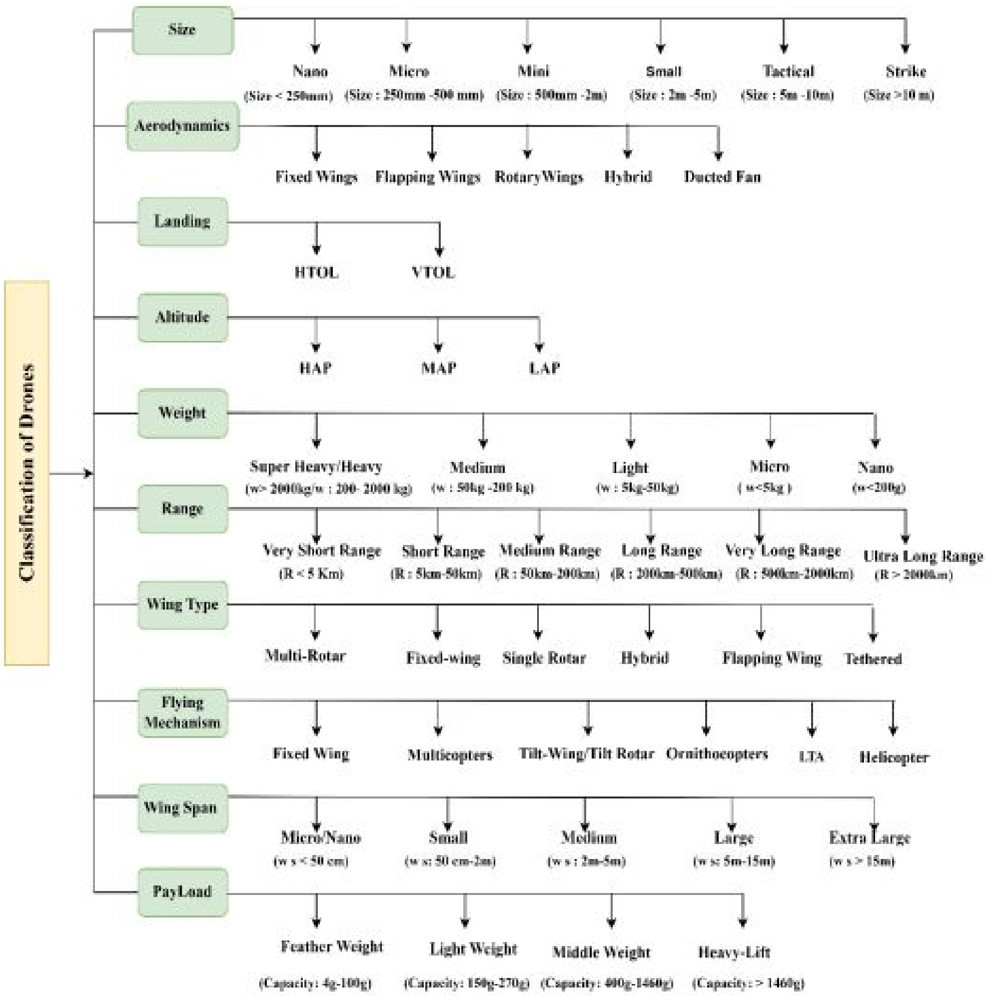


Figure 1.11. Classification of Drone [76]

1.4.3 Basic Components of a Drone

Typical drone parts include the following:

- **Frame:** All the components are held together by the structural frame.
- **Motors and Propellers:** Provide the important thrust for carry and maneuvering.
- **ESC (Electronic Speed Controller):** Controls the speed of the motors.
- **Flight Controller:** The drone's brain is in charge of navigation and stability.
- **Battery:** supplies electricity to the drone's electrical systems.
- **Additional Modules:**such as communication systems, cameras, and GPS.

Figure 1.12 shows the basic parts of a drone.

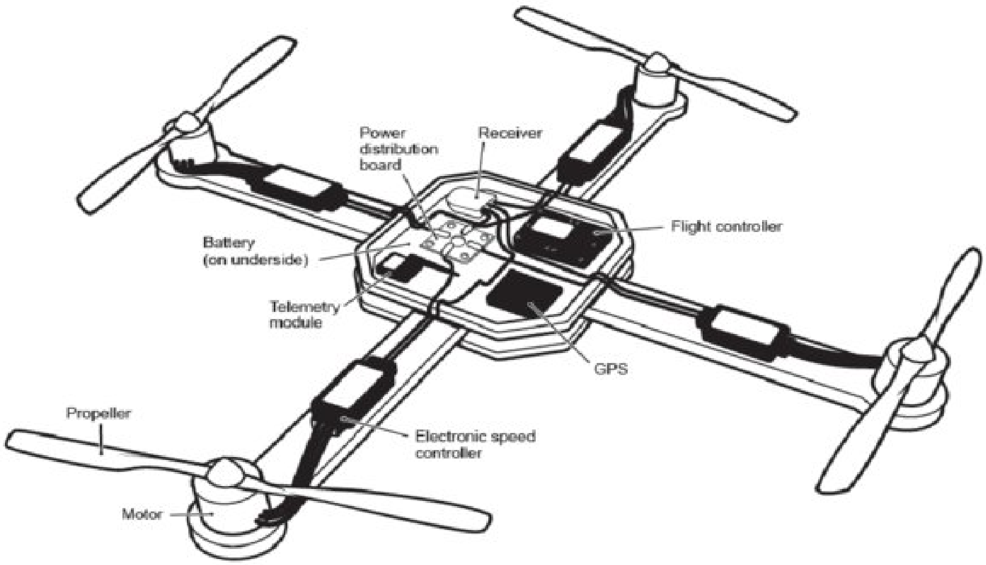


Figure 1.12. Labeled structure of a quadrotor drone, showing main components including frame, propellers, ESCs, flight controller, battery, GPS, and receiver [41].

1.4.4 Key Technical Features

Modern drones come with cutting-edge technology, such as:

- **Autonomous Navigation:** using inbuilt sensors and GPS.
- **Stable Hovering:**In particular, multirotor drones.
- **Real-Time Data Transmission:** For video streaming or sensor readings.
- **Sensor-Based Stabilization:** Such as gyroscopes and magnetometers.

1.5 Role of Drones in Agriculture

1.5.1 Introduction to Agricultural Drones

Agricultural drones are unmanned aerial vehicles (UAVs) equipped with sensors and imaging capabilities, designed specifically for use in farming. These drones enable precise monitoring and management of crops, soil, and livestock, thereby enhancing decision-making and reducing labor costs [82].

Figure 1.13 shows : Agriculture Drone.

Figure 1.13. Agriculture Drone [19]

1.5.2 Applications of Drones in Modern Farming

Drones are used in various agricultural tasks, such as:

- Crop health monitoring via multispectral imaging.
- Precision spraying and fertilization.
- Livestock tracking.
- Soil analysis and field mapping.

These applications improve yield, optimize resources, and reduce environmental impact [74, 24].

1.5.3 Comparison Between Manual and Drone-Assisted Farming

Compared to manual methods, drone-assisted farming offers several advantages:

- Higher efficiency and coverage.
- Real-time data collection.
- Reduced human labor and exposure to chemicals.

However, drone systems require technical knowledge and initial investment, which might not be accessible to all farmers [36].

1.5.4 Challenges in Implementing Drone Technology in Agriculture

Despite their benefits, drones face several implementation barriers in agriculture:

- Regulatory restrictions and airspace limitations.
- High cost of advanced equipment.
- Lack of training and awareness among farmers.
- Battery limitations affecting flight time.

Addressing these challenges is key to the widespread adoption of agricultural drones [39].

1.5.5 Drone And Pollination

The use of drone technology is increasingly becoming a key factor in improving agricultural practices, especially in artificial pollination. Drones have been used for artificial pollination of date palms in Middle Eastern countries, with this technology being applied in the orchards of Oman for pollinating cultivars such as Naghal, Khanesi, Fard, and Khasab [64]. Additionally, an autonomous drone-based pollination system has been developed for tomato cultivation, as tomato farming faces significant challenges in pollination, which is crucial given the high global demand for tomatoes [33].

Figure 1.14 shows :Pollination system configuration using drones.

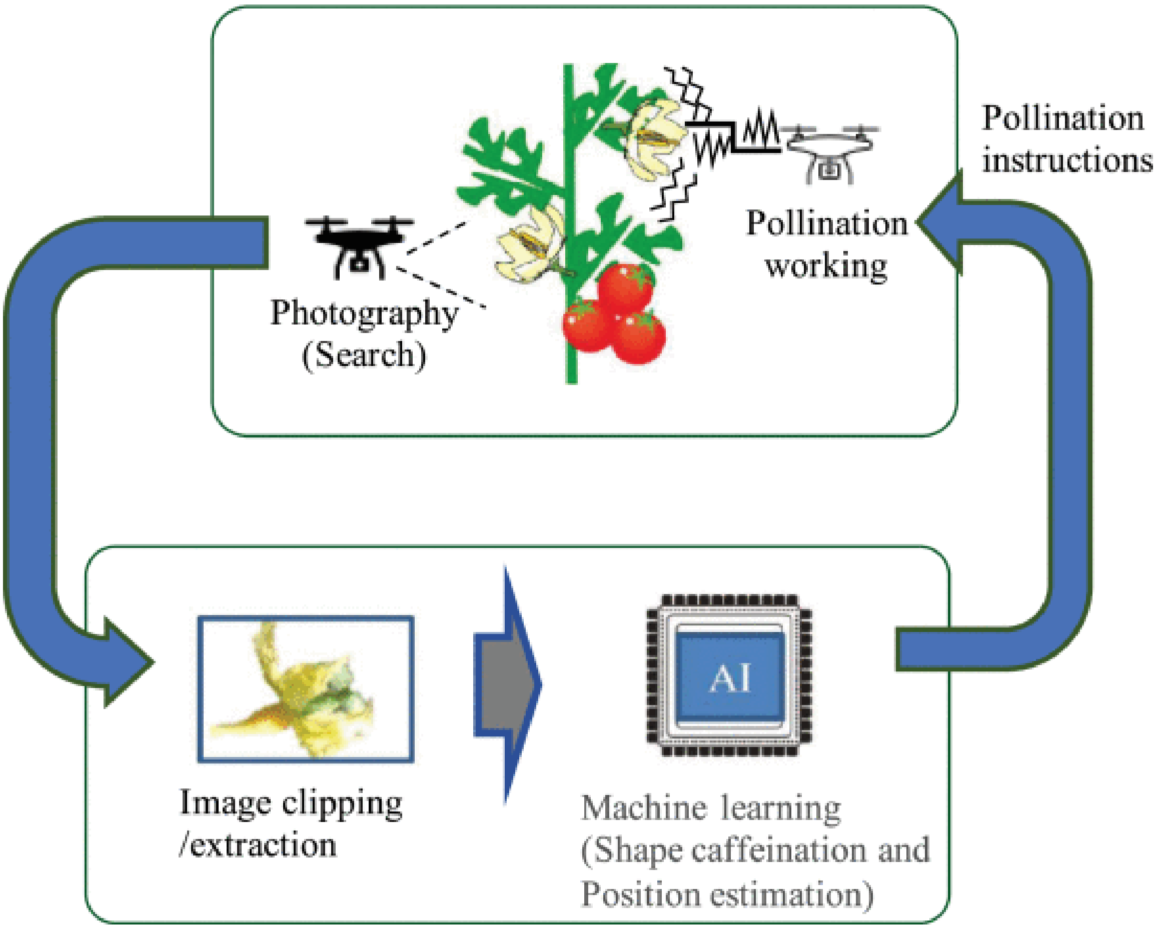


Figure 1.14. Pollination system configuration using drones [33]

1.5.6 Benefits of Drone-Based Pollination

- **Increased Efficiency and Precision in Pollination:**

  Drone-based pollination offers targeted and uniform pollen delivery, reducing waste and enhancing fruit set compared to traditional methods [83].

- **Cost Reduction and Improved Labor Efficiency:**

  With labor shortages in agriculture, drones reduce dependency on manual workers and lower long-term operational costs [69].

- **Pollination in Controlled Environments (Greenhouses, Vertical Farms):**

  Drones are especially beneficial in enclosed spaces like greenhouses or vertical farms where natural pollinators are absent. Their small size and programmable navigation make them ideal for such environments [79].

- **Case Studies on Successful Drone Pollination Applications:**

  Several studies have demonstrated the effectiveness of drones in pollinating crops like kiwifruit, apple trees, and strawberries, with comparable or better results than manual pollination [60].

1.6 Existing Applications of Drones in Smart Agriculture

The domain of smart agriculture has witnessed a significant surge in the application of Unmanned Aerial Vehicles (UAVs) for diverse tasks, notably precision crop spraying. This section outlines four pertinent research endeavors that employ comparable techniques or address closely related aspects to our current work, albeit potentially differing in specific objectives and implementations:

1.6.1 Smart Agriculture Drone for Crop Spraying Using Image-Processing and Machine Learning Techniques: Experimental Validation

The work [67] presents a smart agricultural drone designed for crop spraying, integrated with Internet of Things (IoT) technologies and machine learning using TensorFlow Lite with the EfficientDetLite1 model. It is trained on a custom dataset to detect three crop types: pineapple, papaya, and cabbage, achieving an inference time of 91 milliseconds.

The drone offers two spray modes:

- Mode A: 100% spray capacity
- Mode B: 50% spray capacity

These modes are selected based on real-time data, showcasing the potential of IoT for real-time monitoring and autonomous decision-making.

Powered by the X500 development kit, the drone features:

- Payload: 1.5 kg
- Flight time: 25 minutes
- Speed: 7.5 m/s

- Altitude: 2.5 m

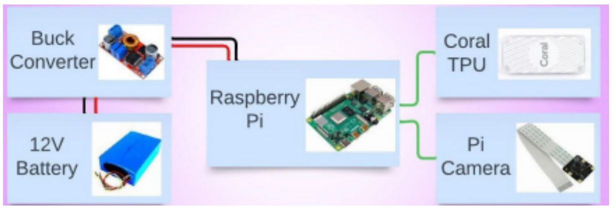


Figure 1.15. Image processing system [67]

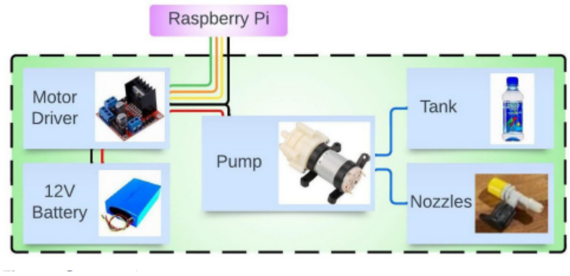


Figure 1.16. Crop-spraying system [67]

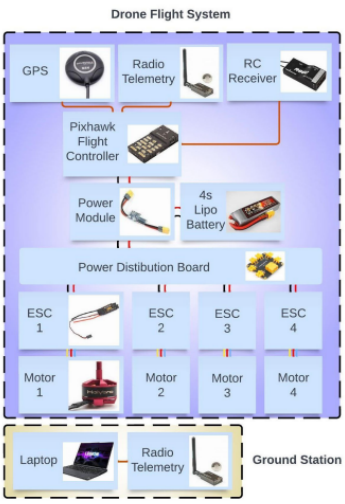


Figure 1.17. Drone system and ground station [67]

1.6.2 Innovative and Effective Spray Method for Artificial Pollination of Date Palm Using Drone

This work [6] aimed to:

1. Develop a fast, efficient, and low-cost drone-based pollination method using water-suspended pollen (3 g/L), reducing pollen use and labor costs.
2. Evaluate its impact on fruit set (FS) percentage, pollination efficiency (PE), fruit retention, total yield, and fruit quality.

Three date palm cultivars Barhi, Lulu, and Khesab were pollinated using:

- Hand pollination (HP)
- Spray pollination (HS)
- Drone pollination (DS) with water-suspended pollen

Key findings:

- DS had significantly lower FS in Lulu and Khesab compared to HP and HS, but not in Barhi.
- PE was not affected in Barhi and Lulu, but lower in Khesab with DS (0.81) compared to HS (0.94) and HP (0.99).
- DS showed reduced fruit retention and bunch weight, but significantly improved physical fruit quality in all cultivars.
- Fruit color, firmness, TSS%, acidity, pH, and vitamin C were not affected by the pollination method.

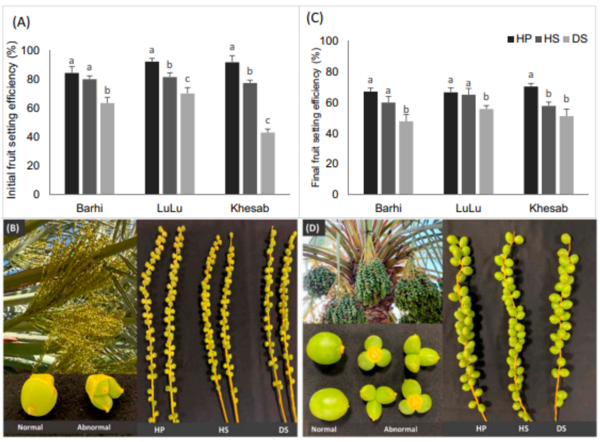


Figure 1.18. Effect of Different Pollination Methods on Date Palm Fruit Set [6]

1.6.3 Impact of Autonomous Drone Pollination in Date Palms

This work [77] investigated the application of drone-assisted pollination in date palm cultivation in Oman during the 2022 season. The findings indicated a notable enhancement in pollination efficiency, particularly among tall palm trees, where the fruit set percentage surpassed 66%. Furthermore, the research underscored the challenges associated with limited drone accessibility in densely planted areas and among shorter palms. The integration of liquid pollen suspensions, GPS-guided spraying systems, and high-resolution imaging technologies significantly reduced manual labor and pollination time by more than 95% positioning drone technology as a viable and efficient tool within the domain of precision agriculture. Nonetheless, regulatory constraints and technical limitations continue to impede the widespread adoption of this approach.

1.6.4 Development of Pear Pollination System Using Autonomous Drones

This work [54] developed an autonomous pollination system using a drone equipped with a RealSense depth camera. The drone captures images of trees, which are analyzed by an external server running the YOLOv7 model trained on images of pear blossoms. The system accurately detects pollinable flowers and calculates their coordinates using depth data and the camera's intrinsic parameters. A Raspberry Pi mounted on the drone handles flight control and data processing via ROS and MAVROS, while image and telemetry data are transmitted using the MAVLink protocol. Once the flower locations are identified, the drone navigates to the target coordinates and performs pollination using a micro-electrostatic spray mechanism. Experiments demonstrated high accuracy in flower detection and stable navigation between closely spaced trees, supported by RTK-GNSS technology.

1.7 Our Contribution

Based on previous studies on drone-based spraying and autonomous pollination, we aim to develop a prototype of a smart multi-drone system for date palm pollination in a natural and unstructured environment, where palm trees are distributed randomly.

Each drone is equipped with a Pixhawk flight controller and a Raspberry Pi board. A Pi Camera mounted on the drone captures real-time images, which are processed locally using the YOLOv12 object detection model to identify palm trees and detect pollen availability. The Raspberry Pi also controls the pollination pump and communicates with a centralized ground system using the UDP protocol.

The system is designed to enable centralized coordination of multiple drones, allowing for efficient navigation and precise pollination in complex environments. This prototype serves as a foundation for future enhancements such as autonomous path planning and scalable multi-drone operation. The main goal is to reduce manual labor, optimize pollination efforts, and promote the use of smart agricultural technologies.

1.8 Conclusion

This chapter provided an overview of modern agricultural practices and the challenges associated with traditional pollination methods. We introduced drones as a promising technological solution, detailing their types and potential applications in precision agriculture. Particular emphasis was placed on their role in automated pollination, highlighting various mechanisms and strategies employed in existing research.

Additionally, we reviewed several related works to understand the current state of development in drone-assisted pollination systems. This foundational background sets the stage for the next chapter, where we will present the design and architecture of our proposed system.

Chapter 2

Algorithms and AI in Pollination

2.1 Introduction

This chapter focuses on the essential technologies and algorithms that power intelligent pollination systems using drones. It explores the core mechanisms of object detection, quadrotor flight control, and communication between drones and the ground station. Each section presents the principles, architectures, and methodologies required to ensure accuracy, coordination, and high performance. By understanding these components, we can design autonomous systems capable of handling complex agricultural environments and executing pollination tasks efficiently and reliably. This foundational knowledge is key to developing scalable and intelligent drone-based solutions in precision agriculture.

2.2 Object Detection

Object detection has evolved from traditional computer vision methods to deep learning-based approaches that significantly improved accuracy and efficiency. However, several challenges persist, particularly in detecting partially occluded or small objects within complex scenes. Variations in object scales due to perspective and distance further complicate detection tasks. Additionally, real-time processing requirements pose limitations, especially in critical applications such as autonomous driving and surveillance. Recent advancements, including attention mechanisms and transformer architectures, are being explored to overcome these obstacles and enhance detection robustness[75].

Several key methodologies have emerged in the field of object detection, including:

1. **Traditional Methods:** These early techniques relied on handcrafted features and include the Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), Haar-like features, and the sliding window technique. Despite their foundational role, these methods struggled with object variability and complex environments[3, 75, 62].

2. **Machine Learning Approaches:** With the introduction of supervised learning, models such as Decision Trees (DTs), k-Nearest Neighbors (KNN), Support Vector Machines (SVMs), and ensemble techniques like AdaBoost gained popularity. Deformable Part Models (DPMs) also enhanced performance by modeling object variability[75, 46].

3. **Deep Learning-Based Methods:** The field was revolutionized by Convolutional Neural Networks (CNNs), leading to the development of models like Region-based CNNs (R-CNNs), Single Shot Detectors (SSDs), and You Only Look Once (YOLO) architectures. Anchor-free detectors such as CornerNet, CenterNet, and ExtremeNet further refined detection precision[27, 75].

4. **Advanced Techniques:** Recent innovations focus on enhancing model robustness and efficiency. Vision Transformers (ViTs), multi-scale and context-aware detection via Feature Pyramid Networks (FPNs), and lightweight architectures like MobileNet, SqueezeNet, and EfficientNet are designed for resource-constrained devices. Additional model optimization techniques, such as pruning, quantization, and knowledge distillation, are actively used to improve deployment

scalability[75].

This figure illustrates the main categories of object detection approaches, presenting the various methodologies in accordance with technological advancements from traditional techniques to modern deep learning-based methods.

