UiT The Arctic University of Norway

Faculty of Science and Technology
Department of Computer Science

AI in the Sky: Diverse Approaches to Drone Swarm Command, Control, Connection and Communication

Modhubroty Dey Barnile

Master thesis in Applied Computer Science
15 May 2024

Abstract

This master's thesis investigates alternative control methodologies for unmanned aerial vehicles (UAVs), focusing on integrating advanced human-machine interaction techniques. Traditional remote-controlled and sensor-based systems pose limitations, particularly for users with physical disabilities. This study explores the potential of virtual keyboards, hand gesture recognition, and eye movement tracking as feasible alternatives to provide more accessible, intuitive control options for the swarm of drone operation. The research methodology encompasses the design and testing of three interaction systems. Each system utilizes computer vision and machine learning technologies to translate human gestures or gazes into drone commands. The virtual keyboard allows users to input commands through eye interactions, hand gestures are captured and processed to control drone movements, and eye movements are mapped to specific flight commands. Findings indicate that while these methods offer significant improvements in user accessibility and control precision, they also present challenges. These include the need for precise timing in eye interaction, inaccuracies in gesture recognition due to insufficient training data, and the potential for bias in command interpretation from eye movement datasets. The thesis discusses these challenges and proposes potential improvements, emphasizing the need for balanced training datasets and adaptive learning systems. It also explores the broader implications of this research for cognitive science and smart city applications, highlighting how enhanced UAV control interfaces could contribute to more autonomous and efficient drone operations. This work contributes to the understanding and development more accessible swarm of UAV control systems that leverage human-machine interaction technologies. While not groundbreaking, these advancements offer meaningful insights into the potential for more inclusive and responsive drone technologies in various practical applications.

Acknowledgements

I would like to extend my deepest gratitude to all those who made the completion of this thesis possible. Firstly, I am profoundly thankful to my supervisor, Bernt Arild Bremdal, whose expertise, understanding, and patience, added considerably to my graduate experience. His guidance helped me in all the time of research and writing of this thesis.

My sincere thanks also go to Andreas Dyrøy Jansson, Shayan Dadman and Tom Stian Andersen, who provided me with the facilities necessary to complete my research. I am also grateful to the Department of Applied Computer Science of UiT The Arctic University of Norway for providing the resources needed to conduct this study.

A special word of thanks to my friend ABM Saleheen, whose camaraderie, support, and exchange of ideas were immensely beneficial throughout the research process.

I must express my very profound gratitude to my father, Babul Chandra Dey, my mother, Dipali Dey and my sister, Jogo Broty Dey Diganta, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

Table of Contents

List of Figures

List of Tables

1

Introduction

Unmanned Aerial Vehicles, also known as drones, have brought about major changes in various industries, from agriculture to defence. Nevertheless, the typical methods for managing these airborne devices — such as remote and sensor-based systems — come with restrictions that can be particularly burdensome for people with physical disabilities. However, the typical methods for controlling these devices can present significant challenges for individuals with physical disabilities for several reasons:

Operating a standard drone often involves using a handheld remote control with joysticks and multiple buttons. This requires fine motor skills to manipulate the controls accurately, which can be difficult or impossible for those with limited hand and finger mobility.

Drones generally require visual monitoring, either by watching the drone itself or through a camera feed on a control screen. This can be a barrier for individuals with visual impairments who may not be able to effectively monitor the drone's position or interpret visual data from the drone's camera.

Managing a drone can also require moving around to maintain the line of sight or travelling to different locations to launch or retrieve the drone. This can be a challenge for people with mobility issues, such as those who use wheelchairs or have limited stamina.

The interfaces used to control drones can be complex, involving simultaneous inputs and monitoring. This complexity can be a hurdle for individuals with cognitive disabilities who might find the systems overwhelming or difficult to understand.

While sensor-based systems offer some autonomy in drone navigation, they still often require initial setup, monitoring, or intervention that may not be fully accessible to all users.

Accessible drone control technologies, such as head trackers, finger trackers, and highly featured remotes, are often significantly more expensive than standard equipment. These devices are designed to allow individuals who cannot use traditional controls to operate drones by translating physical movements (like head tilts or minimal finger motions) into drone commands.