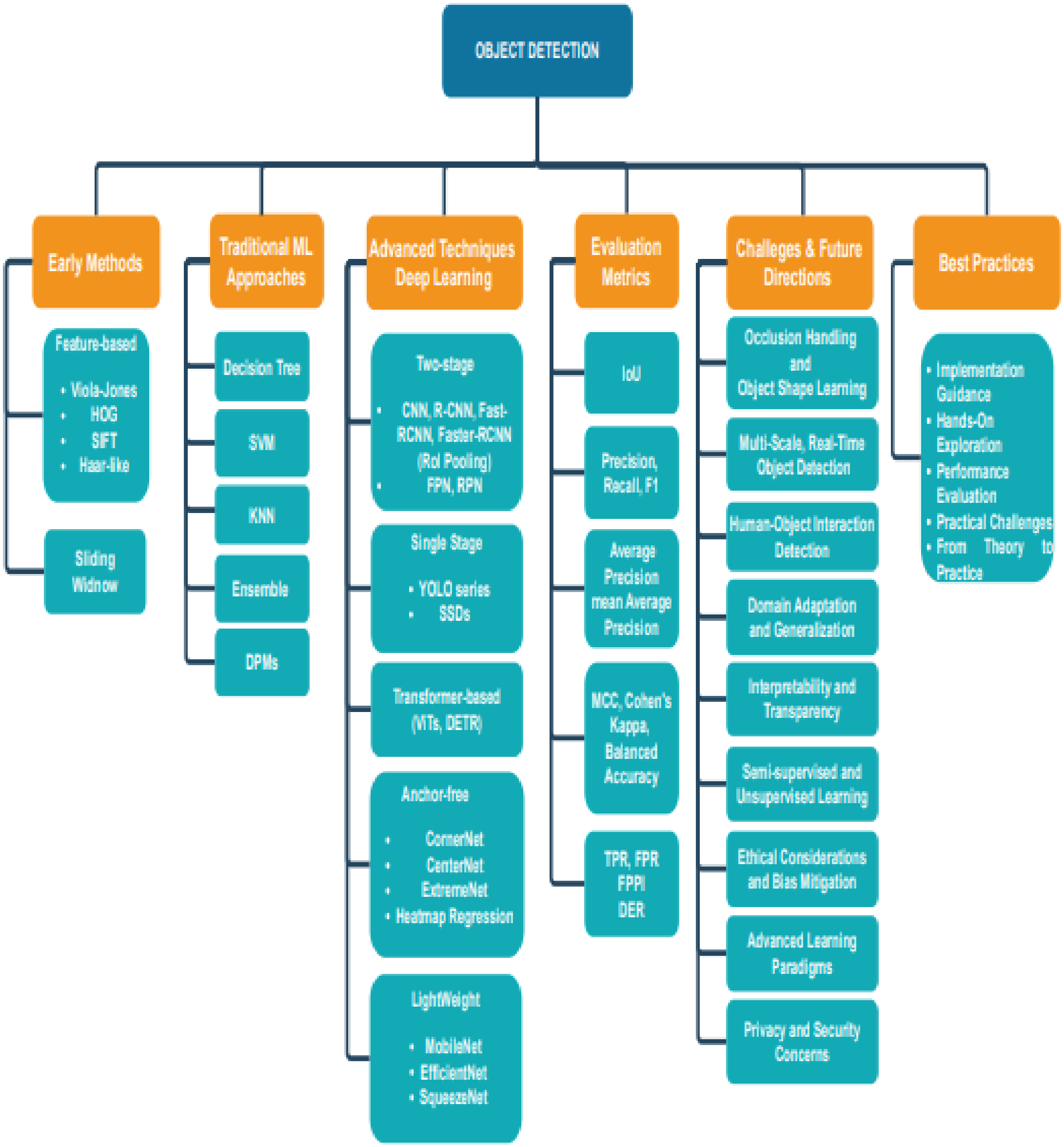


Figure 2.1. An overview of the object detection landscape [75].

2.3 Control of Quadrotor Drones

This section follows the approach outlined in [32], which provides a comprehensive analysis of quadrotor control systems, covering their structure, dynamics, and control strategies.

quadrotor control involves:

- **Underactuation**: Only four control inputs are available to manage six degrees of freedom.
- **Nonlinear dynamics**: There is a strong coupling between translational and rotational motions.
- **External disturbances**: Wind gusts, payload variations, and sensor noise can significantly affect stability.

2.3.1 Control Architecture

The control system of quadrotors typically adopts a **hierarchical architecture** comprising two nested loops:

- **Outer loop (Position control)**: Generates desired orientation angles based on a predefined reference trajectory.
- **Inner loop (Attitude control)**: Stabilizes and tracks the orientation using torque commands.

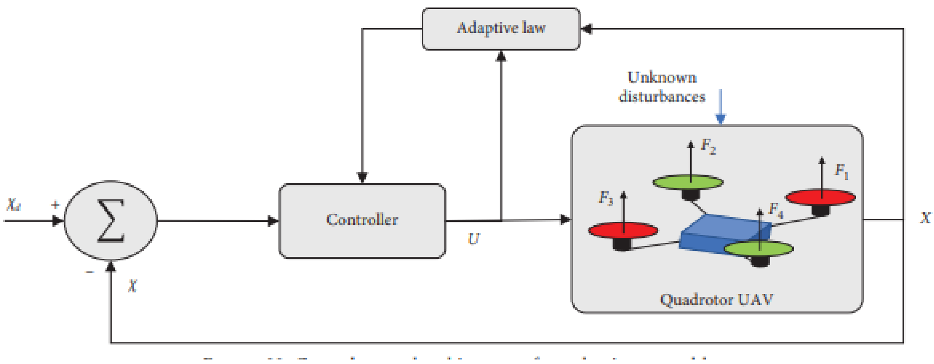


Figure 2.2. Cascade control structure for quadrotors [32]

2.3.2 Control Strategies

UAV control relies on diverse strategies to ensure stable and efficient flight.

2.3.2.1 Linear Control Methods

Classical linear techniques such as Proportional-Integral-Derivative (PID) control and Linear Quadratic Regulator (LQR) are widely adopted for their simplicity and effectiveness in well-modeled, noise-limited environments.

2.3.2.2 Nonlinear Control Methods

To manage the inherent nonlinear behavior of quadrotor dynamics, nonlinear strategies such as Backstepping and Sliding Mode Control are employed. These techniques offer enhanced robustness to modeling uncertainties and external disturbances.

2.3.2.3 Intelligent Control Methods

Recent advances have introduced Artificial Intelligence-based control, particularly Neural Networks, which provide adaptability and learning capability in dynamic and unpredictable conditions.

2.3.3 Quadrotor Dynamics

The dynamics of a quadrotor are governed by Newton-Euler equations, which describe both translational and rotational motion. The control inputs thrusts generated by the four rotors are mapped via a control allocation matrix to achieve the desired trajectory and orientation.

2.3.4 Validation Methods

Control systems for quadrotors are commonly validated through a combination of simulation and hardware-in-the-loop testing:

- **Model-in-the-Loop (MIL)**: The control algorithm is tested within a simulated model.
- **Software-in-the-Loop (SIL)**: The auto-generated code is verified in simulation.
- **Processor-in-the-Loop (PIL)**: The code is deployed on the actual processor to evaluate real-time performance.

2.3.5 Future Directions

Emerging trends and open research problems in quadrotor control include:

- Designing hybrid control architectures that combine multiple strategies to enhance performance.
- Developing fault-tolerant controllers to ensure reliability under system failures.
- Integrating edge computing technologies to support onboard AI-based decision-making.

2.4 Communication System Between Drones

This section presents key communication topologies used in multi-UAV systems.

2.4.1 Communication Topologies

In multi-UAV systems, the communication topology defines how drones are interconnected to share data and coordinate actions. The most common topologies include the following:

- **Star Topology :**

  All drones communicate through a central node, usually a ground station or a lead UAV. This structure is simple and efficient for small-scale missions but suffers from a Single Point of Failure (SPOF) if the central node fails, the entire network is affected [51, 14].

Figure 2.3 shows : Star Topology.

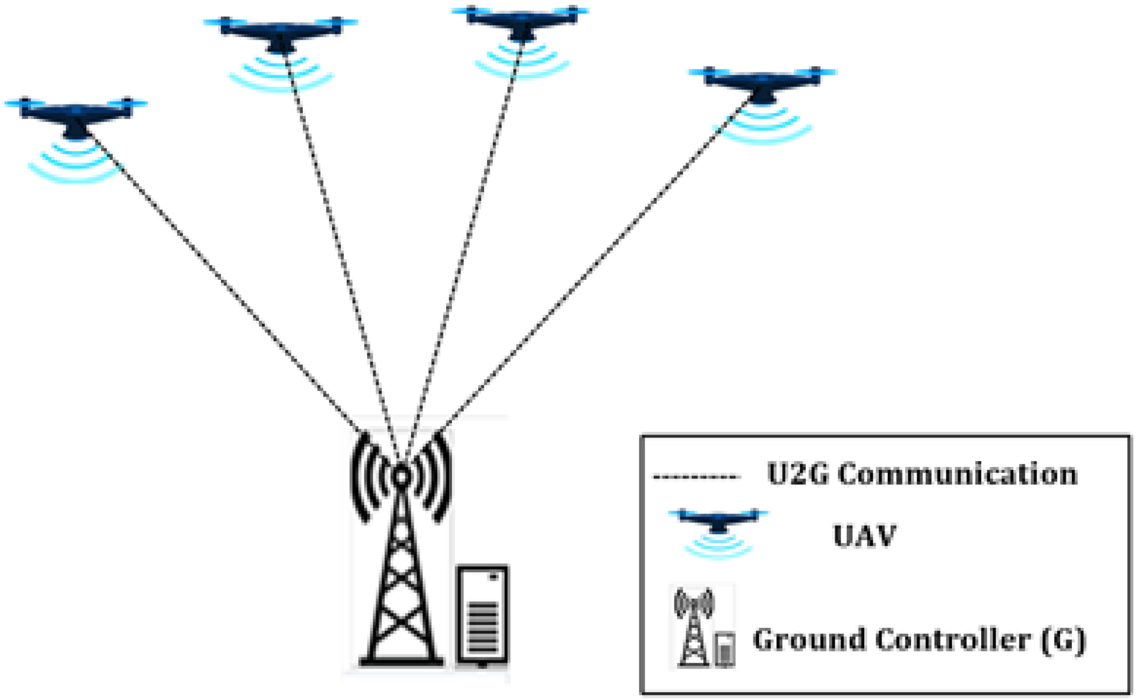


Figure 2.3. Star Topology [8]

- **Ring topology :**

  Drones are connected in a loop, each linked to two neighbors. It supports alternative routing in case of link failure, but is less scalable and becomes complex as the network grows [14].

Figure 2.4 shows :Ring topology.

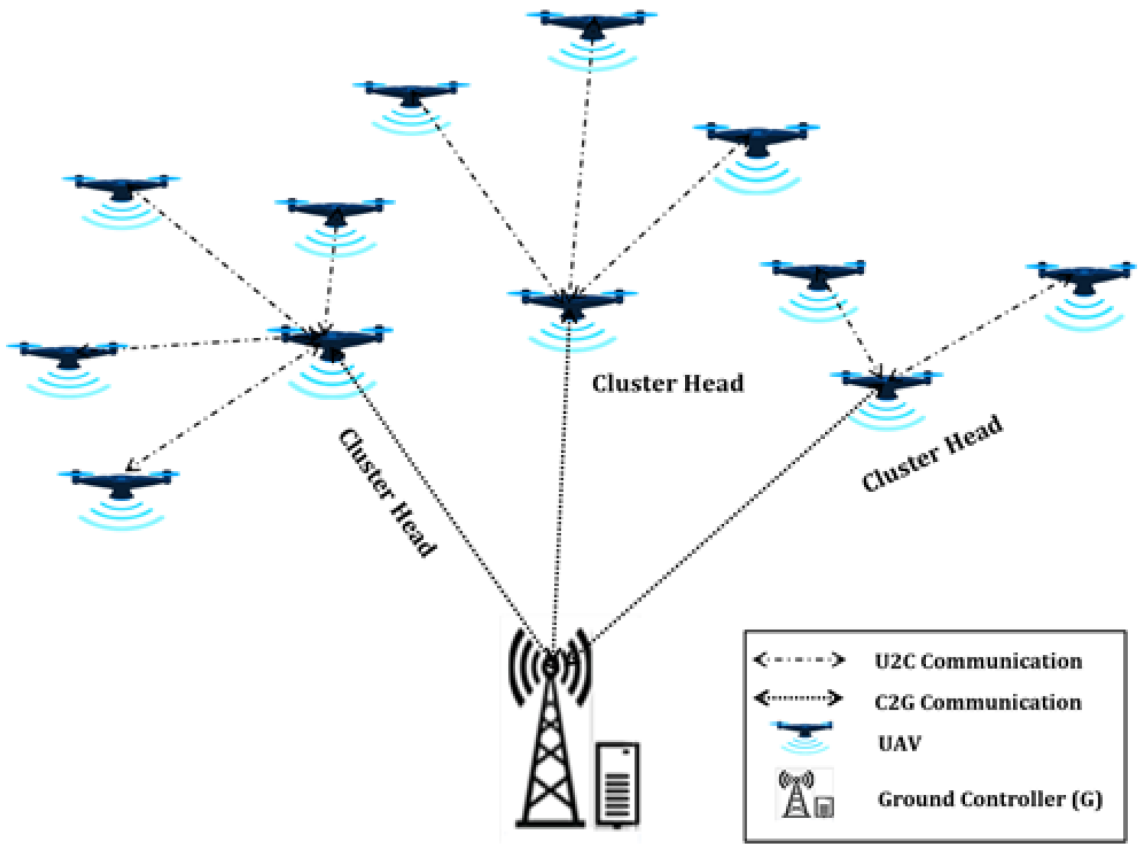


Figure 2.4. Ring topology [8]

- **Mesh Topology :**

  Each drone connects with multiple others, creating a robust and scalable network. It allows dynamic routing and fault tolerance, making it ideal for large, distributed UAV systems [8][14].

Figure 2.5 shows : Mesh Topology

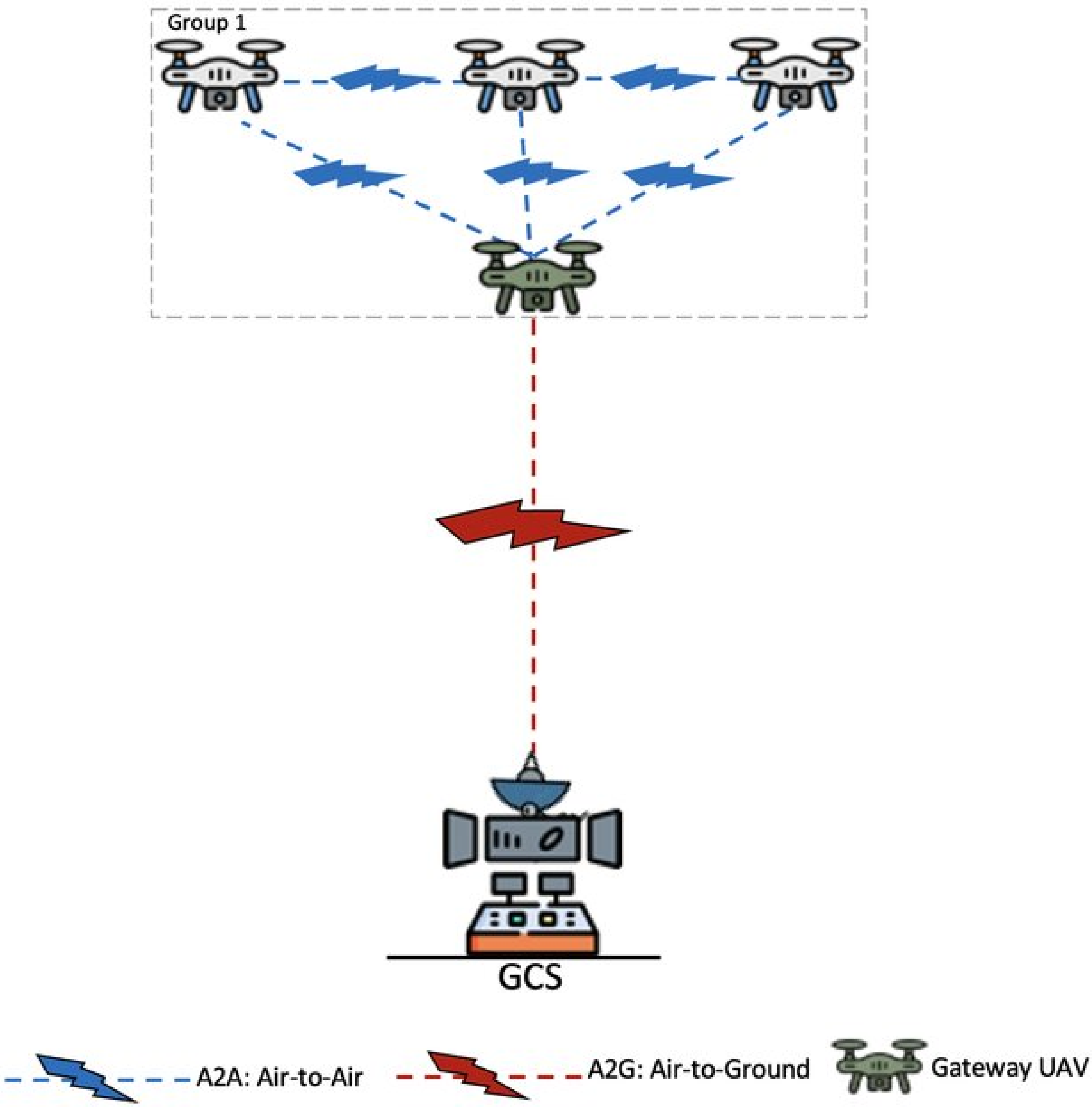


Figure 2.5. Mesh Topology [1]

- **Chosen Topology: Star**

  We adopted a Star Topology where all drones communicate with the ground station. This structure is simple and suits our current setup.

  If future improvements allow for multiple connections between drones, we plan to move toward a Mesh Topology to improve coordination and flexibility.

2.4.2 Communication Protocols for UAV-Ground Station Interaction

Communication protocols are essential for ensuring efficient, timely, and accurate exchange of data between unmanned aerial vehicles (UAVs) and the ground control station (GCS). They support telemetry, command, control, and payload data transmission. The most commonly used protocols in UAV systems include:

- **TCP (Transmission Control Protocol):** A connection-oriented protocol that ensures reliable data transmission through acknowledgments and retransmissions. Although highly reliable, its latency and overhead make it less suitable for real-time UAV applications [47].

- **UDP (User Datagram Protocol):** A connectionless protocol known for low latency and minimal overhead. It does not guarantee delivery, making it ideal for real-time and time-sensitive UAV operations [38].
- **MAVLink (Micro Air Vehicle Link):** A lightweight, open-source protocol developed for communication between UAVs and ground stations. It supports a broad range of message types, including telemetry, position, command, and status information [5, 20].
- **DDS (Data Distribution Service):** A real-time middleware protocol based on a publish/subscribe architecture, suitable for large-scale autonomous UAV networks requiring scalability and Quality of Service (QoS) [22].
- **ROSLink:** An open-source communication protocol designed to connect ROS-based robotic systems to external networks, whether through a local area network (LAN) or the Internet (WAN). It uses JSON-formatted messages transmitted over TCP or UDP, making it lightweight and easy to integrate into various applications [45].
- **RTP over UDP (Real-Time Transport Protocol):** RTP over UDP is a transport protocol used for transmitting real-time video streams over IP networks. It adds packet sequencing and timestamping to ensure synchronized and low-latency delivery, making it suitable for continuous video transmission and real-time visual monitoring [53, 49].

Table 2.1 presents our structured comparison of communication protocols used in UAV systems, based on relevant technical attributes.

Feature	**TCP [47]**	**UDP [38]**	**MAVLink [5, 20]**	**DDS [22]**	**ROSLink [45]**	**RTP over UDP [53, 49]**
Reliability	Reliable	Unreliable	Unreliable (optional ACK)	Reliable	Optional ACK	Unreliable
Latency	High	Very Low	Low (UDP), High (TCP)	Low to Medium	Medium	Very Low
Protocol Overhead	High	Very Low	Very Low	Medium	Medium	Low
Real-time Video	Not suitable	Suitable	Not suitable	Possible	Not suitable	Highly suitable
Security Support	Yes	No	Yes	Yes	Yes	No
Real-time Control	Not suitable	Suitable	Highly suitable	Suitable	Medium	Not suitable
Integration Complexity	Medium	Low	Low	High	Medium	Low
UAV-specific Messages	No	No	Yes	Partial	No	No

Table 2.1. Comparison of UAV Communication Protocols

Chosen Protocols for the Project: For this project, which employs a **Star Topology** where UAVs communicate only via a central ground station, the following protocols are selected:

- **UDP** is used for data transmission between each UAV and the ground station due to its low-latency characteristics, which are critical in real-time systems.
- **MAVLink** is used on the UAV side to interpret and structure the received data. It enables standardized and reliable communication between the onboard autopilot system and other subsystems.
- **RTP over UDP** is used for transmitting real-time video streams from UAVs to the ground station. Its packet sequencing and timestamping ensure synchronized, low-latency video transmission, enabling effective real-time visual monitoring at the ground station.

This combination provides a balance between performance and interoperability: **UDP** ensures fast communication, **MAVLink** provides a structured protocol for telemetry and commands, and **RTP** supports continuous low-latency video streaming essential for autonomous UAV operations.

2.4.3 Coordination

In multi-UAV systems, coordination plays a crucial role in ensuring efficient and mission-oriented operation. As the number of drones in the airspace increases, managing their paths, commands, and interactions becomes more complex. Thus, coordination strategies are designed to organize UAV behaviors and ensure synchronized task execution.

2.4.3.1 Coordination Models

- **Centralized Coordination:**

  In centralized approaches, a ground control station (GCS) or a central decision-making unit plans and manages the actions of all UAVs. This model simplifies decision-making and enables global optimization of tasks. However, it introduces a single point of failure and limits scalability, as all drones depend on the central controller for real-time guidance and updates [9, 35].

- **Distributed Coordination:**

  Distributed strategies allow each UAV to make decisions based on its local environment and received information from neighboring UAVs. This model enhances robustness and scalability, particularly in large or dynamic environments. However, it can suffer from inconsistencies and delays if not properly synchronized [70, 35].

- **Hybrid Coordination:**

  Hybrid systems combine centralized planning with decentralized execution. The GCS assigns high-level objectives, while UAVs independently handle local decision-making. This balance offers improved scalability and reliability, particularly in partially connected or intermittently available networks [13, 35].

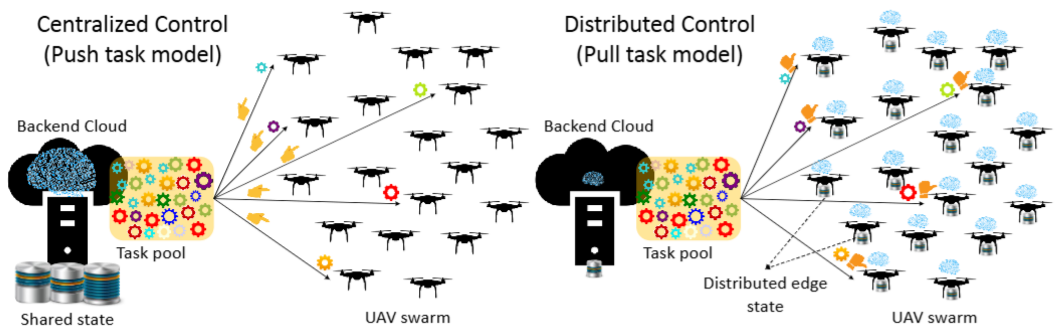


Figure 2.6. Comparison of Centralized and Distributed Coordination in UAV Swarms [35]

Table 2.2 presents our comparison of centralized, distributed, and hybrid coordination approaches in multi-UAV systems.

Feature	**Centralized Coordination [9, 35]**	**Distributed Coordination [70, 35]**	**Hybrid Coordination [13, 35]**
Control Location	Central Ground Control Station (GCS) or cloud	Individual UAVs	GCS for high-level tasks, UAVs for local execution
Decision-Making	Global, centralized decisions	Local, individual UAV decisions	Mixed: Centralized planning with local autonomy
Scalability	Limited due to centralized bottlenecks	High, suitable for large-scale swarms	Moderate to high scalability
Robustness	Low: Single point of failure	High: No central dependency	Balanced and fault-tolerant
Communication Overhead	High	Low to moderate	Moderate
Latency	May be high due to centralized processing	Low, decisions made locally	Depends on task division
Best Used In	Small swarms with strong connectivity	Large, dynamic environments	Environments with partial connectivity or hybrid needs

Table 2.2. Comparison Between Centralized, Distributed, and Hybrid UAV Coordination Models

2.4.3.2 Chosen Approach for This Project

This project adopts a hybrid control approach, where drones operate autonomously to carry out pollination tasks, while the ground station retains a supervisory role and intervenes when necessary. This strategy aims to balance operational autonomy with centralized oversight to ensure efficiency and precision in task execution.

2.5 Spatial Awareness in Drones

Unmanned Aerial Vehicles (UAVs), commonly known as drones, rely on various technologies to achieve spatial awareness, enabling autonomous navigation and operation in diverse environments. This section outlines the primary technologies facilitating this capability.

2.5.1 Global Positioning System (GPS)

GPS technology allows drones to determine their precise geographical location by connecting to a network of satellites. This capability is fundamental for navigation, flight stabilization, and executing pre-defined flight paths [40].

2.5.2 Visual Navigation

Visual Navigation (VNav) systems utilize onboard cameras and computer vision algorithms to interpret the drone's surroundings, enabling navigation without reliance on GPS signals. This approach is particularly useful in GPS-denied environments [61].

2.5.3 Simultaneous Localization and Mapping (SLAM)

SLAM technology enables drones to construct a map of an unknown environment while simultaneously keeping track of their location within it. This is achieved through the integration of sensor data, such as from cameras and inertial measurement units, allowing for real-time 3D mapping and navigation [21].

2.5.4 Light Detection and Ranging (LiDAR)

LiDAR systems emit laser pulses to measure distances to surrounding objects, creating detailed 3D maps of the environment. Drones equipped with LiDAR can perform high-precision mapping and obstacle detection, even in low-light conditions [15].

2.5.5 Radio Frequency (RF) Positioning

RF-based positioning systems enable drones to determine their location by analyzing radio wave reflections. This method is effective in environments where GPS signals are unavailable or unreliable, such as indoors or underground [59].

2.5.6 Artificial Intelligence (AI) and Deep Learning

AI and deep learning algorithms enhance a drone's ability to interpret complex environments, make decisions, and adapt to dynamic conditions. These technologies improve object recognition, path planning, and autonomous decision-making processes [65].

2.5.7 Sensor Fusion

Sensor fusion involves combining data from multiple sensors, such as GPS, LiDAR, cameras, and inertial measurement units, to achieve more accurate and reliable navigation information. This integrated approach enhances the drone's situational awareness and operational robustness [80].

2.6 Conclusion

This chapter provided a comprehensive overview of the key technologies that enable autonomous drone-based pollination. It addressed object detection methods, quadrotor control strategies, and the communication protocols that ensure efficient and reliable interaction between drones and the ground station. These elements are essential for achieving precise navigation, stable flight, and coordinated operation in agricultural environments. The next chapter focuses on the architectural design of the proposed system, including its structure, components, and functional logic.

Chapter 3

System Design

3.1 Introduction

In this chapter, we present the design and development process of our autonomous agricultural drone. The goal was to create a low-cost system capable of performing tree detection and precise spraying using onboard sensors and intelligent control. The system combines hardware components such as Raspberry Pi, Pixracer, sensors, motors, and a water pump, all working together to form a functional drone that can be deployed for real-time tasks. The following sections detail the system architecture and the integration of each module.

3.2 Objectives and Principle of the Proposed Solution

This section presents the architectural and structural design of the proposed system. It includes the use case diagram, class diagram, and the core algorithms involved in the system implementation.

3.2.1 Principle of the Proposed Solution

Figure 3.1 presents the conceptual overview and architectural workflow of the system. The process consists of six main steps:

1. **Farm Area Selection:** The user defines the location and boundaries of the palm tree farm through the system interface.
2. **Zoning:** The system automatically divides the farm into multiple zones to distribute the workload efficiently among the available drones.
3. **Drone Deployment:** Each drone is assigned to a specific zone and receives navigation and mission instructions from the system.
4. **Palm Tree Detection:** Within its designated zone, each drone uses a YOLOv12-based object detection model to identify palm trees that require pollination.
5. **Pollination:** After detecting the target trees, the drone autonomously performs the pollination task within its assigned zone.
6. **Completion Feedback:** Upon mission completion, each drone reports its status back to the system for real-time monitoring and verification.

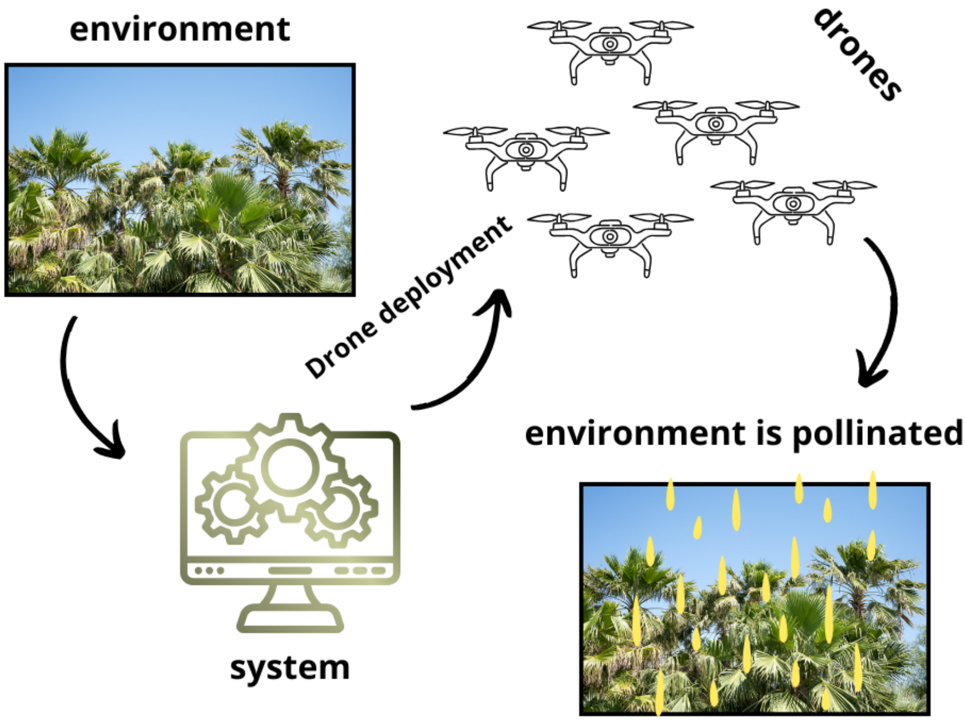


Figure 3.1. Conceptual overview of the proposed solution

To complement the conceptual design, Figure 3.2 illustrates the operational workflow of the proposed system. It visually demonstrates the steps of drone deployment, zone navigation, palm tree detection, and execution of pollination in the field.

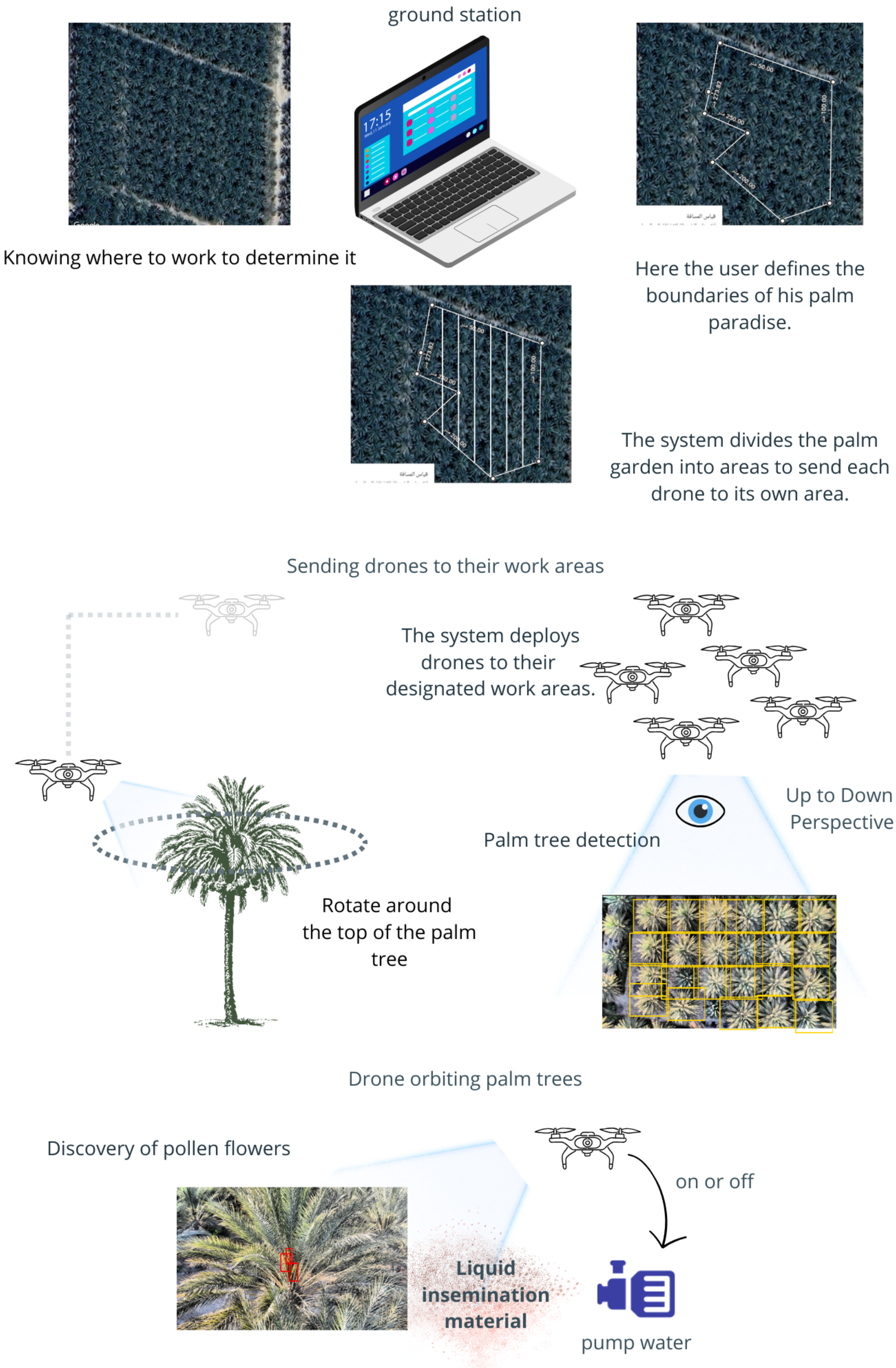


Figure 3.2. Operational workflow of the proposed system

3.2.2 System Development Phases

Figure 3.3 outlines the main development stages of the proposed system. These stages reflect the transition from hardware specification to complete software integration:

1. **Hardware Specification:** Selection of core components based on functional requirements, including the flight controller, motors, GPS, camera, and various sensors.
2. **Hardware Assembly and Integration:** Physical integration and testing of the components to ensure proper communication between hardware modules, especially between the Raspberry Pi and Pixracer.
3. **Software Development:** This phase involves two key aspects:
   - **AI Module Integration:** Implementation and deployment of palm tree detection using the YOLOv12 algorithm.
   - **Communication Module:** Configuration of communication protocols such as UDP and MAVLink to enable real-time data exchange between drones and the ground station.

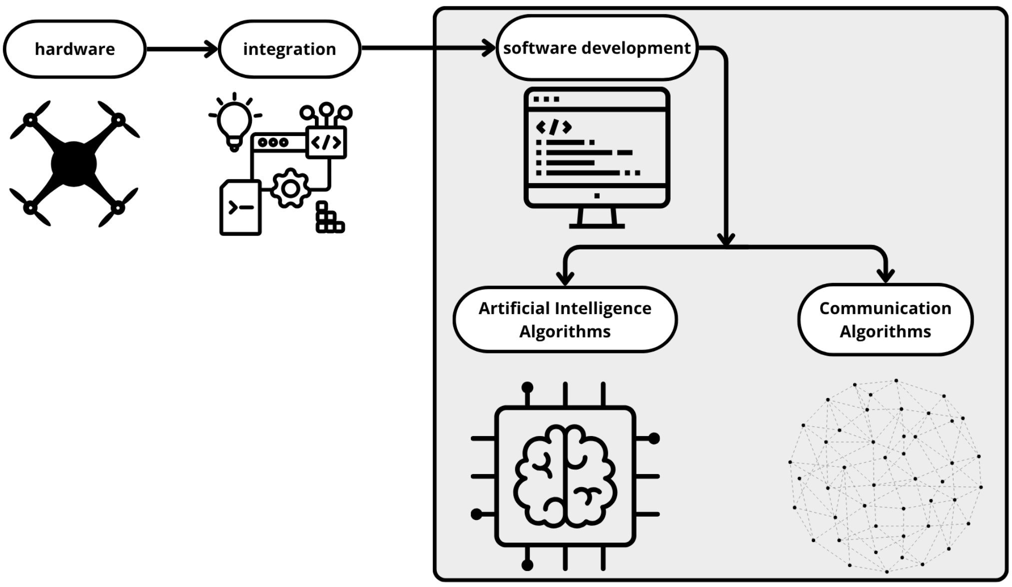


Figure 3.3. System development phases

3.2.3 Communication Protocols

The communication framework of the proposed system, shown in Figure 3.4, is designed to support real-time telemetry, coordination, and video streaming between the drones and the ground station using both external and internal communication layers.

Each drone includes a Raspberry Pi 4B, which handles mission execution and communication. Externally, it communicates with the ground station over Wi-Fi using UDP for low-latency status updates and RTP for streaming live camera feeds. This ensures that the ground station can receive frequent updates and visual feedback during pollination.

Internally, the Raspberry Pi interfaces with the Pixracer flight controller using the MAVLink protocol via a direct USB (virtual serial) connection. This link supports reliable transmission of telemetry data, GPS readings, and control commands such as arming and return-to-home.

This layered communication architecture maintains a clear separation between onboard control and centralized supervision, ensuring responsiveness, modularity, and robustness during autonomous operation.

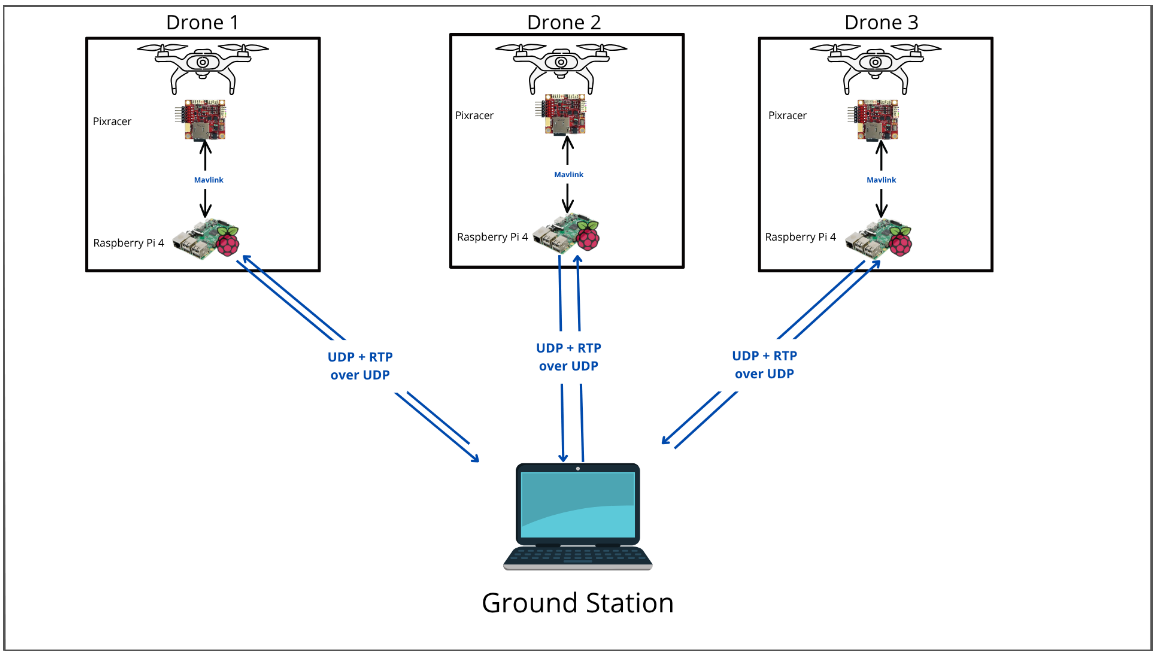


Figure 3.4. Communication architecture using UDP, MAVLink, and RTP protocols

3.3 System Design Description Using UML Diagrams

This section outlines the main functionalities of the Smart Pollination System through UML diagrams.

3.3.1 Use case diagram

The interaction between the primary actors namely the Operator and the Farmer and the system's core functionalities during the pollination process is illustrated in Figure 3.5.

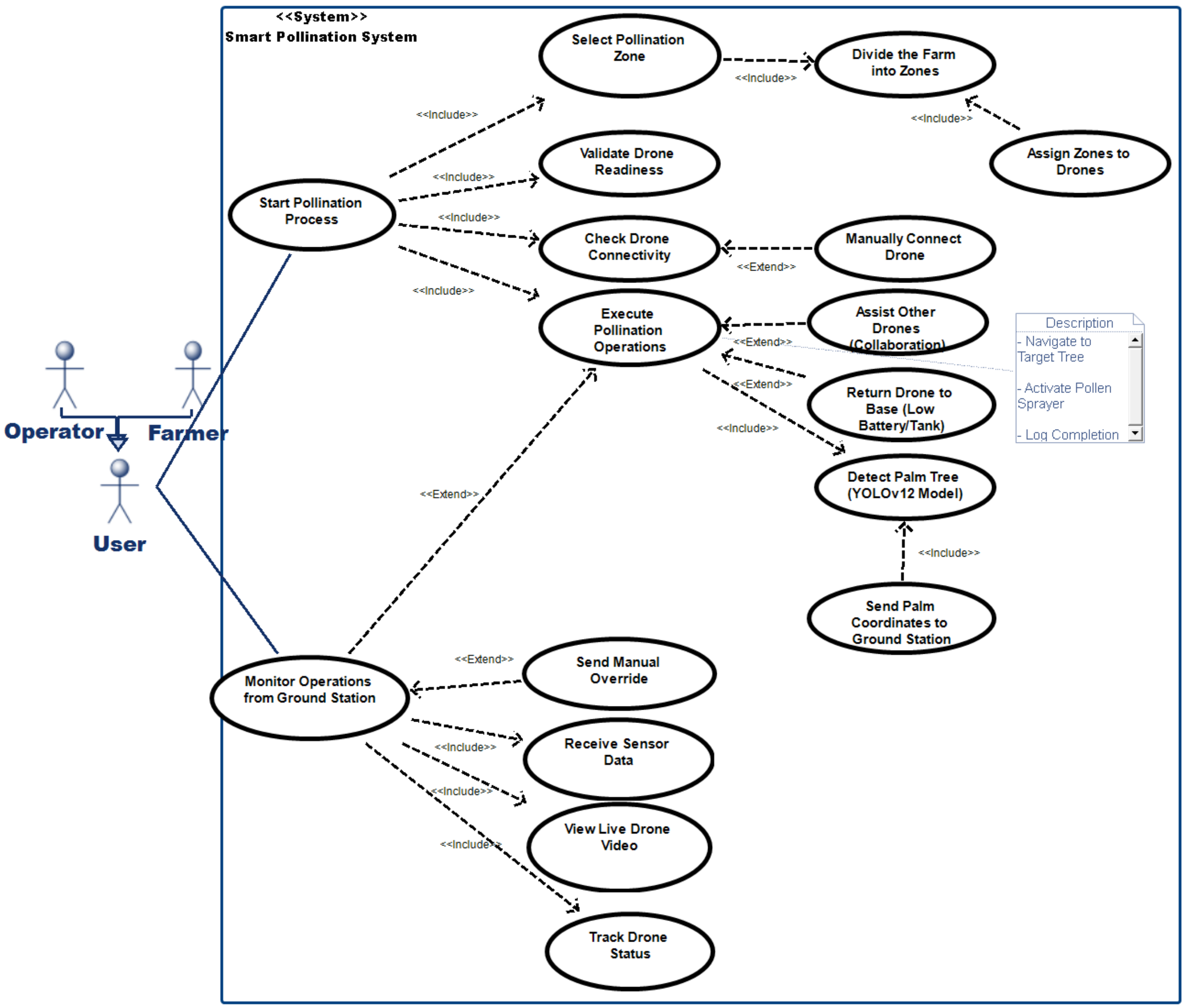


Figure 3.5. Use case diagram of the Smart Pollination System

Table 3.1 outlines each functionality of the use case diagram with its description.

Functionality	Description
Start Pollination Process	Launches the pollination workflow after ensuring all drones are ready and zones are assigned.
Select Pollination Zone	The user selects the target agricultural zone to be pollinated.
Validate Drone Readiness	Ensures drones are powered, connected, and capable of completing the task.
Check Drone Connectivity	Confirms real-time communication between drones and the ground station.
Manually Connect Drone	Allows manual pairing of drones with the system in case of failed automatic detection.
Divide the Farm into Zones	Automatically segments the selected area into manageable zones for assignment.
Assign Zones to Drones	Allocates segmented zones to available drones for pollination.
Execute Pollination Operations	Each drone performs pollination based on the assigned zone and task.
Assist Other Drones (Collaboration)	Enables idle drones to assist others in case some zones are unfinished.
Return Drone to Base (Low Battery/Tank)	Drones automatically return when battery or pollen tank is low.
Detect Palm Tree (YOLOv12 Model)	Uses onboard AI model to detect palm trees in real-time video.
Send Palm Coordinates to Ground Station	Sends the detected palm tree coordinates to the ground station for monitoring.
Monitor Operations from Ground Station	Allows real-time monitoring of drone activities, video, and data.
Send Manual Override	Sends direct control commands from the operator to drones during emergency.
Receive Sensor Data	Collects telemetry from drones (e.g., battery level, tank status, GPS).
View Live Drone Video	Streams the onboard video feed to the ground station interface.
Track Drone Status	Displays drone location, task progress, and operational metrics.