Learning to use these advanced control systems effectively can require significant time and effort, especially if the user adapts to a new disability or has no prior experience with similar technology. The training itself can be a costly and resource-intensive process. This thesis thoroughly investigates different approaches to UAV control that deviate from traditional practices.

The area of swarm robotics is intricate, encompassing an understanding of concepts like defining swarm behaviour, the potential necessity for swarms to reach a specific size, and the specifications and attributes of swarm systems. In a swarm, the failure of a single drone does not terminate the operation, as the rest of the swarm can continue the task. This redundancy ensures more reliable operation, which is crucial in critical applications such as emergency response. It also reduces pressure on the user to perform urgent manual interventions.

Swarms can cover larger areas more quickly and thoroughly than a single drone. This is particularly useful in applications like agriculture, search and rescue, or environmental monitoring, where large terrain or dispersed objects need to be covered. For users with physical disabilities, this means achieving broader operational goals without physically moving over large distances.

Swarms are generally managed through more automated processes, where the user inputs simple commands that are then executed by the swarm as a whole. For example, a user could designate an area to be monitored or an object to be transported, and the swarm would autonomously coordinate to accomplish the task. This reduces the cognitive load and physical demands on the operator.

Drones in a swarm can be assigned specialized roles according to the task at hand, such as surveillance, delivery, or environmental sensing. This division of labour allows complex tasks to be broken down into simpler, more manageable components, which can be easier for users with disabilities to oversee.

This study focuses on exploring ways to enhance the accessibility of UAVs using computer vision technologies. By utilizing OpenCV and TensorFlow, different control systems are suggested and evaluated in this research. These systems interpret eye and hand movements, converting them into specific commands for groups of drones. This comprehensive method opens up opportunities for people with physical disabilities to efficiently direct and manage UAVs, overcoming a capability that has traditionally been inaccessible to them.

This study explores the basics of drone functioning, the existing constraints in swarms of drone and human communication, and the anticipated technological progress that these novel approaches could bring. By implementing practical applications and conducting thorough testing, this investigation assesses the viability and efficiency of virtual control systems. The prospect for these systems to not just improve the operational abilities of UAVs but also make their usage more accessible is a central focus of this thesis.

1.1 Problem Statement

Despite the transformative impact of Unmanned Aerial Vehicles (UAVs) across various industries including agriculture, environmental monitoring, and delivery systems, current control mechanisms predominantly necessitate manual operation through handheld devices requiring fine motor skills, visual monitoring, and physical stamina. These requirements present substantial accessibility barriers for individuals with physical disabilities. Additionally, the high cost and limited availability of specialized, accessible control technologies (such as head trackers, finger trackers, and advanced remote controls) further exacerbate these barriers, restricting the participation of disabled individuals in UAV operations.
The introduction of drone swarms, managed by algorithms that allow for collective behaviour based on simple user commands, promises a significant reduction in the need for direct, precise control and physical oversight. However, the potential of drone swarms to enhance accessibility for users with physical disabilities remains underexplored. This report aims to investigate how the adoption of drone swarm technology could mitigate existing challenges faced by individuals with physical disabilities, offering them greater opportunities to engage with UAV technologies effectively and independently. The study will assess the extent to which drone swarms can provide more accessible, efficient, and cost-effective solutions compared to traditional single-drone operations, thereby promoting inclusivity in UAV-related fields.

1.2 Research Questions

- What are the feasible alternative methodologies for controlling unmanned aerial vehicles (UAVs) that do not rely on traditional remote or sensor-based control systems?

- In what ways can computer vision technologies be integrated into drone control systems to enhance their operational capabilities?

- Which methods are considered the most effective methods for improving the interaction between humans and drones? And why?

- Which computational techniques should be applied to optimize the command and control sequences of drone swarms to enhance their operational efficiency and accessibility for users with physical disabilities?

1.3 Goals/Objectives

This research aims to address significant gaps in current UAV technology, particularly in terms of accessibility and operational efficiency. The goals are:

- **Explore Alternative Control Systems:** Investigate and develop alternative drone control methodologies that do not rely on traditional remote controls or sensor-based systems, focusing on inclusivity for individuals with physical disabilities.