Table 3.1. Functionalities of the Smart Pollination System

3.3.2 Class diagram

Figure 3.6 illustrates the static structure of the smart palm tree pollination system. It defines key components such as drones, the ground station, communication modules, and sensors, along with their attributes and relationships. This representation helps to understand how the system's objects interact and how responsibilities are distributed.

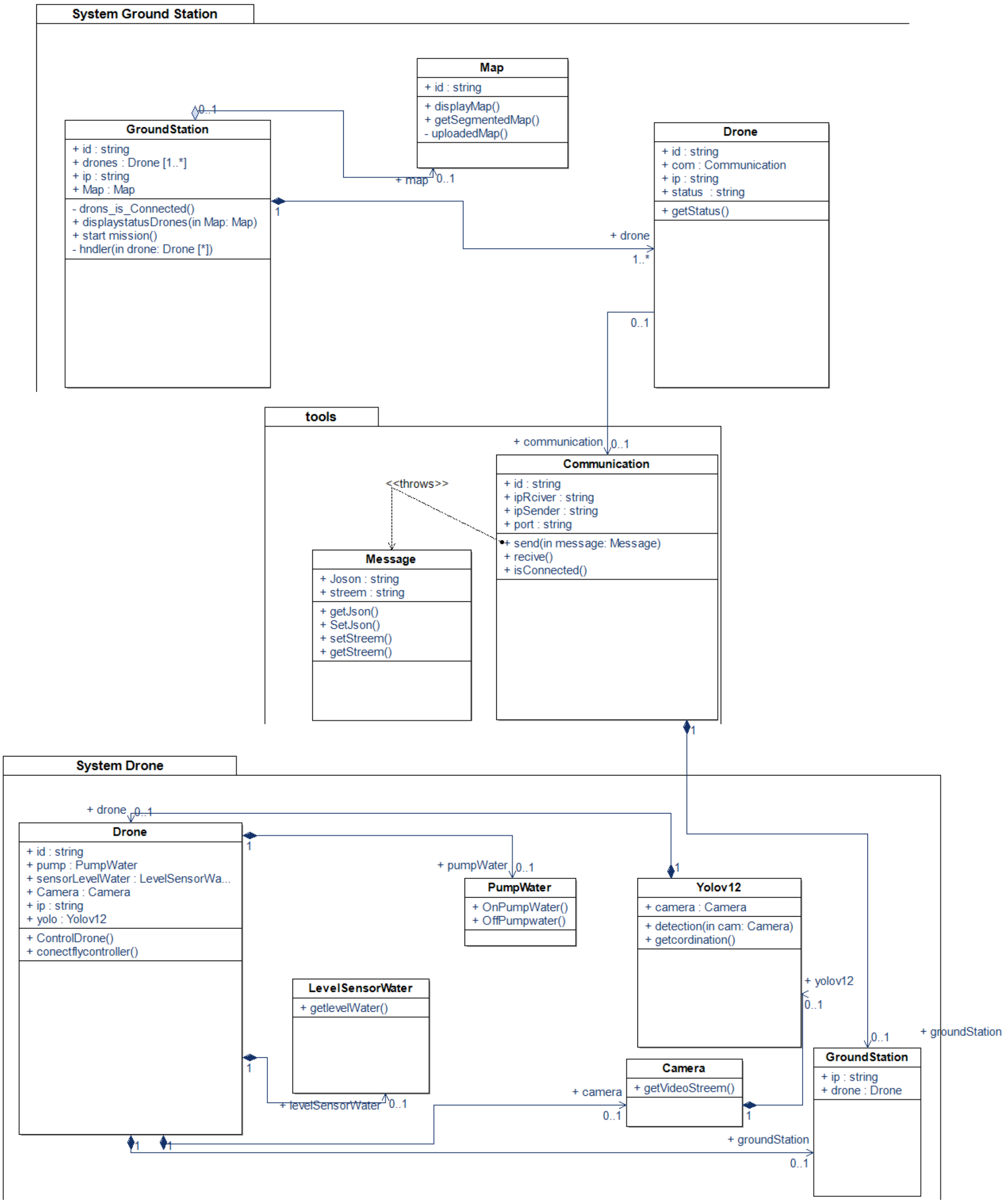


Figure 3.6. Class diagram

Table 3.2 outlines each class of the system along with its corresponding description.

Functionality	**Description**
Map	Handles display and upload of the agricultural map. Provides segmentation functionalities.
GroundStation	Central system that manages drone connectivity, assigns tasks, displays drone statuses, and starts missions.
Drone	Represents an autonomous drone. Stores ID, IP, communication data, and status. Retrieves its operational status.
Communication	Handles data transmission between drones and ground station using UDP. Sends and receives messages.
Message	Encapsulates messages exchanged between drones and ground station, including video stream and telemetry in JSON.
PumpWater	Controls the water pump used for pollination. Includes methods to start and stop pumping.
LevelSensorWater	Monitors the water tank level and provides status updates.
Camera	Captures real-time video streams for palm detection and transmission to the ground station.
YoloV12	Embedded AI model on the drone. Detects palm trees using real-time video frames from the camera.

Table 3.2. Class Descriptions of the Autonomous Drone Pollination System

3.3.3 Activity Diagram

Figure 3.7 illustrates the dynamic workflow of the system, covering stages from initialization to mission completion. It details key processes such as map uploading, drone connection, mission execution, and error handling. This visual representation provides insight into how the system behaves in real-time during a pollination task.

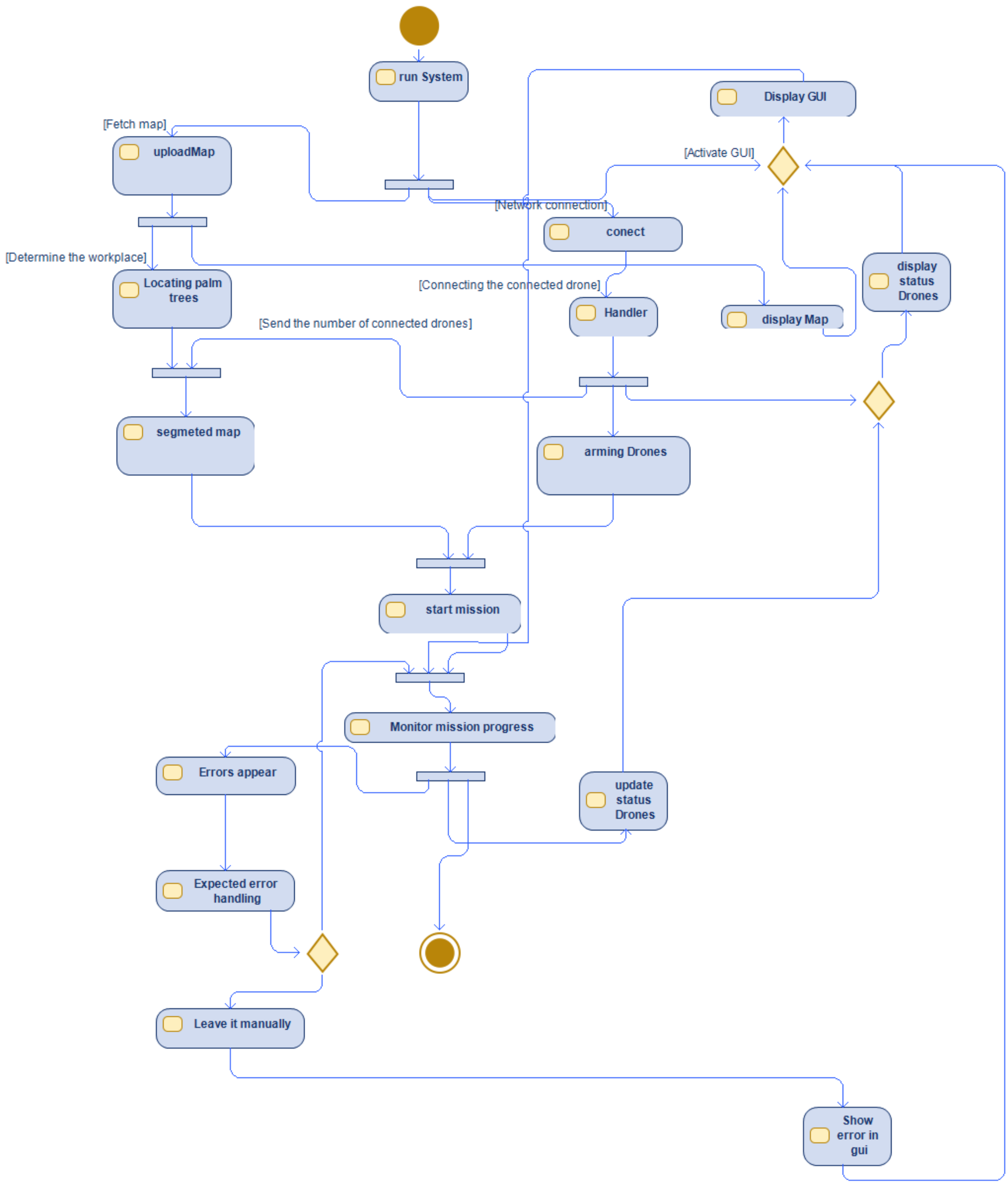


Figure 3.7. Activity diagram of the smart pollination system

According to the activity diagram, Table 3.3 presents the key activities involved in the system workflow, along with their corresponding descriptions.

Functionality	**Description**
run System	Initializes the pollination system including GUI, network, and map loading.
uploadMap	Loads the selected agricultural map into the system.
Locating palm trees	Identifies palm tree positions from the map or video feed.
segmented map	Divides the agricultural map into zones for drone assignment.
connect	Establishes the network connection between drones and ground station.
Handler	Detects and manages newly connected drones.
arming Drones	Prepares and arms all connected drones for mission launch.
start mission	Launches the pollination operation after drones are assigned and ready.
Monitor mission progress	Continuously checks drone progress, status, and task completion.
update status Drones	Periodically updates and displays the status of all drones in GUI.
Display GUI	Shows the graphical interface for user interaction and monitoring.
display Map	Displays the uploaded agricultural map in the GUI.
display status Drones	Displays the current status and metrics of all drones in GUI.
Errors appear	Handles system or mission execution errors during operations.
Expected error handling	Executes pre-defined recovery procedures for common errors.
Show error in gui	Displays any error messages on the user interface for troubleshooting.
Leave it manually	Allows user to terminate the mission manually if required.

Table 3.3. Activity Diagram Descriptions for Drone Pollination System

3.4 Algorithms

This section outlines the main algorithms that drive the core functionalities of the system. These algorithms include task distribution, drone coordination, real-time monitoring, and return-to-base decision-making. Each algorithm is designed to ensure optimal performance, reliability, and autonomy of the system during operation.

3.4.1 Drones Control Algorithm

This subsection outlines the control algorithms used to manage drone arming, disarming, manual movement, and navigation around detected palm trees. These algorithms rely on MAVLink protocol commands and coordinate-based control.

3.4.1.1 Arm and Disarm Control

The following algorithms handle motor arming and disarming via MAVLink messages.

Algorithm 1: Drone Arm Control Algorithm

Input: None
Output: `status` (String) – confirmation message after arming
Send MAVLink arm command
```
command_long_send(target_system, target_component,
  MAV_CMD_COMPONENT_ARM_DISARM, 0, 1, 0, 0, 0, 0, 0, 0)
```
return "Armed"

Algorithm 2: Drone Disarm Control Algorithm

Input: None
Output: `status` (String) – confirmation message after disarming
Send MAVLink disarm command
```
command_long_send(target_system, target_component,
  MAV_CMD_COMPONENT_ARM_DISARM, 0, 0, 0, 0, 0, 0, 0, 0)
```
return "Disarmed"

3.4.1.2 Manual Drone Control

This algorithm sends axis-based velocity commands to manually control the drone's motion using bounded input values.

Algorithm 3: Manual Drone Control Algorithm

Input: x, y, thrust, yaw (int)
Output: status (String)
Limit control values:
$x \leftarrow \max(-1000, \min(1000, x))$
$y \leftarrow \max(-1000, \min(1000, y))$
$thrust \leftarrow \max(0, \min(1000, thrust))$
$yaw \leftarrow \max(-1000, \min(1000, yaw))$
Send control command:
`manual_control_send(target_system, x, y, thrust, yaw, 0)`
return "Control sent"

3.4.1.3 Palm Detection and Navigation

The following set of algorithms manage how the drone reacts to palm tree detections, decides if a palm has already been visited, moves toward the target, and performs a circling maneuver for pollination.

Algorithm 4: Is New Palm Check

Input: xpc, ypc (float), threshold (float)
Output: Boolean
foreach (x_{prev}, y_{prev}) *in visited_palms* **do**
  $dist \leftarrow \sqrt{(xpc - x_{prev})^2 + (ypc - y_{prev})^2}$
  if $dist < threshold$ **then**
    return False
  end
end
return True

Algorithm 5: Go to Target Point

Input: xp, yp, thrust, yaw (float/int), interval, tolerance, gain (float)
Output: None
while *True* **do**
  $dx \leftarrow xp - xc$
  $dy \leftarrow yp - yc$
  if $|dx| < tolerance$ ***and*** $|dy| < tolerance$ **then**
    break
  end
  $x_{control} \leftarrow \text{int}(-gain \times dx)$
  $y_{control} \leftarrow \text{int}(-gain \times dy)$
  `manual_control_send(target_system,` $x_{control}, y_{control}, thrust, yaw, 0)$
  Wait interval seconds
end

Algorithm 6: Circle Around Palm Tree

```
Input: xpc, ypc, radius, steps, thrust, interval
Output: None
for i ← 0 to steps − 1 do
    angle ← (2π·i) / steps
    xp ← xpc + radius · cos(angle)
    yp ← ypc + radius · sin(angle)
    go_to_point(xp, yp, thrust)
    Wait interval seconds
end
Add (xpc, ypc) to visited_palms
```

Algorithm 7: Process Detected Palms

```
Input: detections (list)
Output: None
foreach (xpc, ypc, w, h, conf) in detections do
    xp ← xpc − w/2
    yp ← ypc
    if is_new_palm(xpc, ypc) then
        go_to_point(xp, yp)
        circle_around_palm(xpc, ypc, w/2)
    end
end
```

3.4.2 Pump Activation Control Algorithm

This algorithm activates or deactivates the pump responsible for releasing pollen, depending on the control signal received.

Algorithm 8: Pump Control Algorithm

```
Input: signal (Boolean)
Output: pumpStatus (Boolean)
if signal == true then
    activate_pump()
    pumpStatus ← ON
end
else
    deactivate_pump()
    pumpStatus ← OFF
end
return pumpStatus
```

3.4.3 Water Sensor Detection Algorithm

This algorithm checks the state of the drone's water tank using a GPIO-based sensor input to determine if water is present.

Algorithm 9: Water Sensor Detection Algorithm

```
Input: pin (Integer)
Output: waterDetected (Boolean)
GPIO.setmode(GPIO.BCM)
GPIO.setup(pin, GPIO.IN, pull_up_down=GPIO.PUD_DOWN)
readValue ← GPIO.input(pin)
if readValue == GPIO.HIGH then
    waterDetected ← true
end
else
    waterDetected ← false
end
return waterDetected
```

3.4.4 Camera Streaming Algorithm

This algorithm handles the real-time video streaming from the drone's onboard camera using MJPEG format.

Algorithm 10: Camera Streaming Algorithm

```
Input: None
Output: JPEG frame stream
Initialize picam2 as new Picamera2 instance
Set frame size and format
Configure and start camera
while True do
    frame ← Capture frame
    buffer ← Encode to JPEG
    jpegFrame ← Convert to bytes
    Yield:
    -frame˚|
    Content-Type: image/jpeg˚˚|
    jpegFrame
    ˚|
end
```

3.5 Conclusion

This chapter introduced the structural and architectural design of the smart palm tree pollination system. It covered the core components, communication setup, UML diagrams, and functional logic that define how the system operates conceptually. These design elements provide a clear blueprint for system behavior and coordination. The next chapter moves from theory to practice by detailing the actual implementation, hardware integration, algorithm deployment, and real-world testing of the proposed solution.

Chapter 4

Implementation and Testing

4.1 Introduction

This chapter outlines the key stages of implementing our autonomous drone system for palm tree pollination. It briefly presents the system architecture, hardware and software components, AI integration, and the desktop control interface. The dataset preparation, model training, and evaluation results are also highlighted, along with the main technical challenges encountered during development.

4.2 System Architecture

The system architecture is composed of multiple hardware components connected to ensure the autonomous operation of the drone. The Raspberry Pi 4 Model B serves as the main processing unit responsible for running object detection and tree recognition algorithms. It communicates with Pixracer R15, which is used for flight control. The drone is equipped with four brushless motors controlled by ESCs and powered by an 11.1V Li-Po battery connected via a power distribution board.

A logic-level converter ensures safe communication between the Raspberry Pi and 5V peripherals such as the distance sensor (ultrasonic), water level sensor, servo motor, and relay module. The relay controls a 6V DC water pump used for spraying, powered by an independent 9V battery. A converter regulates the voltage from 11.1V to 5V to safely power the Raspberry Pi.

Figure 4.1 shows the overall architecture of the system, including the interconnections between the components and power distribution.

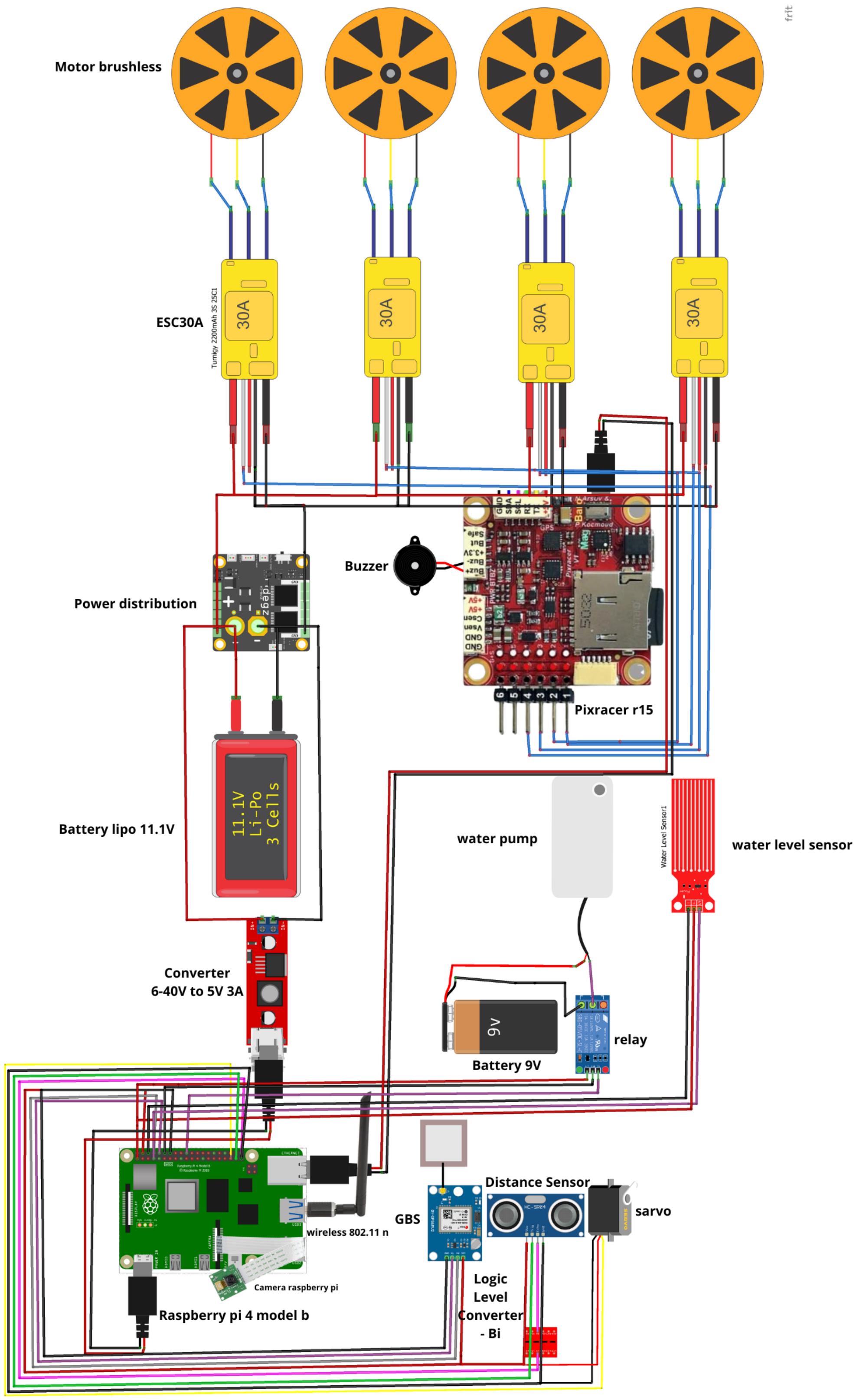


Figure 4.1. System architecture of the autonomous agricultural drone

4.3 Tools and Technologies

In this section, we present the hardware and software tools used in the design and implementation of our autonomous agricultural drone system.

4.3.1 Hardware Components

4.3.1.1 Drone Components

Pixracer R15: A flight control unit designed for small drones. It acts as the central brain that receives data from various onboard sensors (such as IMU, GPS, and compass), interprets user or autonomous commands, and adjusts motor speeds accordingly to maintain stable and controlled flight. It features a compact design and supports open-source software like PX4 and ArduPilot, enabling complex autonomous missions.

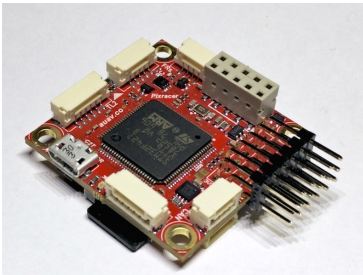


Figure 4.2. Pixracer R15

Brushless Motors (x4): These motors provide the necessary thrust for flight. They are known for their efficiency and high torque-to-weight ratio, making them ideal for drones. They are controlled via ESCs.

Figure 4.3. Brushless Motors

Propellers: Crucial parts that create lift for flying by converting rotational motion into linear thrust.

Figure 4.4. Drone Propellers

Electronic Speed Controller (ESC 30A) (x4): ESCs are used to control the rotation speed of brushless motors. They receive PWM signals from the Pixracer flight controller and convert them into appropriate current to drive the motors. A 30A capacity is suitable to handle the required current during flight.

Figure 4.5. ESC 40A

Power Distribution Board (3DR): The 3DR Power Distribution Board distributes power from the Li-Po battery to the drone's components such as ESCs and the flight controller. It helps organize connections and ensures stable voltage delivery.

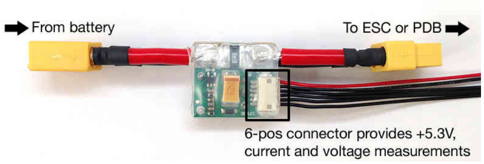


Figure 4.6. Power Distribution Board (3DR)

Li-Po Battery 11.1V 3S 3300mAh 35C: This battery serves as the main power source for the drone. It operates at a nominal voltage of 11.1V (3 cells in series), with a capacity of 3300mAh and a discharge rate of 35C. This configuration provides sufficient energy and current to power the motors, flight controller, and other onboard components. Li-Po batteries with high C-ratings like this one are ideal for drones due to their ability to deliver high bursts of current while maintaining a lightweight form factor.

Figure 4.7. Li-Po Battery 11.1V 3S 3300mAh 35C

Buzzer and Safety Button: The buzzer emits sound alerts for various states such as startup, errors, or task completion, while the safety button is a manual emergency switch used to instantly cut off power to the motors for safety.

Figure 4.8. Buzzer and Safety Button

Frame f330 : The F330 is a compact and robust quadcopter frame made of glass fiber and nylon. It serves as the structural base for mounting all drone components and includes pre-designed paths for electrical wiring and power distribution.

Figure 4.9. frame f330

4.3.1.2 IoT and Smart Control Components

Raspberry Pi 4 Model B: The Raspberry Pi 4 is used as a high-level controller for image processing and decision-making based on AI algorithms. It also manages communication between other components and connects to the Pixracer unit via the MAVLink protocol. It sends and receives data from sensors and actuators through GPIO ports.

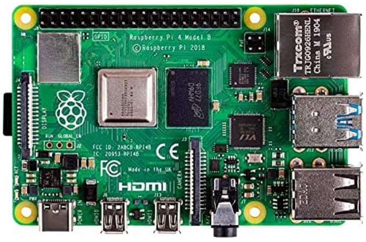


Figure 4.10. Raspberry Pi 4 Model B

DC-DC Converter (12V to 5V 3A): This converter is used to step down the high voltage from the battery to a stable 5V to power the Raspberry Pi and other sensitive components. It ensures safe and reliable power supply.

Figure 4.11. DC-DC Converter

Water Pump (6V): This pump is used to spray the pollination solution onto the flowers , and is controlled by the Relay module and Raspberry Pi.

Figure 4.12. 6V Water Pump

Relay Module: The relay module acts as an electrical intermediary, allowing the Raspberry Pi to control the operation of the pump without exposing the low-voltage computer directly to high voltage.

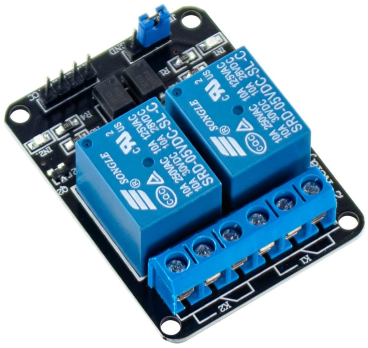


Figure 4.13. Relay Module

GPS GT U7 : This module provides the drone with geographic location data, mainly used for navigation.

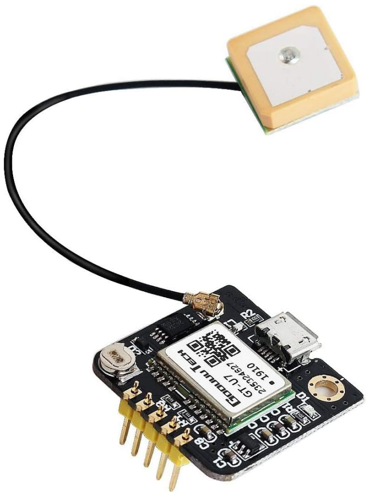


Figure 4.14. GPS GT U7

Logic Level Converter: This converter allows safe interfacing between devices that operate at 3.3V (like the Raspberry Pi) and those that work at 5V, protecting the components from voltage mismatches.

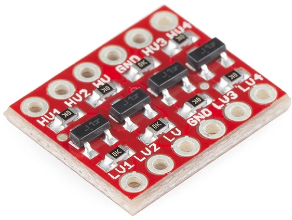


Figure 4.15. Logic Level Converter

Ultrasonic Sensor: This sensor is used to measure general distances between the drone and surrounding objects, such as palm fronds or obstacles, using ultrasonic waves. It supports obstacle detection and assists with navigation and positioning.

Figure 4.16. Ultrasonic Sensor

Water Level Sensor: This sensor is used to detect whether the pollination tank contains liquid. It helps prevent the pump from operating when the tank is empty, protecting the system and extending the pump's lifespan.

Figure 4.17. Water Level Sensor

Servo Motor: The servo motor provides precise motion in the pollination mechanism. It can rotate to specific angles based on commands from the Raspberry Pi, allowing accurate directional control of the spraying action.

Figure 4.18. Servo Motor

USB Wireless Adapter (802.11n): This unit is used to enable Raspberry Pi to connect wirelessly via a local network or the internet using an external Wi-Fi adapter that supports the 802.11n standard. This allows for remote control and monitoring of drone operations.

Figure 4.19. Wireless 802.11n (USB Wi-Fi Adapter)

Camera raspberry pi: This camera is directly connected to the Raspberry Pi and is used for capturing images and video.

Figure 4.20. Camera raspberry pi

Battery 9v: This battery provides an independent power source specifically for the electric pollination pump. It ensures the pump operates reliably without affecting the main power system of the drone.

Figure 4.21. Battery 9v

USB Type-C and USB Micro-B Cables: These cables are used to connect various modules for both power and data transfer. The USB Type-C cable connects the Pixracer to the Raspberry Pi for data communication, while the USB Type-C cable also delivers regulated power from the DC-DC converter to the Raspberry Pi.

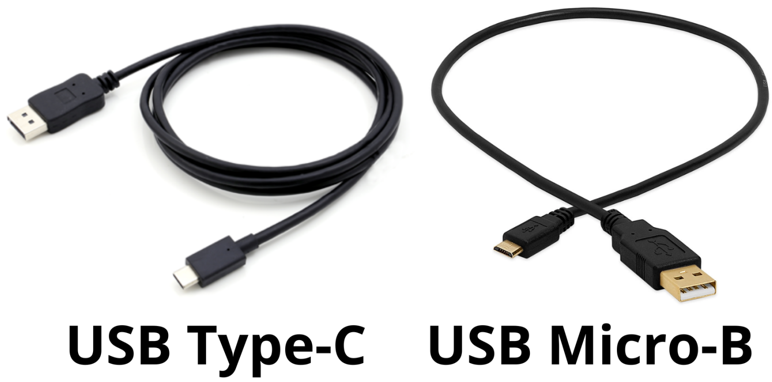


Figure 4.22. USB Type-C and USB Micro-B Cables

4.3.2 Software Components

4.3.2.1 Operating Systems and Tools

Raspberry Pi Imager: An official tool used to flash operating system images onto SD cards.

Figure 4.23. Imager

Mission Planner: A ground control software used to configure, calibrate, and monitor flight controllers like Pixhawk.

Figure 4.24. mission planner

Raspberry Pi OS Lite: A lightweight version of Raspberry Pi OS without a desktop environment, optimized for embedded applications.

Figure 4.25. Raspberry Pi OS Lite

ArduPilot OS: A real-time autopilot firmware that runs on flight controllers like Pixracer. It supports autonomous drone control, including flight stabilization, navigation, and mission execution.

Figure 4.26. ArduPilot System

VS Code IDE: A versatile code editor for writing, debugging, and managing code with extensions support.

Figure 4.27. VS Code IDE

SSH: A secure protocol that allows remote access and control of the Raspberry Pi via the command line.

Figure 4.28. SSH

4.3.2.2 Programming Languages and Libraries

Python 3: A high-level programming language used for sensor control, data processing, and system logic.

Figure 4.29. Python

HTML, CSS, JS: Standard web technologies used to build and style the drone's user interface.

Figure 4.30. HTML, CSS, JS

Flask: A lightweight Python web framework used to develop web-based dashboards and remote control interfaces.

Figure 4.31. Flask

MAVProxy: A command-line ground station tool using MAVLink protocol for communication between Raspberry Pi and Pixracer.

Figure 4.32. MAVProxy

4.3.2.3 Artificial Intelligence Tools and Libraries

Roboflow: A platform for image dataset preparation, labeling, and augmentation to train object detection models.

Figure 4.33. Roboflow

Albumentations: A powerful image augmentation library used to enrich datasets and improve model robustness.

Figure 4.34. Albumentations

OpenCV: An open-source computer vision library for image and video processing tasks like tree detection.

Figure 4.35. OpenCV

YOLO: "You Only Look Once," a real-time object detection model used for identifying trees from drone images.

Figure 4.36. YOLO

Qt Designer: A graphical user interface (GUI) design tool used to create and customize interfaces for applications based on the Qt framework, allowing for drag-and-drop widget layout and UI logic design without manual coding.

Figure 4.37. Qt Designer Icon

4.4 Drone Assembly and Flight Performance Estimation

The components of the drone have been previously listed. After full assembly, the total weight of the drone is approximately **1300 grams**. Each **1000KV** brushless motor is capable of lifting up to **800 grams**, and with four motors, the total theoretical lift is:

$$4 \times 800 = 3200 \text{ grams}$$

To ensure stable flight and account for a safety margin, we divide the total thrust by 2:

$$\frac{3200}{2} = 1600 \text{ grams}$$

Since this value is greater than the actual drone weight (1300g), the configuration is considered safe and adequate for flight.

As for the flight time estimation, the drone is powered by a **3300mAh** 3S (11.1V) LiPo battery rated at **35C**. The maximum current the battery can deliver is calculated as:

$$35 \times 3.3 = 115.5 \text{ Amps}$$

Given that the drone consumes approximately **86 Amps** under full load, the battery can safely supply the required current without exceeding its discharge limit.

The theoretical flight time can be estimated using the formula:

$$\text{Flight Time (minutes)} = \left(\frac{\text{Battery Capacity (Ah)} \times 60}{\text{Current Consumption (A)}} \right)$$

$$\text{Flight Time} = \left(\frac{3.3 \times 60}{86} \right) \approx 2.3 \text{ minutes}$$

This represents the maximum expected duration under continuous full-load operation. In real conditions, the consumption may vary depending on the flight dynamics and payload usage.

4.5 Autonomous Drone Mechanism for Precision Palm Tree Pollination

The autonomous operation of the drone in our system progresses through four main stages, each increasing in intelligence and precision, as shown in Figure 4.38.

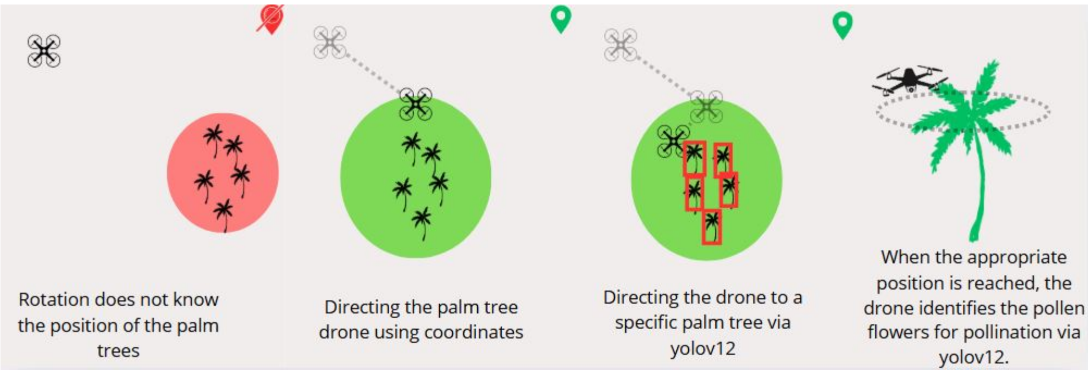


Figure 4.38. Design and Implementation of an Autonomous Drone Mechanism for Precision Palm Tree Pollination

1. **Initial State**
   The drone does not possess any coordinates of the palm trees and therefore remains idle, as shown in Figure 4.38.
2. **Receiving Coordinates**
   The ground station transmits the geographical coordinates of the palm trees to the drone. Based on this data, the drone navigates autonomously toward the specified area containing the palm trees, as shown in Figure 4.39.

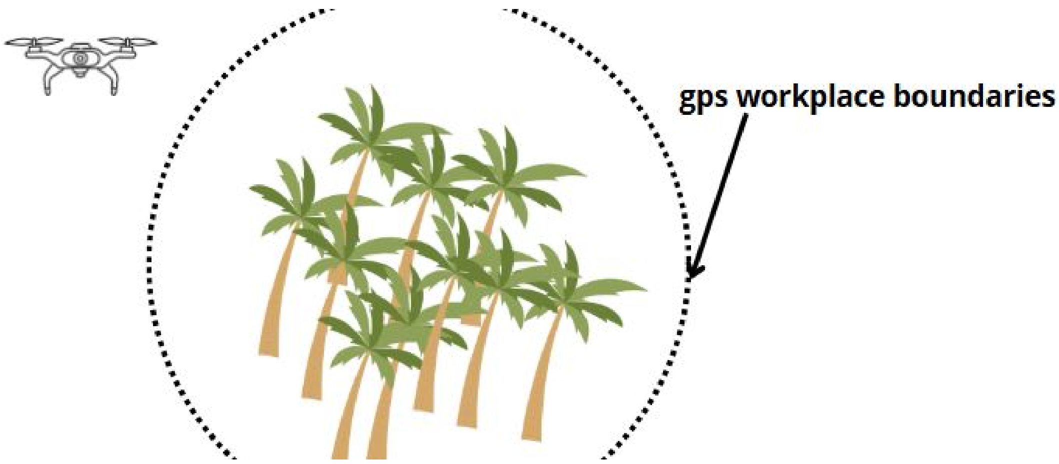


Figure 4.39. Receiving Coordinates

3. **Sequential Palm Tree Targeting via YOLOv12**
   Upon entering the vicinity, the drone activates its onboard camera and uses the YOLOv12 model running locally on the Raspberry Pi to detect palm trees in real-time. It then proceeds to each tree one by one based on visual confirmation rather than relying solely on GPS, as shown in Figure 4.40.

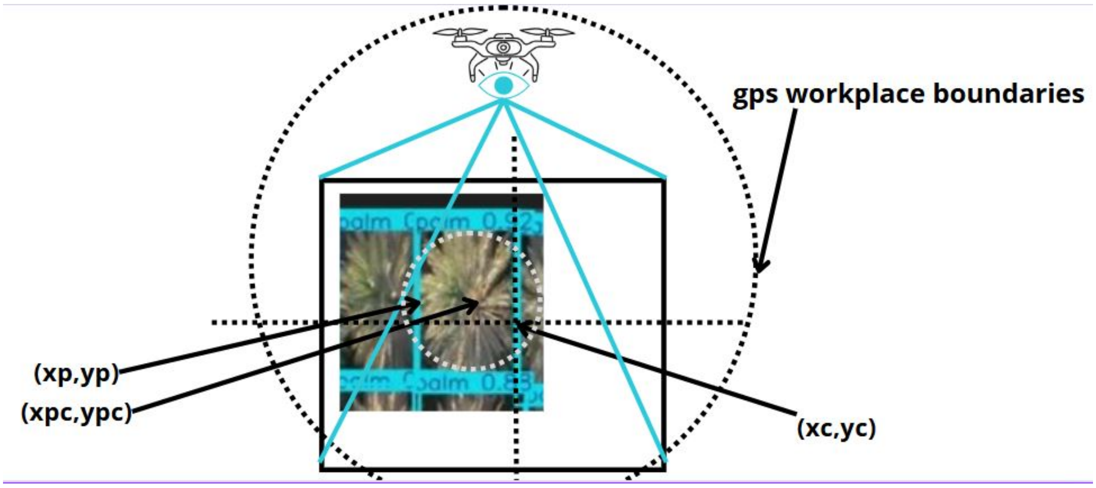


Figure 4.40. Sequential Palm Tree Targeting via YOLOv12

4. **Precision Pollination Using Top-View Detection**
   Once the drone reaches a palm tree, it slightly offsets from the center, descends, and initiates a top-down rotation around the palm crown. This maneuver enables the YOLOv12 model specifically trained on this viewpoint to detect the pollen flowers. Upon successful detection, the drone activates the pollination mechanism, as shown in Figure 4.41.

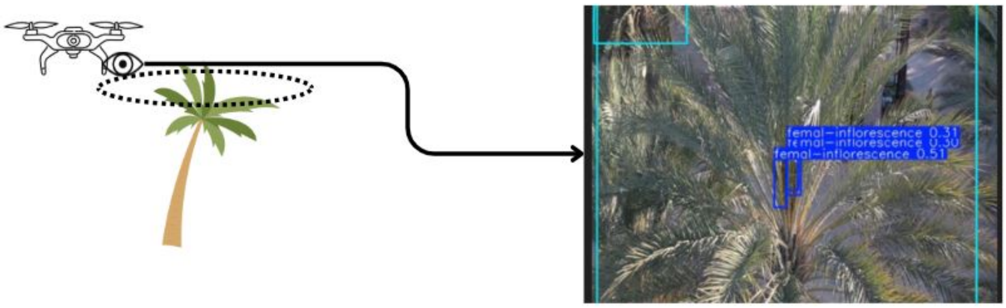


Figure 4.41. Precision Pollination Using Top-View Detection

4.6 Data Set

The dataset used in this project was specifically created to meet the requirements of the intelligent palm tree pollination task. We manually collected the images using a **DJI drone**, ensuring that the dataset accurately reflects the visual perspectives encountered by drones during real pollination missions in the field.

The image collection process focused on two primary viewpoints:

4.6.1 Top-down images

These were captured from above the palm trees, simulating the drone's view when approaching the center of the tree during flight, as illustrated in Figure 4.42.

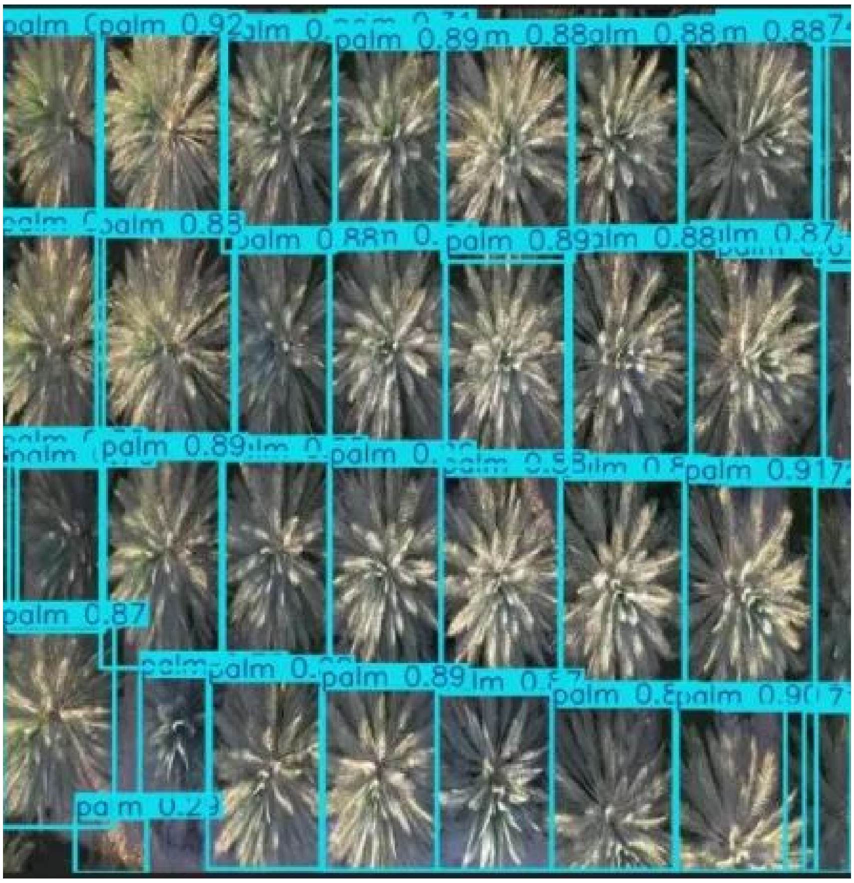


Figure 4.42. Top-down images

4.6.2 Surrounding crown images

These were taken while the drone hovered above the crown of the palm and gradually descended, capturing clear views of the pollen clusters, as shown in Figure 4.43.

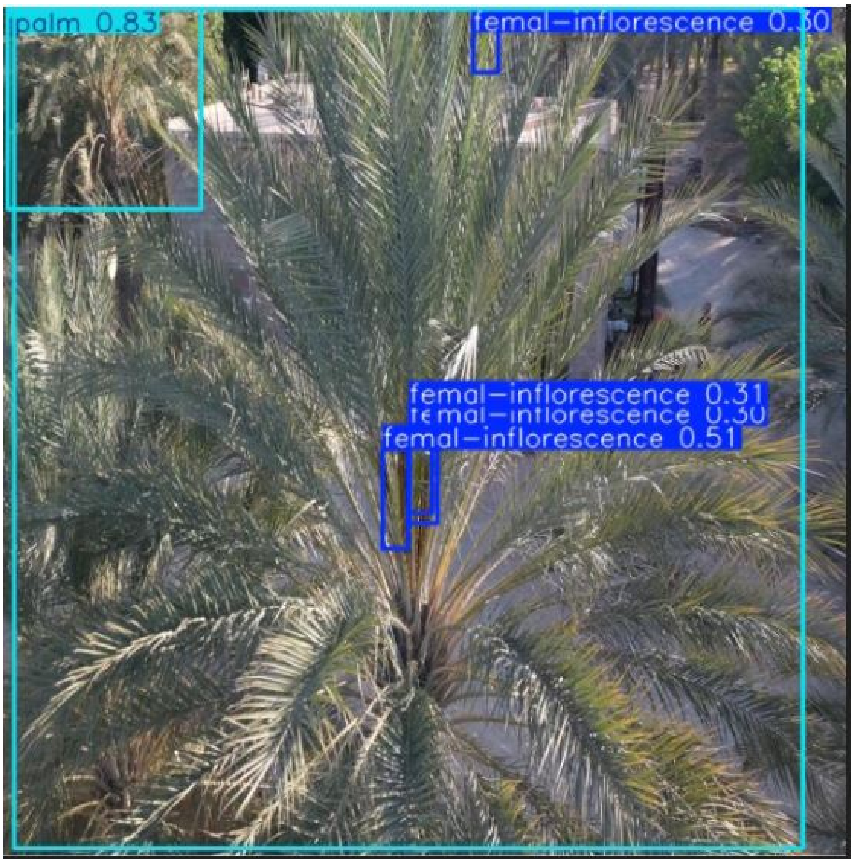


Figure 4.43. Surrounding crown images

These angles were deliberately chosen to match the drone's actual movement during pollination, where it slightly shifts from the center of the palm crown and lowers its altitude. Training the YOLOv12 model on such realistic images significantly enhances its detection accuracy during real-world deployment.

4.7 Applying YOLOv12 for Object Detection Using a Custom Dataset

This section introduces the YOLOv12 object detection model and highlights its integration into our system for real-time palm detection, as presented in [73].

4.7.1 Overview of YOLOv12

YOLOv12 (You Only Look Once – Version 12) is one of the latest advancements in the YOLO family of real-time object detection models. It marks a significant departure from purely convolutional architectures by introducing an attention-centric design, while still maintaining the low-latency performance that has made YOLO highly popular in real-world applications.

This version was developed to address the growing need for enhanced contextual awareness in object detection, without sacrificing computational efficiency. Unlike earlier attempts to integrate attention mechanisms often hindered by high complexity and memory overhead YOLOv12 incorporates attention in a highly optimized and lightweight manner.