- **Integrate Computer Vision:** Examine how computer vision technologies can be integrated into UAV control frameworks to enhance real-time data processing, environmental interaction, and autonomous decision-making capabilities.

- **Enhance Human-Drone Interaction:** Identify and evaluate the most effective methods to improve human-drone interaction, aiming to establish more intuitive and accessible control interfaces that can adapt to the user's capabilities and needs.

- **Develop Unified Swarm Control:** Propose and test a unified control system for UAV swarms that allows a single operator to issue commands that are dynamically interpreted and executed by each drone, enhancing the coordination and efficiency of swarm operations in disaster response and other complex scenarios.

- **Assess Technology Integration:** Assess the integration of AI, machine learning, and deep learning in enhancing UAV operability and control, ensuring that these technologies can be practically and safely implemented within existing drone systems.

Through this research, the goal is to contribute to the field of UAV technology by enhancing the usability and functionality of drones, thereby facilitating their broader applicability and more effective deployment across various sectors.

2

Related Works

The background section of this thesis provides a thorough exploration of the evolution and current state of unmanned aerial vehicles (UAVs), their integration into various sectors, and the technological advancements that have shaped their development. This examination spans from the initial military applications of UAVs to their widespread adoption in civilian roles, such as agriculture and commercial transportation, highlighting the transformative impact of these technologies across multiple domains. The utilization of unmanned aerial vehicles, commonly referred to as drones, has experienced notable growth in recent years across a variety of sectors. This expansion in drone use has sparked progress in technology and uses, establishing drones as essential components within agricultural practices and commercial transportation. While current usage involves remote control systems, the development of a deep learning-based control system would signify a significant advancement in human technological achievement within the drone industry. [1]

2.1 Unmanned Aerial Vehicles(UAVs) Development and Application

Early studies focused on the adoption of UAVs in military operations, emphasizing advancements in endurance, payload capacity, and autonomy. Through technological evolution, miniaturization of sensors, and improvements in bat-

tery technology, UAVs began to see civilian applications, expanding the breadth of research into areas such as aerial photography and surveying [2].
Considerable research has been devoted to communication and control systems for UAVs. Initial systems relied heavily on line-of-sight radio control, with advancements leading to satellite communication enabling beyond-line-of-sight operations. Recent studies investigate the use of cellular networks for UAV communication, offering enhanced bandwidth and reliability [3].
The push towards fully autonomous UAVs has driven research into sophisticated navigation systems. Studies by Kim et al. (2017) introduced machine learning algorithms for obstacle avoidance and path planning [4], while other research has focused on the integration of computer vision and sensor fusion for real-time navigation and environment mapping [5].
Cutting-edge research focuses on enhancing UAV endurance through solar power, exploiting AI for dynamic flight control, and developing UAVs capable of operating in challenging environments like subterranean spaces or dense urban areas [6].
The literature on UAVs indicates a rapidly growing field with diverse research areas. Advancements in technology are paralleled by an expanding scope of applications, underscoring the transformative potential of UAVs across multiple sectors of society.

2.2 Swarm Technology and Collective Behaviour

Murmuration is a natural phenomenon commonly observed in flocks of starlings, wherein numerous birds exhibit intricately coordinated flight patterns [7]. This collective behaviour not only captivates observers but also fulfils multiple purposes for the birds, including safety from predators, heat conservation, and potential communication and joint decision-making. Swarm robotics represents a fascinating area of research that draws inspiration from natural swarms, such as those observed in ant colonies or flocks of birds, to design robotic systems capable of achieving complex tasks through decentralized coordination and cooperation [8]. Swarm robotics is a field of multi-robotics in which a large number of robots are coordinated in a distributed and decentralised way. It is based on the use of local rules, and simple robots compared to the complexity of the task to achieve, and inspired by social insects. A large number of simple robots can perform complex tasks in a more efficient way than a single robot, giving robustness and flexibility to the group [9] The concept of UAV swarms, where multiple drones operate in coordination, has been a significant area of exploration. Swarm technology facilitates complex missions through distributed sensing and collective behaviour. The research by Kumar and Michael (2012) provided foundational insights into the algorithms that govern swarm behaviour, focusing on collective decision-making and flight formations[10].