4.7.1.1 Key Innovations and Enhancements in YOLOv12:

- **Area Attention Module** A novel mechanism that divides the feature map into horizontal or vertical segments, enabling a larger receptive field at a significantly reduced computational cost. It avoids complex operations such as window partitioning found in traditional Transformers.
- **R-ELAN (Residual Efficient Layer Aggregation Networks)** An improved feature aggregation module that introduces residual connections to overcome gradient blockage and instability in deep networks, particularly in larger-scale models.
- **FlashAttention Integration** This module addresses memory access inefficiencies typically associated with attention by optimizing GPU I/O operations. The result is reduced latency and enhanced inference speed.
- **Simplified Architectural Design** YOLOv12 removes unnecessary components such as positional encoding and replaces traditional linear layers (Linear + LayerNorm) with convolutional layers (Conv2D + BatchNorm), improving efficiency and compatibility with the YOLO ecosystem.
- **Comparison with Previous Versions** YOLOv12 outperforms previous versions such as YOLOv10 and YOLOv11 in both accuracy and efficiency. It achieves up to +2% higher mAP while maintaining comparable or faster inference speeds and requiring fewer parameters and FLOPs. These advantages make YOLOv12 particularly suitable for latency-critical applications such as autonomous driving, intelligent surveillance, and real-time industrial automation.

4.7.2 YOLOv12 Architecture

YOLOv12 is a state-of-the-art real-time object detection architecture designed to enhance both accuracy and speed in vision-based tasks. It introduces advanced mechanisms such as recursive extended layers (R-ELAN), area attention modules (A2C2f), and a multi-level feature aggregation neck, making it suitable for computationally efficient applications like autonomous drone navigation and precision agriculture.

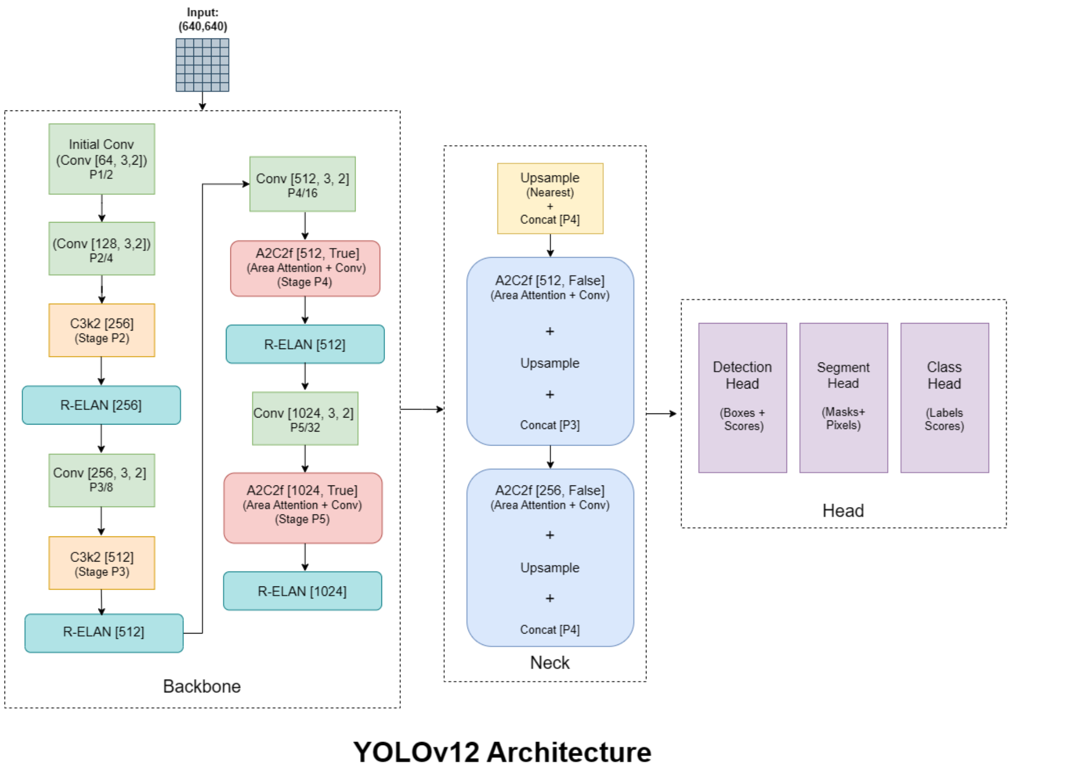


Figure 4.44. YOLOv12 Architecture [1]

Figure 4.44 above illustrates the full pipeline of the YOLOv12 architecture. The core components can be summarized as follows:

- **Backbone:** Responsible for initial feature extraction from the input image.
  - *Initial Conv + Downsampling:* Reduces spatial dimensions while increasing feature depth.
  - *C3k2 Blocks:* Convolutional structures using dual paths to capture diverse contextual information.
  - *R-ELAN Blocks:* Recursive Extended Layers for Aggregation of Network features, enhancing gradient flow and representation capacity.
- **A2C2f Modules:** Area Attention + Convolutional Fusion. These blocks combine spatial attention and convolutional operations to enhance semantic understanding. The Boolean flag (`True/False`) indicates whether fusion is applied.
- **Neck:** Aggregates multi-scale feature maps using upsampling and concatenation to preserve fine-grained and semantic information.
  - Integrates features from stages P3, P4, and P5 through top-down and lateral connections.
- **Head:** Contains three parallel output branches:
  - **Detection Head:** Outputs bounding boxes and confidence scores for detected objects.

  - **Segment Head:** Predicts pixel-wise masks for segmentation tasks.
  - **Class Head:** Assigns classification labels and scores to detected objects.

It is important to note that the core architecture of YOLOv12 remains structurally the same across its variants (e.g., YOLOv12n, YOLOv12s, YOLOv12l). The primary differences lie in the number of layers, parameters, and computational complexity. In our work, we adopted the `YOLOv12n` variant due to its lightweight design and suitability for real-time inference on embedded systems such as the Raspberry Pi 4.

4.8 Dataset Characteristics

4.8.1 General Statistics

- **Number of images:** 277
- **Image resolution:** 3840×2160 pixels (Median)
- **File format:** JPEG
- **Annotation format:** YOLOv12 (TXT)
- **Total annotations:** 3220 (average 11.6 per image)
- **Total classes:** 2
- **Classes and object counts:**
  - Palm: 2654
  - Female Inflorescence: 566

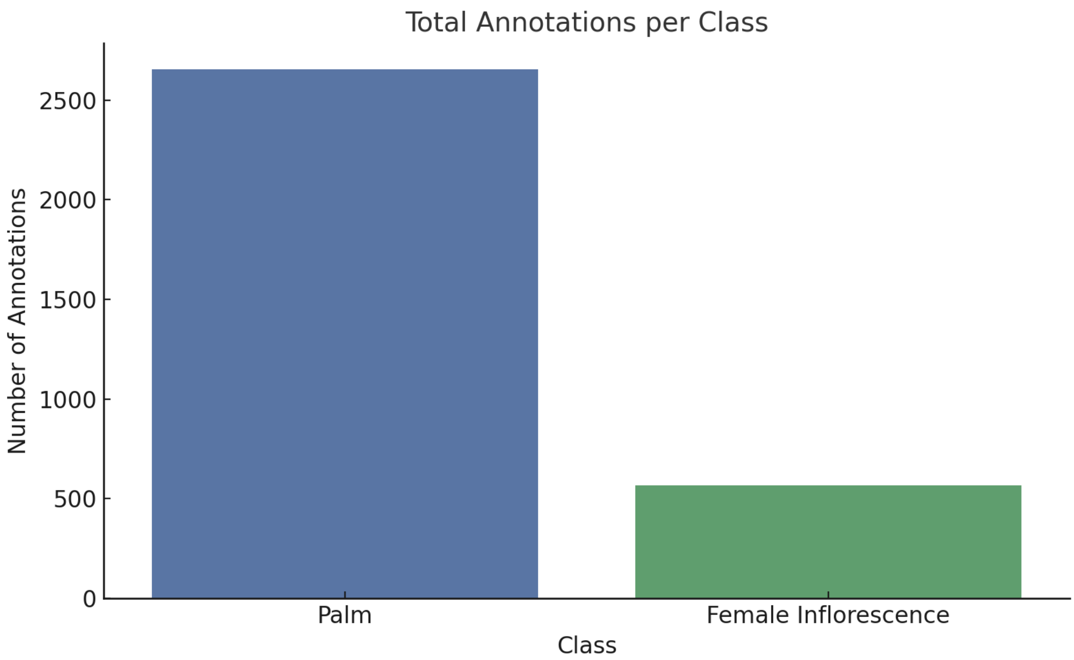


Figure 4.45. Total Annotations per Class

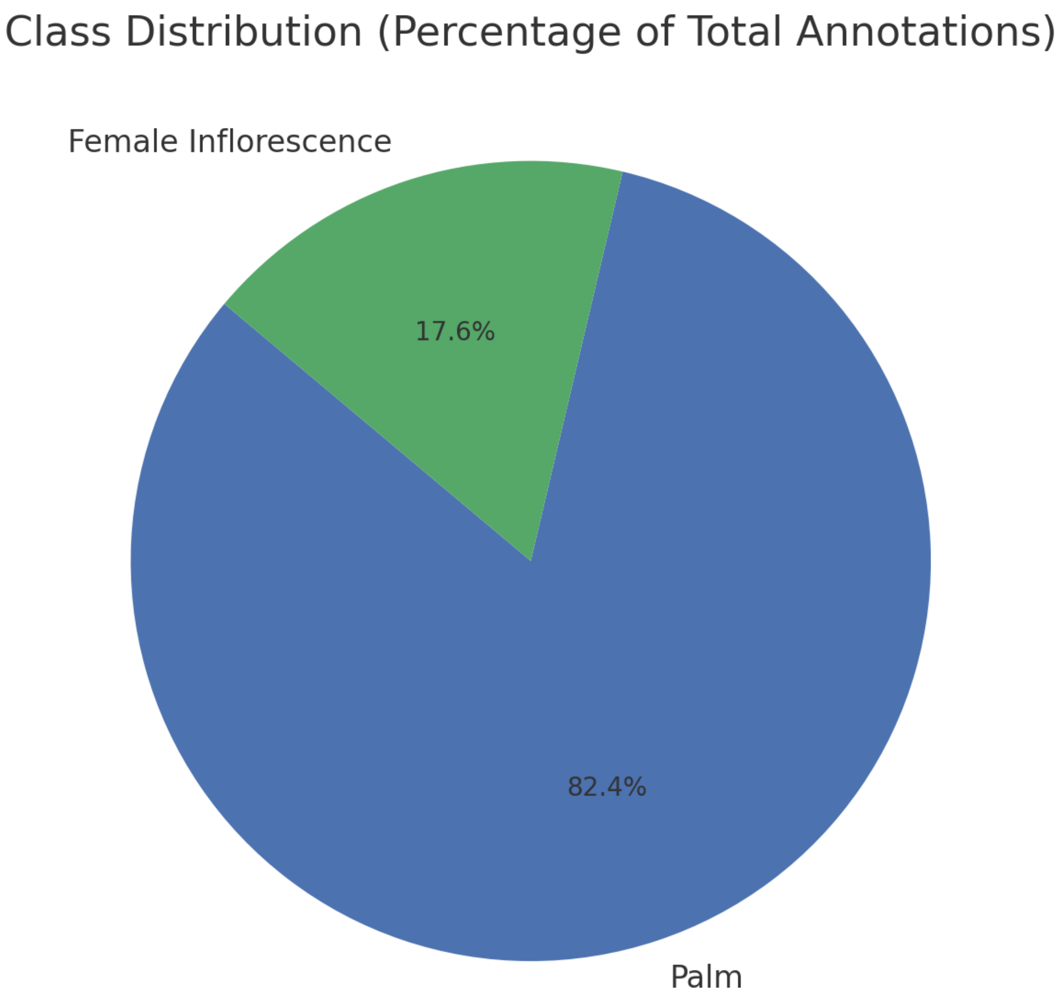


Figure 4.46. class Bistribution (percentage of Total Annotations)

4.8.2 Data Split

- **Training set 91% :** 2409 (Palm), 522 (Female Inflorescence)
- **Validation set 6% :** 178 (Palm), 26 (Female Inflorescence)
- **Test set 3% :** 67 (Palm), 18 (Female Inflorescence)

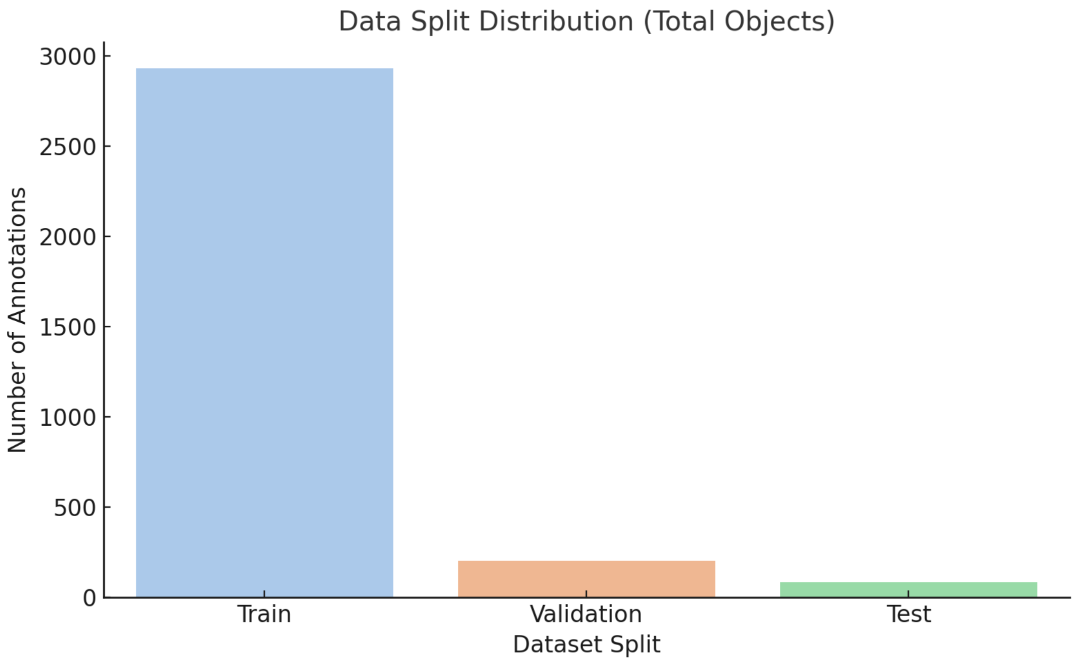


Figure 4.47. data split Distribution

Observation and Limitation: Despite our efforts to complete the annotation process in a timely manner, we were only able to annotate 277 images due to time constraints. As illustrated in the dataset analytics, there is a significant imbalance between the number of annotations for palm trees (2654) and female inflorescences (566). This disparity is likely to cause the YOLO model to develop a bias toward detecting palms more effectively than inflorescences.

Furthermore, the limited number of total images negatively impacts the overall training process, potentially reducing the model's generalization performance. However, as a growing startup, we are committed to continuous improvement. This includes extending and balancing our dataset to achieve more robust and satisfactory results in future iterations.

4.9 Dataset Preparation Using RoboFlow

In order to train YOLOv12 effectively on a custom object detection task, a well-annotated and properly structured dataset is essential. For this purpose, the RoboFlow platform was used to annotate, manage, and export the dataset in a format compatible with the YOLO framework, as illustrated in Figures 4.48 and 4.49.

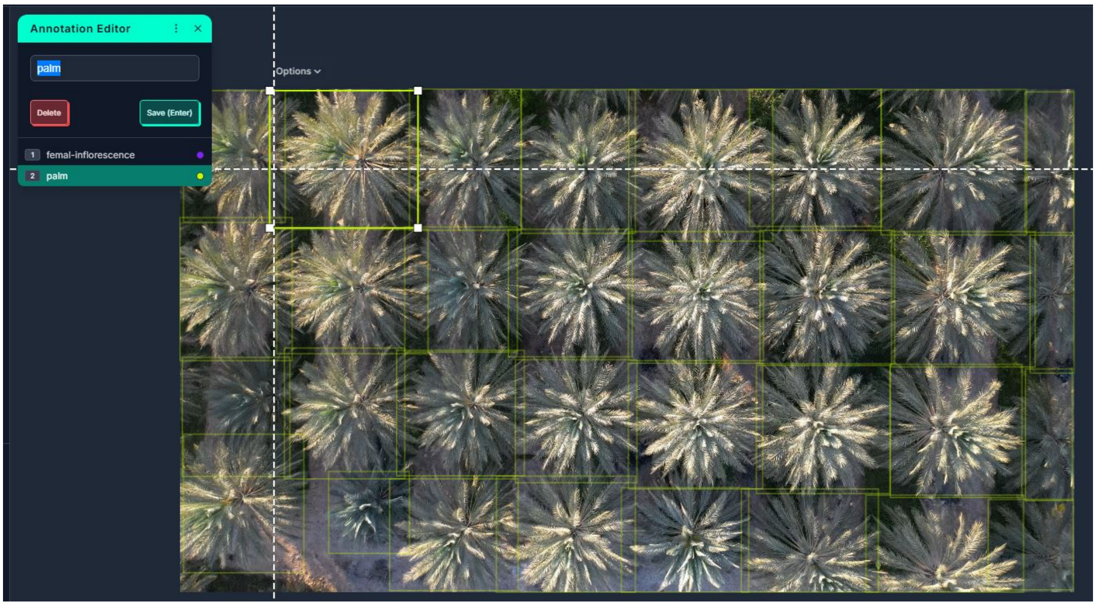


Figure 4.48. Annotation (palm)

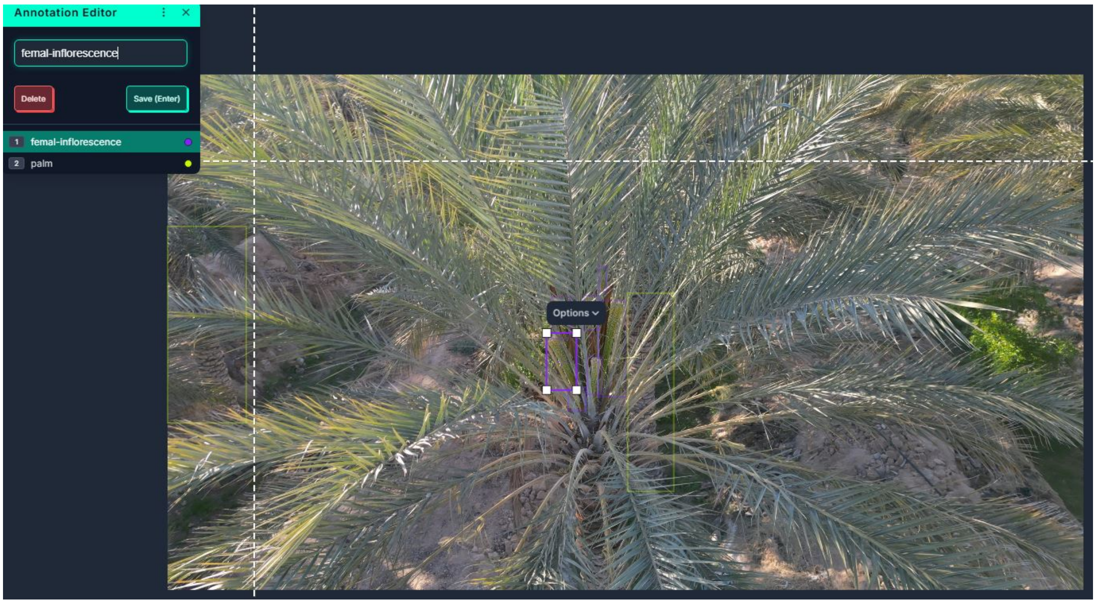


Figure 4.49. Annotation (femal-inflorescenc)

4.9.0.1 Image Upload and Annotation

The dataset preparation process began by uploading a collection of raw images to RoboFlow. These images were collected specifically for the target detection task and included a variety of objects under different lighting conditions, orientations, and backgrounds to ensure generalization, as shown in Figure 4.50.

Annotation was performed using RoboFlow's web-based annotation tool, which supports bounding box labeling. Each object instance within an image was manually labeled using rectangular bounding boxes, and a corresponding class name was assigned to each label. This step is crucial for supervised learning, as the model requires accurate localization and classification targets during training.

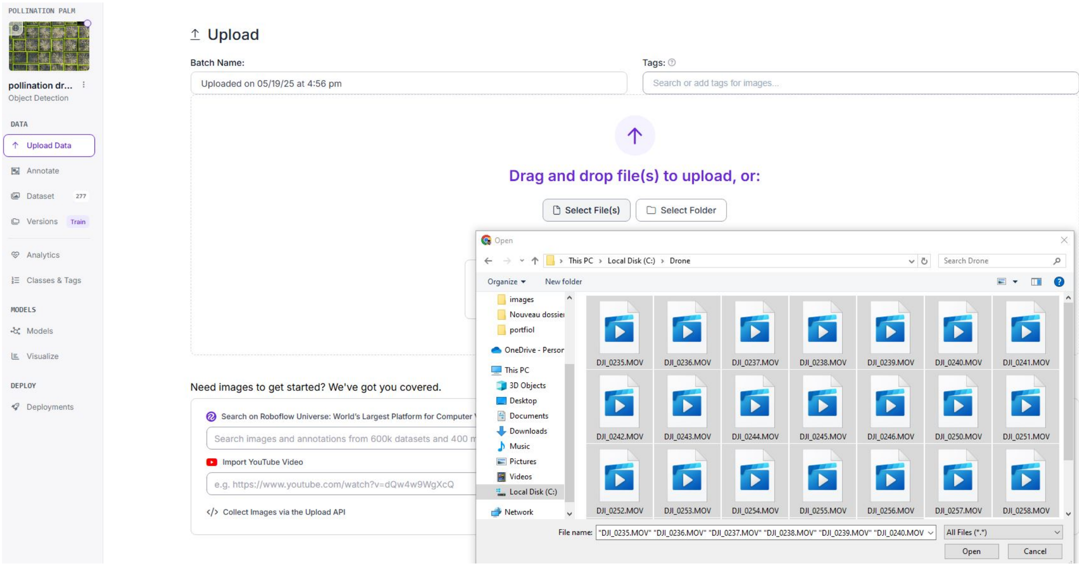


Figure 4.50. Upload images to RoboFlow

4.9.0.2 Dataset Versioning and Augmentation (Optional)

The dataset preparation process began by uploading a collection of raw images to RoboFlow. These images were collected specifically for the target detection task and included a variety of objects under different lighting conditions, orientations, and backgrounds to ensure generalization, as shown in Figures 4.51, 4.52, and 4.53.

Figure 4.51. Training Batch 0

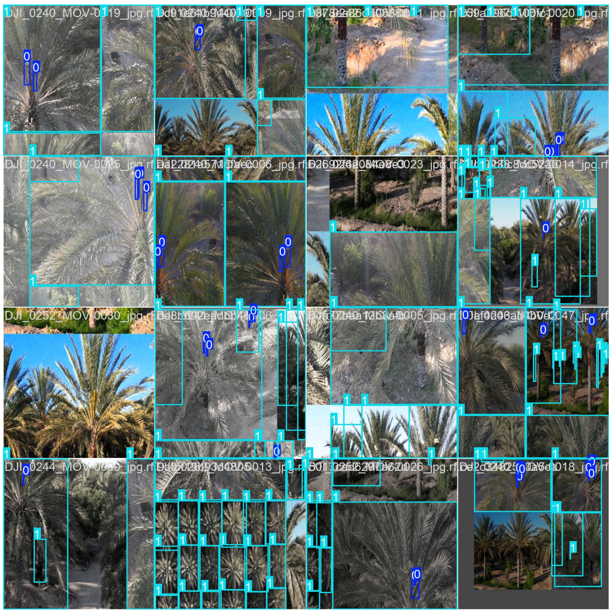


Figure 4.52. Training Batch 1

Figure 4.53. Training Batch 2

Annotation was performed using RoboFlow's web-based annotation tool, which supports bounding box labeling. Each object instance within an image was manually labeled using rectangular bounding boxes, and a corresponding class name was assigned to each label. This step is crucial for supervised learning, as the model requires accurate localization and classification targets during training. One example of dataset augmentation using geometric transformation is illustrated in Figure 4.54.

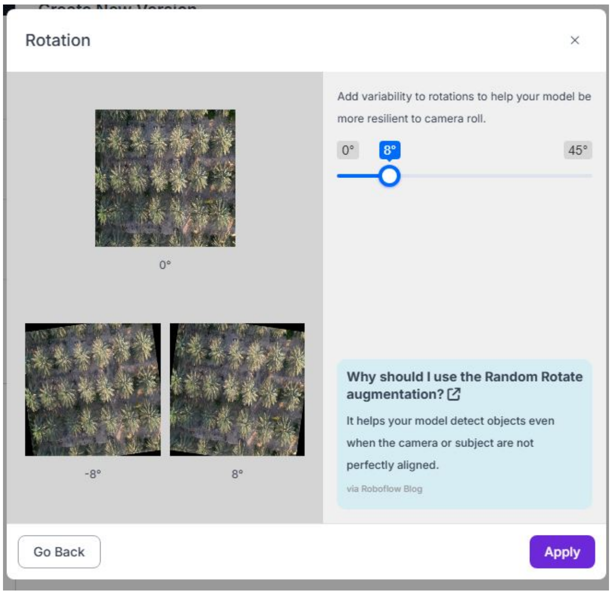


Figure 4.54. Augmentation (Using Rotation)

4.9.0.3 Exporting to YOLO Format

The annotated dataset was exported from RoboFlow using the YOLOv12 PyTorch format, which is fully compatible with YOLOv12. This export includes the following directory structure:

```
dataset/
  train/
    images/
    labels/
  valid/
    images/
    labels/
  test/
    images/
    labels/
  data.yaml
```

Each image in the images/ directory has a corresponding.txt annotation file in the labels/ directory. These text files follow the YOLO format, where each line represents one object instance in the following structure:

```
<class_id> <x_center> <y_center> <width> <height>
```

The content of the data.yaml file is as follows:

```yaml
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 2
names: ['femal-inflorescence', 'palm']

roboflow:
  workspace: pollination-palm
  project: pollination-drone
  version: 5
  license: CC BY 4.0
  url: https://universe.roboflow.com/pollination-palm/pollination-drone/dataset/5
```

Figure 4.55. content of the data.yaml

The data.yaml file plays a crucial role in the training process of YOLO-based models. It serves as a configuration file that provides essential information about the dataset, including:

- **The paths to the image folders** for training, validation, and testing.
- **The number of object classes** the model needs to detect.
- **The names of the classes** as defined during annotation.
- **Optional metadata**, such as dataset source information if using platforms like RoboFlow.

4.9.1 Training Configuration and Execution

The YOLOv12n model was trained using the Ultralytics framework with the custom dataset labeled in `pollination-drone-5/data.yaml`. Training was conducted on a single NVIDIA GPU with CUDA support enabled.

- **Model Architecture:** YOLOv12n (`yolov12n.yaml`)
- **Training Epochs:** 200
- **Batch Size:** 64
- **Image Size:** 640×640 pixels
- **Augmentations:**
  - `mosaic`: 1.0
  - `copy_paste`: 0.1
  - `mixup`: 0.0
  - `scale`: 0.5
- **Device:** CUDA GPU (device=0)

The training results were visualized with multiple metrics, including box loss, classification loss, and detection-focused losses (`dfl_loss`), along with standard metrics such as mAP@0.5 and mAP@0.5–0.95 (see Fig. 4.56).

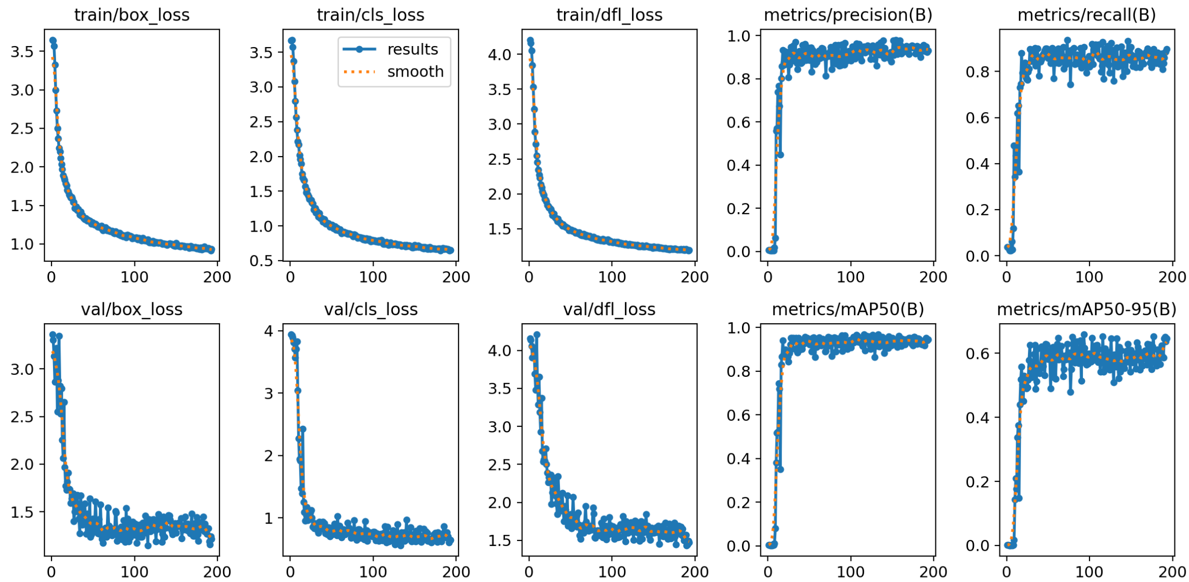


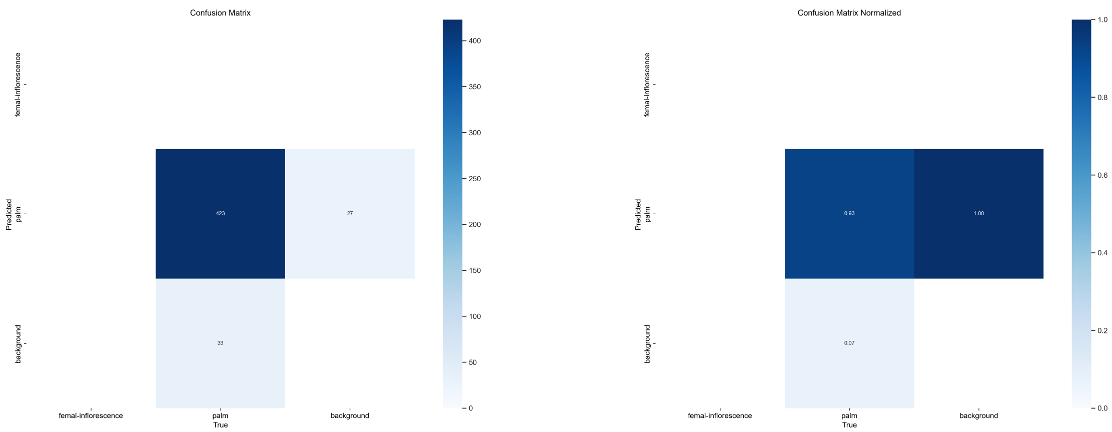
Figure 4.56. Training and Validation Metrics over 200 Epochs

4.9.2 Performance Evaluation

The trained YOLOv12n model was evaluated on a validation set using standard object detection metrics. The evaluation includes both raw and normalized confusion matrices, as well as performance curves for precision, recall, F1-score, and mAP.

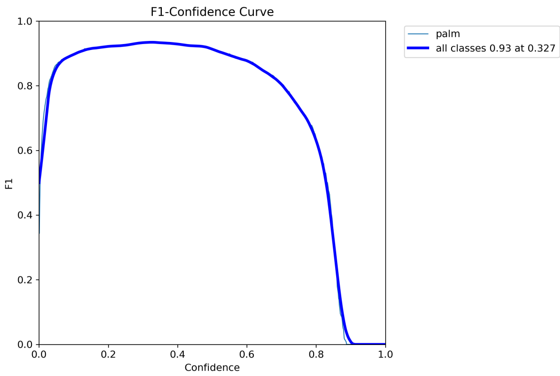
- **mAP@0.5:** 96.3%
- **Precision:** up to 100% at confidence threshold 0.744
- **Recall:** up to 97% at confidence 0.0
- **F1-score:** peaked at 0.93 at threshold 0.327
- **Classes Evaluated:** palm, background, female-inflorescence

Figures 4.57, 4.58, and 4.59 summarize the evaluation metrics.

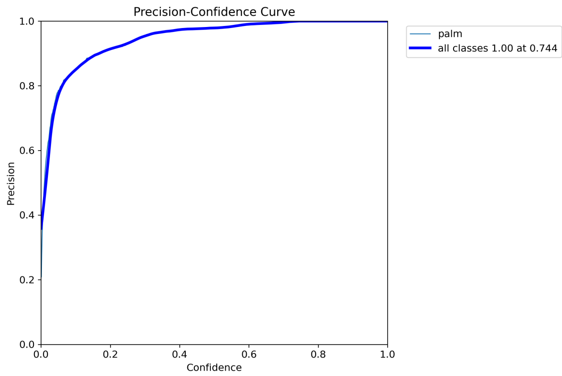


(a) Confusion Matrix (b) Normalized Confusion Matrix

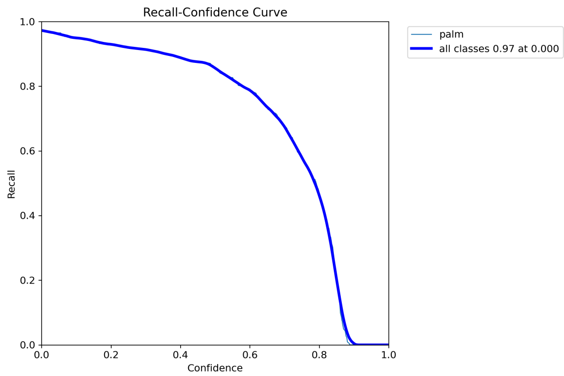
Figure 4.57. Confusion Matrix Results



(a) F1-Confidence Curve



(b) Precision-Confidence Curve



(c) Recall-Confidence Curve

Figure 4.58. Performance Confidence Curves

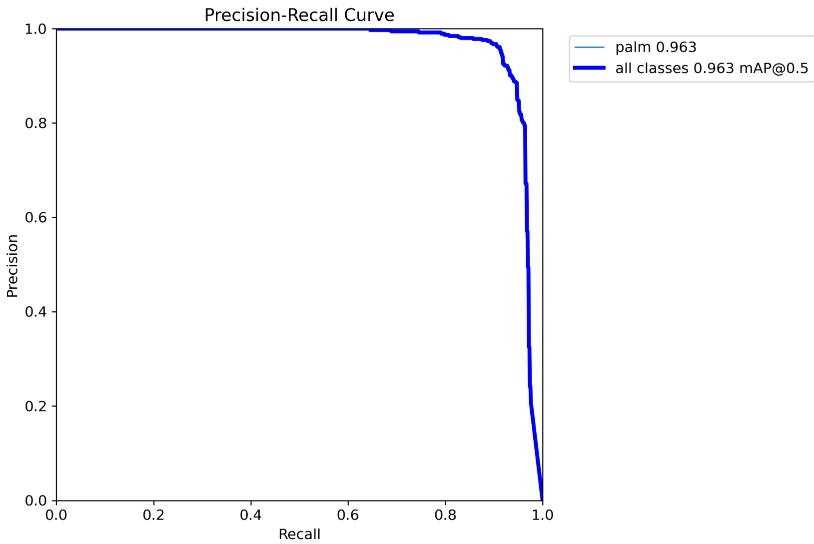


Figure 4.59. Precision-Recall Curve showing mAP@0.5 = 96.3%

4.9.3 Note on the Reliability of the Results:

Although the results presented may appear promising at first glance, a closer analysis of the dataset and training conditions reveals that they are not entirely justified.

Approximately 91% of the dataset was used for training, which led to a form of indirect data leakage into the validation and test sets. This highly unbalanced data split, combined with an excessively large number of epochs relative to the small dataset size (277 samples), strongly indicates that the model has experienced overfitting.

In other words, the model has likely memorized the training data rather than learning to generalize to unseen examples. These results are presented here merely as the final outcomes reached under the constraints we faced, particularly the limited time available and the seasonal nature of the data collection process, which restricted our ability to acquire a more balanced and extensive dataset.

Note: The trained YOLOv12n model is specifically optimized for two viewpoints: the side view and the top-down view used during the drone's operational flight path. As such, the detection performance outside these angles (e.g., extreme oblique or ground-level views) is not guaranteed and may lead to inaccurate predictions.

4.10 System Challenges and Technical Pitfalls

The system under investigation encountered a series of critical challenges during its design and implementation stages. These issues had direct implications on hardware integrity, system safety, and project continuity. The main problems can be summarized as follows:

1. **Use of Second-Hand Components to Cut Costs**
   In an effort to reduce overall expenses, several electronic components such as ESCs and sensors were sourced second-hand. However, a number of these components were found to be partially damaged or previously malfunctioning. This resulted in short circuits and, in some cases, the immediate burnout of parts upon power-up. The lack of proper documentation for these used items, combined with inconsistent voltage requirements, was identified as a primary cause of component incompatibility and failure.

2. **Insufficient Electrical Awareness Leading to System Failures**
   A lack of foundational understanding in electrical principles was evident during the integration process. This was particularly apparent in the misalignment of voltage ratings between power sources and receiving components. The absence of appropriate protective measures such as overcurrent protection and voltage regulation led to critical faults, component degradation, and, in several cases, irreversible damage to key modules.

3. **Improper PID Tuning Resulting in Hazardous System Behavior**
   The system's PID (Proportional, Integral, Derivative) parameters were not calibrated with sufficient care or precision. As a result, the drone exhibited unstable and unpredictable behavior during test flights, including excessive oscillations and abrupt movements. These reactions not only damaged sensitive hardware but also posed physical risks to operators during close-range testing. This highlights the necessity of a systematic, test-driven approach to PID tuning, supported by both simulation and controlled real-world trials.

4. **Lack of Financial Planning and Its Impact on Project Continuity**
   A non-technical but equally impactful issue was the absence of comprehensive

financial planning before project initiation. Several essential components were identified mid-project either as replacements for failed units or as critical upgrades but could not be acquired due to budget constraints. This led to significant delays, forced compromises in hardware selection, and reduced overall system reliability. The experience underscores the importance of preparing a realistic and flexible budget that accounts for unforeseen technical setbacks.

5. **Chemical Reaction Between Polyester and Patex Resulting in Structural Damage**
During the construction phase, polyester resin was combined with Patex adhesive in an attempt to reinforce the frame structure. Unfortunately, an unexpected chemical reaction occurred between the two substances, leading to the degradation and partial melting of the frame material. This incident revealed the importance of verifying chemical compatibility between structural and bonding materials before application, especially in systems exposed to mechanical stress or heat.

(a) Top view showing resin erosion

(b) Side view showing internal damage

(c) Detail of chemical degradation

Figure 4.60. Structural damage due to a chemical reaction between polyester and Patex adhesive. The foam material shows visible signs of erosion and deformation in multiple areas.

4.10.1 Prototype Development

The prototype developed in this project aims to demonstrate a drone-based palm pollination control system by integrating hardware components (Raspberry Pi 4B, Pixracer flight controller, camera, water pump, battery, and sensors) with an interactive software platform, as shown in Figure 4.61.

The prototype includes a controllable drone equipped with modules for receiving and executing commands, along with a real-time monitoring system. This test environment enables developers to dispatch missions, track drone status, and observe behavior. It also validates system responsiveness and full integration under realistic deployment conditions. Figures 4.61a and 4.61b show the prototype from the side and top views, respectively.

(a) Side view of the drone prototype

(b) Top view of the drone prototype

Figure 4.61. Prototype used for system testing from different angles

4.10.2 Desktop Application Functions

A desktop application was built using Python and PyQt5 to enable user interaction with the drones via a graphical interface. The application is composed of two main interfaces: the mission control view and the drone status monitor. It also includes a top menu bar with essential options.

Main Interface – Mission Control View

This is the primary screen used to define mission zones, dispatch tasks, and monitor progress. It includes:

- A control panel with mission buttons on the left.
- An interactive map in the center.
- A real-time status panel on the right.

As shown in Figure 4.62, this interface displays the field segmentation with clear task assignment per drone. The status panel on the right provides real-time indicators such as battery level, connection status, and mission progress.

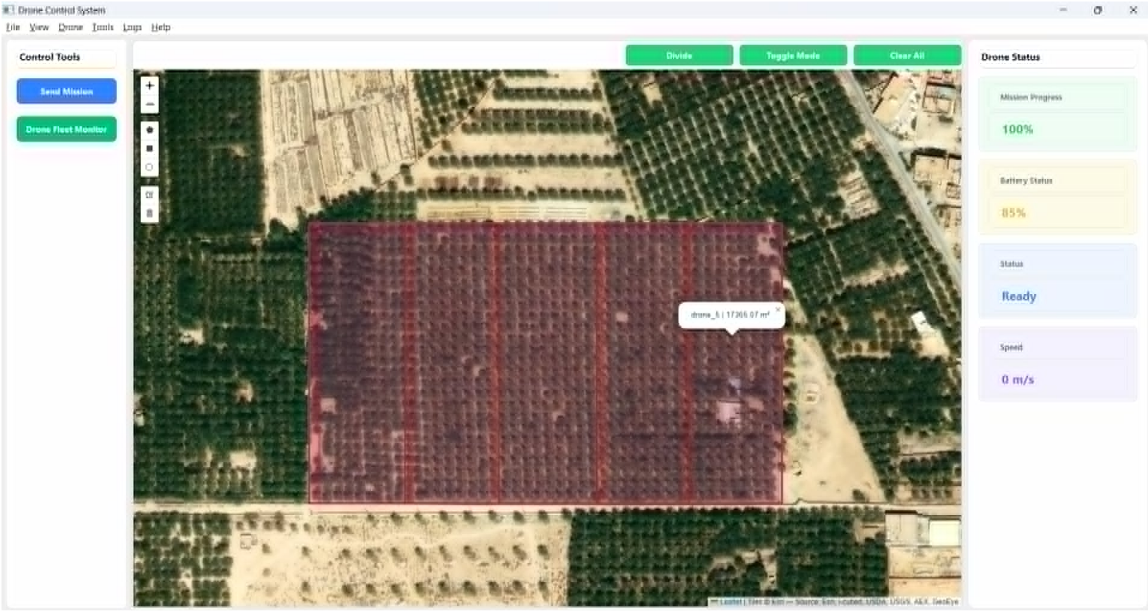


Figure 4.62. Mission control interface showing segmented field zones and drone mission status

Drone Fleet Monitor – Video and Telemetry

This secondary interface provides a tabbed view of each connected drone. It shows the live camera feed, GPS coordinates, altitude, speed, battery status, and connection status.

Figure 4.63 illustrates this real-time monitoring dashboard. Each tab allows the user to observe one drone's visual feed and its associated telemetry data, enabling better tracking, diagnostics, and mission safety.

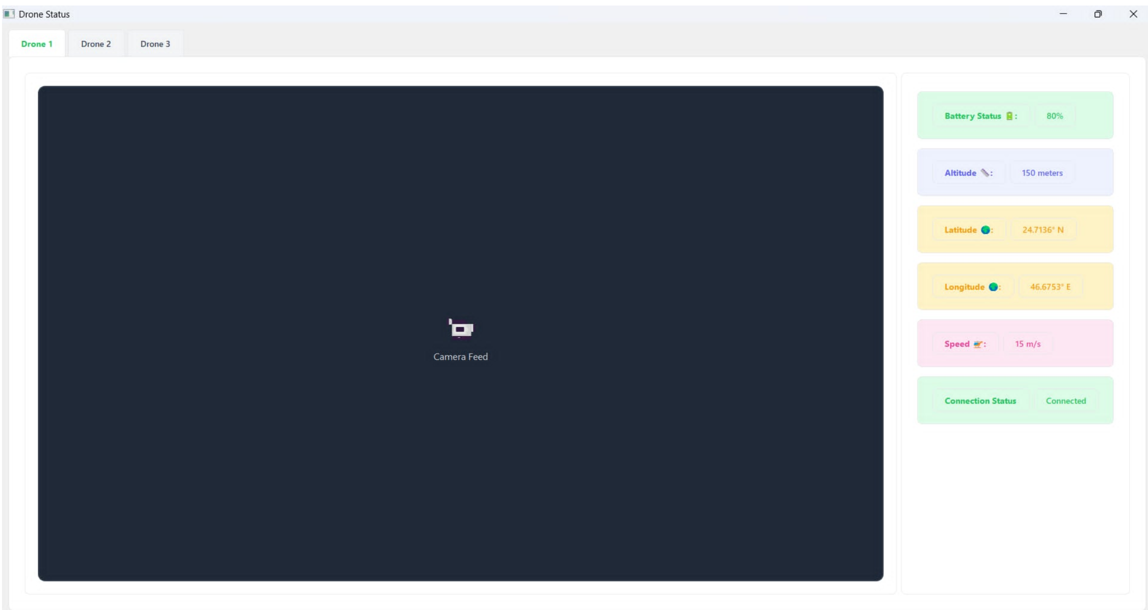


Figure 4.63. Drone fleet monitor interface with live video feed and telemetry panel

Menu Bar

The menu bar at the top provides access to standard operations and configuration tools. It contains:

- **File** – Exit, Save, Load.
- **View** – Toggle visibility of panels and map layers.
- **Drone** – Connect/Disconnect, Drone Info.
- **Tools** – Simulation tools, log export.
- **Logs** – View system logs.
- **Help** – About, User Manual.

These menus enhance usability and allow for easier configuration, debugging, and monitoring during real-time mission deployment.