2.3 Role of Computer Vision

Image processing involves obtaining essential data from images and videos, as well as executing tasks like identifying objects in real-time videos. This encompasses the processes of capturing, interpreting, comprehending, and manipulating objects within visually perceived images and videos.
Although computer vision is a relatively recent concept, OpenCV operates based on human vision principles. [11] By leveraging the power of OpenCV and its extensive range of capabilities in image processing, feature detection and matching, as well as object detection, industries can enhance their efficiency, accuracy, and productivity. [12]
OpenCV (Open Source Computer Vision Library) presents an exemplary solution for projects involving gaze detection and hand gesture recognition in drone swarms, due to its robust capabilities and wide-ranging functionality in the realm of computer vision:

- **Rich Set of Features:** OpenCV offers an extensive suite of over 2,500 algorithms that are specifically designed for image processing and computer vision [13]. This includes advanced tools for real-time image and video analysis, feature detection, and object tracking—essential capabilities for accurately interpreting eye and hand movements. These functions are crucial for developing systems that can effectively detect and track the eye region and hand shapes, which are integral to controlling drone swarms through natural user gestures.

- **Performance and Real-time Processing:** Optimized for high performance, OpenCV excels in the rapid processing of images and video streams[13]. This efficiency is vital in drone operations, where timely and precise command interpretation is necessary to maintain operational effectiveness and safety. OpenCV ensures that gaze and gesture commands are processed quickly, minimizing latency and maximizing responsiveness.

- **Cross-Platform and Language Support:** Supporting multiple programming languages such as Python, Java, and C++, and compatible across various operating systems including Windows, Linux, and macOS, OpenCV facilitates seamless integration into diverse drone control frameworks [13]. This versatility is beneficial for integrating OpenCV-based vision systems into different hardware or software environments, ensuring broad applicability and adaptability.

- **Pre-trained Models and Customizability:** OpenCV includes a variety of pre-trained models that can be utilized for rapid feature detection and object recognition tasks [14], as well as the flexibility to train custom models to meet specific project needs. This adaptability is crucial for tailoring the system to recognize specific hand gestures or eye movement patterns that are unique to controlling drone swarms.

Given these advantages, OpenCV stands out as a highly suitable choice for implementing advanced computer vision systems in UAV applications, particularly those requiring sophisticated and responsive interaction mechanisms. This capability significantly enhances the accessibility and operational efficiency of drone swarms, positioning OpenCV as a vital tool in the development of innovative UAV technologies.

2.3.1 Gaze Detection

Gaze tracking in OpenCV involves detecting the direction of a user's gaze and the level of their concentration. It typically requires using computer vision and image processing techniques to analyze eye orientation and locate pupil position. Initially, facial recognition is employed to find the eyes in the image, followed by specialized algorithms for each eye to estimate where they are looking. Common methods include Hough Transforms for pupil detection and geometric or learning-based models for gaze vector estimation. These approaches can be further enhanced by integrating machine learning algorithms such as support vector machines or convolutional neural networks to improve accuracy and consistency. The collected gaze data is important for applications like enhancing user interface accessibility, supporting psychological research, and improving interaction within virtual environments. [15]

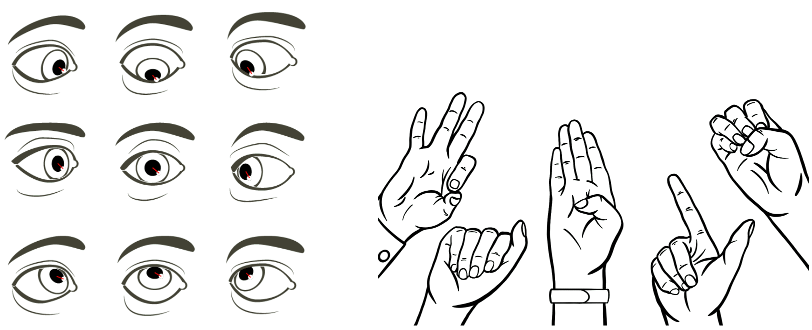


Figure 2.1: The figure indicates eye tracking and hand gesture recognition for intuitive and accessible drone control, enhancing interaction for users with physical disabilities [16] [17].