4.10.3 Mission Execution Workflow

The mission execution flow is designed to follow these steps:

- The user draws a mission zone on the interactive map.
- When the "divide" button is clicked, the zone is automatically divided according to the number of connected drones.
- Each drone receives coordinates for its assigned segment.
- Commands are transmitted through the system's integrated communication layer, allowing for theoretical autonomous mission execution and simulated feedback under controlled testing conditions.
- The system architecture supports real-time status reporting such as position and mission progress.

4.10.4 Live Data Monitoring

The status panel continuously displays:

- Battery percentage.
- Mission status (Pending, In Progress, Completed).
- Current flight speed.
- Mission completion percentage.

4.10.5 Multi-Drone Support and Task Division

The system includes practical support for multi-drone coordination through automated task segmentation. In real-world testing, the designated field area was divided into equal segments based on the number of available drones, and each drone received its corresponding mission coordinates. Although the actual takeoff and mission execution were not completed due to battery limitations, the communication layer successfully transmitted the task assignments, validating the core coordination logic. This feature is designed to enhance efficiency by enabling parallel task execution and reducing overall operation time.

4.11 Conclusion

This chapter presented the final integration and testing of the proposed autonomous drone system for palm tree pollination. Core hardware components including the `Raspberry Pi`, `Pixracer`, camera, sensors, water pump, and servo motor were successfully assembled and interfaced with the YOLOv12 model for real-time palm tree detection.

Internal communication between the Raspberry Pi and Pixracer via the `MAVLink` protocol enabled retrieval of essential telemetry data, including battery level, GPS coordinates, and flight status. External communication with the ground station over `UDP` was validated through successful transmission of status updates and mission zone assignments. Additionally, the water level sensor and pump activation mechanism were successfully tested, confirming the drone's ability to respond to detection events with pollination actions.

While a complete outdoor flight could not be conducted due to power limitations, the mission logic, command delivery, and key functional modules were validated under controlled indoor conditions. These results demonstrate that the system architecture is modular, operational, and ready for further testing.

In conclusion, this phase confirms the system's technical viability and establishes a solid foundation for future improvements in power management, outdoor flight capabilities, and large-scale deployment.

General Conclusion and Future Work

General Conclusion

The development of an autonomous drone system for pollinating palm trees marks a significant advancement in the integration of smart technologies within modern agriculture. This project showcases how a traditionally manual and labor-intensive process can be automated through the combination of wireless communication protocols, computer vision, embedded control systems, and unmanned aerial vehicles.

Each drone is equipped with a `Raspberry Pi`, a `Pixracer` flight controller, and the `YOLOv12` object detection model, enabling real-time palm tree detection and autonomous pollination. The use of both `UDP` and `MAVLink` protocols within a star communication topology supports centralized mission management while allowing local autonomy at the drone level, facilitating effective task allocation and system supervision.

Although full outdoor deployment was limited due to power constraints, the experimental phase validated the system's core functionalities. Key processes such as visual detection, internal data exchange, status retrieval, and pollination mechanism triggering were successfully tested under controlled conditions. These results demonstrate that the system is technically ready for extended trials once energy-related limitations are resolved.

This work lays a solid foundation for future research in precision agriculture. Potential improvements include integrating dynamic path planning algorithms for enhanced obstacle avoidance, enabling multi-drone collaboration, and adapting the system to other crop types or farming environments.

In summary, the project demonstrates the transformative potential of combining real-time communication, autonomous robotics, and artificial intelligence to address critical agricultural challenges especially in regions facing labor shortages and seeking sustainable, efficient solutions.

Future Work

The current system has successfully demonstrated the feasibility of autonomous palm tree pollination using drones equipped with onboard intelligence and centralized task coordination. Nonetheless, several directions can be pursued to improve its functionality, scalability, and adaptability for broader applications in smart agriculture.

First, the system could be extended beyond pollination tasks. By integrating additional sensors and specialized modules, the drones could perform critical agricultural operations such as disease detection, pesticide spraying, and soil condition monitoring. This would allow the system to assess plant health, evaluate soil moisture, and support various crop types positioning it as a versatile smart farming platform.

Second, incorporating dynamic path planning algorithms would enable autonomous navigation with obstacle avoidance and route optimization. This is especially beneficial in complex or irregular farm environments where real-time adaptability is essential.

Third, enhancing inter-drone communication capabilities would allow for collaborative behaviors such as real-time task redistribution, shared decision-making, and adaptive mission support. While the current design uses a star topology with centralized coordination, a hybrid or mesh communication model could significantly improve system robustness and efficiency in large-scale deployments.

Fourth, upgrading the YOLOv12 detection model to recognize flowers at different growth stages and evaluate pollination readiness would improve operational precision. This enhancement requires more diverse training datasets and fine-tuned detection parameters.

Fifth, integrating weather sensors and predictive analytics would allow the system to optimize mission timing based on environmental conditions, improving success rates and reducing unnecessary pollination attempts.

Finally, power autonomy remains a key challenge. Implementing advanced energy management strategies such as solar-powered recharging stations or intelligent path planning based on battery levels could extend flight duration and minimize downtime.

Collectively, these future enhancements would transform the current prototype into a robust, intelligent, and scalable precision agriculture system capable of adapting to a wide range of field conditions and farming needs.

Bibliography

[1] Wilfried Yves Hamilton Adoni, Sandra Lorenz, Junaidh Shaik Fareedh, Richard Gloaguen, and Michael Bussmann. Investigation of autonomous multi-uav systems for target detection in distributed environment: Current developments and open challenges. *Drones*, 7(4):263, 2023.

[2] Bilal Ahmed, Hasnat Shabbir, Syed Rameez Naqvi, and Lu Peng. Smart agriculture: Current state, opportunities, and challenges. *IEEE Access*, 12:1–23, 2024.

[3] Jamilah ALAMRI, Rafika HARRABI, and Slim BEN CHAABANE. Face recognition based on convolution neural network and scale invariant feature transform. *International Journal of Advanced Computer Science and Applications*, 12(2), 2021.

[4] Muawya Alasasfa. Effect of pollination methods on fruit set, yield, physical and chemical properties of hayani date palm cultivar. *International Journal of Environmental and Agriculture Research (IJOEAR)*, 7(1):24–30, 2021.

[5] Azza Allouch, Omar Cheikhrouhou, Anis Koubâa, Mohamed Khalgui, and Tarek Abbes. Mavsec: Securing the mavlink protocol for ardupilot/px4 unmanned aerial systems. *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 1194–1199, 2019.

[6] Mohammed A. S. Alyafei, Abdullah Al Dakheel, Mohamed Almoosa, and Zienab F. R. Ahmed. Innovative and effective spray method for artificial pollination of date palm using drone. *HortScience*, 57(10):1298–1305, 2022.

[7] Ahmad Ali Alzubi and Kalda Galyna. Artificial intelligence and internet of things for sustainable farming and smart agriculture. *IEEE Access*, 11:78686, 2023.

[8] Godwin Asaamoning, Paulo Mendes, Denis Rosário, and Eduardo Cerqueira. Drone swarms as networked control systems by integration of networking and computing. *Sensors*, 21(8):2642, 2021.

[9] Reg Austin. *Unmanned Aircraft Systems: UAVS Design, Development and Deployment*. John Wiley & Sons, Chichester, UK, 2010. See Chapter 13: Control Stations.

[10] Athanasios T Balafoutis, Bernhard Beck, Spyros Fountas, Zisis Tsiropoulos, Jeroen Vangeyte, Thomas van der Wal, Isabel Soto, Manuel Gómez-Barbero, Andrew Barnes, and Vera Eory. Precision agriculture technologies positively contributing to ghg emissions mitigation, farm productivity and economics. *Sustainability*, 9(8):1339, 2017.

[11] Dawei Bie, Daochun Li, Jinwu Xiang, Huadong Li, Zi Kan, and Yi Sun. Design, aerodynamic analysis and test flight of a bat-inspired tailless flapping wing unmanned aerial vehicle. *Aerospace Science and Technology*, 112:106557, 2021.

[12] Claudiu George Bocean. A cross-sectional analysis of the relationship between digital technology use and agricultural productivity in eu countries. *Agriculture*, 14(4):519, 2024.

[13] Hua-Ching Chen, Shih-An Li, Tsung-Han Chang, Hsuan-Ming Feng, and Yun-Chien Chen. Hybrid centralized training and decentralized execution reinforcement learning in multi-agent path-finding simulations. *Applied Sciences*, 14(10):3960, 2024.

[14] Xi Chen, Jun Tang, and Songyang Lao. Review of unmanned aerial vehicle swarm communication architectures and routing protocols. *Applied Sciences*, 10(10):3661, 2020.

[15] UAV Coach. Lidar drones: An in-depth guide [new for 2025], 2025. Accessed: 2025-05-26.

[16] Aiya Cui, Ying Zhang, Pengyu Zhang, and Chunyan Wang. Intelligent health management of fixed-wing uavs: A deep-learning-based approach. In *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, pages 1–6. IEEE, 2020.

[17] Keith S Delaplane and David F Mayer. *Crop pollination by bees*. CABI Publishing, Wallingford, UK, 2000.

[18] Muthumanickam Dhanaraju, Poongodi Chenniappan, Kumaraperumal Ramalingam, Sellaperumal Pazhanivelan, and Ragunath Kaliaperumal. Smart farming: Internet of things (iot)-based sustainable agriculture. *Agriculture*, 12(10):1745, 2022.

[19] DJI. Agriculture drone insurance, June 2024. Accessed on April 11, 2025.

[20] Karel Domin, Iraklis Symeonidis, and Eduard Marin. Security analysis of the drone communication protocol: Fuzzing the mavlink protocol. *Unpublished manuscript, KU Leuven and University of Luxembourg*, pages 1–7, 2016.

[21] DroneU. The future of drone mapping with slam technology, 2025. Accessed: 2025-05-26.

[22] Sami El-Ferik, Basem Almadani, and Siddig M. Elkhider. Formation control of multi unmanned aerial vehicle systems based on dds middleware. *IEEE Access*, 8:44211–44221, 2020.

[23] FAO. Climate-smart agriculture sourcebook, 2013. https://www.fao.org/3/i3325e/i3325e.pdf.

[24] FAO. Drones for agriculture. Food and Agriculture Organization of the United Nations, 2018. Available at: https://www.fao.org/3/I8494EN/i8494en.pdf.

[25] FAO. Why bees matter: The importance of pollinators for food and agriculture. Food and Agriculture Organization of the United Nations, 2019. Available at: https://www.fao.org/3/i9527en/i9527en.pdf.

[26] FAO. The importance of bees and other pollinators for food and agriculture. Food

and Agriculture Organization of the United Nations, 2021. Available at: https://www.fao.org/3/cb2806en/cb2806en.pdf.

[27] James E. Gallagher and Edward J. Oughton. Surveying you only look once (yolo) multispectral object detection advancements, applications, and challenges. *IEEE Access*, 13:7366–7395, 2025.

[28] Nicolas Gallai, Jean-Michel Salles, Josef Settele, and Bernard E Vaissiere. Economic valuation of the vulnerability of world agriculture confronted with pollinator decline. *Ecological Economics*, 68(3):810–821, 2009.

[29] Shivani Garg, Nelson Pynadathu Rumjit, and Swapnila Roy. Smart agriculture and nanotechnology: Technology, challenges, and new perspective. *Advanced Agrochem*, 3:115–125, 2024.

[30] Robin Gebbers and Viacheslav I Adamchuk. Precision agriculture and food security. *Science*, 327(5967):828–831, 2010.

[31] Sewnet Getahun, Habtamu Kefale, and Yohannes Gelaye. Application of precision agriculture technologies for sustainable crop production and environmental sustainability: A systematic review. *The Scientific World Journal*, 2024:12, 2024.

[32] Hamid Hassani, Anass Mansouri, and Ali Ahaitouf. Performance evaluation of control strategies for autonomous quadrotors: A review. *Complexity*, 2024:1–43, 2024.

[33] Takefumi Hiraguri, Hiroyuki Shimizu, Tomotaka Kimura, Takahiro Matsuda, Kazuki Maruta, Yoshihiro Takemura, Takeshi Ohya, and Takuma Takanashi. Autonomous drone-based pollination system using ai classifier to replace bees for greenhouse tomato cultivation. *IEEE Access*, 11:99352–99362, 2023.

[34] Mohammad Enayet Hossain, Saif Shahrukh, and Shahid Akhtar Hossain. Chemical fertilizers and pesticides: Impacts on soil degradation, groundwater, and human health in bangladesh. In V.P. Singh, S. Yadav, K.K. Yadav, and R.N. Yadava, editors, *Environmental Degradation: Challenges and Strategies for Mitigation*, volume 104 of *Water Science and Technology Library*, pages 63–92. Springer, Cham, 2022.

[35] Justin Hu, Ariana Bruno, Drew Zagieboylo, Mark Zhao, Brian Ritchken, Brendon Jackson, Joo Yeon Chae, Francois Mertil, Mateo Espinosa, and Christina Delimitrou. To centralize or not to centralize: A tale of swarm coordination. *arXiv preprint arXiv:1805.01786*, 2018.

[36] Yawei Huang, Jie Liu, and Xinyao Wu. Comparative analysis of manual and drone-based spraying methods in agriculture. *Agricultural Engineering International: CIGR Journal*, 23(1):144–153, 2021.

[37] IPBES. Assessment report on pollinators, pollination and food production. Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services, 2016. Available at: https://www.ipbes.net/assessment-reports/pollinators.

[38] Ayidu N. Ja and O. V. Elaigwu. Probability prediction of user datagram protocol (udp) upstream throughput in a network. *Journal of Energy Technology and Environment*, 4(2):12–21, 2022.

[39] S.V.K. Jagadish, R.N. Bahuguna, and Stefanie Heuer. Challenges and opportunities in the adoption of unmanned aerial systems in agriculture. *Agricultural*

Systems, 178:102761, 2020.

[40] JOUAV. The ultimate guide to gps drones in 2025, 2025. Accessed: 2025-05-26.

[41] Pradeep Kale, Karunakar Pothuganti, and Mesfin Jariso. A review on geo mapping with unmanned aerial vehicles. *International Journal of Advanced Research in Computer and Communication Engineering*, 6(1):313–319, 2017.

[42] Andreas Kamilaris, Andreas Kartakoullis, and Francesc X Prenafeta-Boldú. Agri-iot: A semantic framework for internet of things-enabled smart farming applications. *Computers and Electronics in Agriculture*, 147:131–147, 2018.

[43] Alexandra-Maria Klein, Bernard E Vaissière, James H Cane, Ingolf Steffan-Dewenter, Saul A Cunningham, Claire Kremen, and Teja Tscharntke. Importance of pollinators in changing landscapes for world crops. *Proceedings of the Royal Society B: Biological Sciences*, 274(1608):303–313, 2007.

[44] Denis Kotarski, Petar Piljek, Josip Kasac, and Dubravko Majetic. Performance analysis of fully actuated multirotor unmanned aerial vehicle configurations with passively tilted rotors. *Applied Sciences*, 11(18):8542, 2021.

[45] Anis Koubâa, Basit Qureshi, Mohamed-Foued Sriti, Azza Allouch, Yasir Javed, Maram Alajlan, Omar Cheikhrouhou, Mohamed Khalgui, and Eduardo Tovar. Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones. *Ad Hoc Networks*, 86:46–62, 2019.

[46] J. Laaksonen and E. Oja. Classification with learning k-nearest neighbors. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 3, pages 1480–1483 vol.3, 1996.

[47] Ka-Cheong Leung, Victor O.K. Li, and Daiqin Yang. An overview of packet reordering in transmission control protocol (tcp): Problems, solutions, and challenges. *IEEE Transactions on Parallel and Distributed Systems*, 18(4):522–535, 2007.

[48] Jiajia Li, Mingle Xu, Lirong Xiang, Dong Chen, Weichao Zhuang, Xunyuan Yin, and Zhaojian Li. Foundation models in smart agriculture: Basics, opportunities, and challenges. *Computers and Electronics in Agriculture*, 222, 2024.

[49] Q. Li and Y. Wang. The research of media streaming technology based on rtp protocol. *ResearchGate*, 2022.

[50] Shize Lu and Xinqing Xiao. Neuromorphic computing for smart agriculture. *Agriculture*, 14(11):1977, 2024.

[51] Arindam Majee, Rahul Saha, Snehasish Roy, Srilekha Mandal, and Sayan Chatterjee. Swarm uavs communication. *arXiv preprint arXiv:2405.00024*, 2024.

[52] S.E. McGregor. *Insect pollination of cultivated crop plants*. US Department of Agriculture, Washington, D.C., 1976.

[53] Daniel Mejias, Zaloa Fernandez, Roberto Viola, Ander Aramburu, Igor Lopez, and Andoni Diaz. Towards railways remote driving: Analysis of video streaming latency and adaptive rate control. *arXiv preprint*, 2024.

[54] Kyohei Miyoshi, Takefumi Hiraguri, Hiroyuki Shimizu, Kunihiko Hattori, Tomotaka Kimura, Sota Okubo, Keita Endo, Tomohito Shimada, Akane Shibasaki, and Yoshihiro Takemura. Development of pear pollination system using autonomous

drones. *AgriEngineering*, 7(3):68, 2025.

[55] Ashish Mohan, Anirudh Muraleedharan, Shruthi CM, and Akshay Nagarajan. Autonomous payload delivery using hybrid vtol uavs for community emergency response. In *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1234–1240. IEEE, 2020.

[56] Ghulam Mohyuddin, Muhammad Adnan Khan, Abdul Haseeb, Shahzadi Mahpara, Muhammad Waseem, and Ahmed Mohammed Saleh. Evaluation of machine learning approaches for precision farming in smart agriculture system: A comprehensive review. *IEEE Access*, 12:60155, 2024.

[57] Md. Najmul Mowla, Neazmul Mowla, A. F. M. Shahen Shah, Khaled M. Rabie, and Thokozani Shongwe. Internet of things and wireless sensor networks for smart agriculture applications: A survey. *IEEE Access*, 11:145813, 2023.

[58] Ravinder Nath, Harpreet Singh, and Santanu Mukherjee. Insect pollinators decline: an emerging concern of anthropocene epoch. *Journal of Apicultural Research*, 2022.

[59] MIT News. Engineers enable a drone to determine its position in the dark and indoors, 2025. Accessed: 2025-05-26.

[60] Burak Ozkan, Ibrahim Demir, and Harun Cimen. Drone-assisted artificial pollination: A practical approach for precision horticulture. *Precision Agriculture*, 23(1):111–125, 2022.

[61] Palantir. The future of drone navigation, 2025. Accessed: 2025-05-26.

[62] Yanwei Pang, Yuan Yuan, Xuelong Li, and Jing Pan. Efficient hog human detection. *Signal Processing*, 91(4):773–781, 2011.

[63] Simon G Potts, Jacobus C Biesmeijer, Claire Kremen, Peter Neumann, Oliver Schweiger, and William E Kunin. Global pollinator declines: trends, impacts and drivers. *Trends in Ecology & Evolution*, 25(6):345–353, 2010.

[64] V. J. Rehna and Mohammad Nizamuddin Inamdar. Impact of autonomous drone pollination in date palms. *International Journal of Innovative Research and Scientific Studies*, 5(4):297–305, 2022.

[65] Innovations Report. How drones use spatial ai to navigate their environment, 2025. Accessed: 2025-05-26.

[66] Ricardo Salomón-Torres, Robert Krueger, Juan Pablo García-Vázquez, Rafael Villa-Angulo, Carlos Villa-Angulo, Noé Ortiz-Uribe, Jesús Arturo Sol-Uribe, and Laura Samaniego-Sandoval. Date palm pollen: Features, production, extraction and pollination methods. *Agronomy*, 11(3):504, 2021.

[67] Edward Singh, Aashutosh Pratap, Utkal Mehta, and Sheikh Izzal Azid. Smart agriculture drone for crop spraying using image-processing and machine learning techniques: Experimental validation. *IoT*, 5(2):250–270, 2024.

[68] Nagendra Singh, Akhilesh Kumar Sharma, Indranil Sarkar, Srikanth Prabhu, and Krishnaraj Chadaga. Iot-based greenhouse technologies for enhanced crop production: a comprehensive study of monitoring, control, and communication techniques. *Systems Science & Control Engineering: An Open Access Journal*,

12(1):2306825, 2024.

[69] Rajan Singh and Ankit Sharma. Agriculture 4.0: Use of unmanned aerial vehicles in precision farming. *Computers and Electronics in Agriculture*, 168:105121, 2020.

[70] Chong Tao and Baoyuan Liu. Distributed coordinated motion control of multiple uavs oriented to optimization of air-ground relay network. *Scientific Reports*, 14(31501), 2024.

[71] Bedir Tekinerdogan. Strategies for technological innovation in agriculture 4.0. Talk/Presentation at "Strategies for Technological Innovation in Agriculture 4.0", Istanbul, 2018. Accessed on April 8, 2025.

[72] Yeong Sheng Tey and Mark Brindal. Factors influencing the adoption of precision agricultural technologies: a review for policy implications. *Precision Agriculture*, 13:713–730, 2012.

[73] Yunjie Tian, Qixiang Ye, and David Doermann. Yolov12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.

[74] Pratap Tokekar, James Hook, David Mulla, and Volkan Isler. Sensor planning for a symbiotic uav and ugv system for precision agriculture. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5321–5326. IEEE, 2016.

[75] Maria Trigka and Elias Dritsas. A comprehensive survey of machine learning techniques and models for object detection. *Sensors*, 25(1):214, 2025.

[76] Bas Vergouw, Huub Nagel, Geert Bondt, and Bart Custers. *Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments*, volume 27, pages 21–45. 10 2016.

[77] Rehna V.J and Mohammad Nizamuddin Inamdar. Impact of autonomous drone pollination in date palms. *International Journal of Innovative Research and Scientific Studies*, 5(4):297–305, 2022.

[78] Sjaak Wolfert, Lan Ge, Cor Verdouw, and Marc-Jan Bogaardt. Big data in smart farming – a review. *Agricultural Systems*, 153:69–80, 2017.

[79] Jian Yang, Rong Chen, and Sanghoon Lee. Development of a micro-drone pollination system for greenhouse crops. *Biosystems Engineering*, 200:123–131, 2021.

[80] Kun Yue. Multi-sensor data fusion for autonomous flight of unmanned aerial vehicles in complex flight environments. *Drone Systems and Applications*, 12(1):1–15, 2024.

[81] Zhaoyu Zhai, José Fernán Martínez, Victoria Beltran, and Néstor Lucas Martínez. Decision support systems for agriculture 4.0: Survey and challenges. *Computers and Electronics in Agriculture*, 170:105256, 2020.

[82] Changying Zhang and John M Kovacs. Precision agriculture a worldwide overview. *Computers and Electronics in Agriculture*, 36(2):113–132, 2012.

[83] Lin Zhang, Xiaoyu Wang, and Jing Li. Application of uavs in agricultural pollination: An overview. *Journal of Agricultural Engineering Research*, 63(4):301–308, 2018.

[84] Qin Zhang, Minghua Wang, and Ning Wang. Precision agriculture a worldwide overview. *Computers and Electronics in Agriculture*, 36(2-3):113–132, 2002.