2.3.2 Hand Detection

Hand detection in OpenCV involves identifying and tracing the hand within a video frame or a sequence of images. This is achieved by first isolating the hand region through skin colour segmentation, which often entails converting the colour space from RGB to HSV for better modelling of skin colour distributions. HSV stands for Hue, Saturation, and Value, and it is a colour space that represents colours in a way that is more aligned with how humans perceive and describe colours. This makes it particularly useful in image processing tasks, such as skin colour segmentation in hand detection. Following segmentation, algorithms such as contour detection can be used to identify the contours of the hand. Additional processing techniques like applying morphological operations such as erosion and dilation help refine the shape of the hand in the image. Advanced approaches may employ pre-trained deep learning models based on neural networks for precise detection and tracking of hands, supporting applications in sign language interpretation, virtual reality experiences, and interactive systems. These models are usually trained on extensive datasets and have the ability to recognize various hand positions and movements under different lighting conditions [18].

Both gaze and hand detection are integral parts of human-computer interaction, allowing for the development of more intuitive and accessible technology. OpenCV provides a robust framework for developing these applications, combining traditional image processing techniques with advanced machine learning models.

2.4 Literature Review

As discussed by Eduard Graells Pina delves into the many significant facets of UAV swarm design, including swarm architecture, communication protocols, navigation, trajectory planning, and work allocation [19]. It highlights how UAV swarms may increase task efficiency, improve reliability by using redundancy, and carry out intricate tasks that individual UAVs cannot. This entails utilizing cutting-edge algorithms for self-organization, investigating the application of open-source tools for analysis and simulation, and considering both decentralized and centralized methods for communication and control among UAV swarms. The master thesis of Eduard Graells Pina also emphasizes the significance of fault tolerance, adaptability, and scalability in UAV swarm operations, and it suggests creative approaches to task delegation and execution that make the most of UAVs' combined capabilities.
According to a book of Hamann, swarm robotics is the study of how to make

robots collaborate and collectively solve a task, that would otherwise be impossible to solve by a single individual of these robots. This true collaboration is the beauty of swarm robotics, it is the ideal of teamwork. Every swarm member contributes equally and everyone shares the same higher-level objectives [20].

The literature on UAV navigation and control technologies highlights a growing interest in leveraging AI to enhance operational efficiencies and address traditional system limitations. The study by J. A. Subramanian et al. represents a significant contribution to the navigation field. It outlines an innovative approach that combines computer vision and photogrammetry to improve the precision and reliability of UAV landings in static environments. This methodology underscores the potential of integrating AI-driven techniques, such as machine learning algorithms and image processing, to augment UAV capabilities beyond conventional GPS-based systems. Key findings from the paper demonstrate the technical approach for source localization and landing trajectory identification for Autonomous Aerial Vehicle (AAV) landing, leveraging computer vision and photogrammetry techniques. More specifically, utilization of ORB for identifying the landing area and implementing the A* algorithm for optimal trajectory identification. ORB is a fusion of a fast key point detector and BRIEF descriptor with many modifications to enhance performance. Oriented Fast and Rotated Brief (ORB) is an efficient alternative to SIFT or SURF from the perspective of vision programming interface [21]. ORB (Oriented Fast and Rotated Brief), BRIEF (Binary Robust Independent Elementary Features), SIFT (Scale Invariant Feature Transform), and SURF (Speeded Up Robust Features) are key feature detection algorithms in computer vision.

A vision-based approach is introduced by the study of S.Chen et al. Specifically, A real-time automated approach for ensuring proper use of personal protective equipment (PPE) in construction sites using deep learning. Rapidly handling NPU, NMU, NHU, and NVU identification approaches are aligned with object detection. The vision-based approach inspired this research to work with drones .[22]

Honor, a prominent Chinese tech firm, has marked its global presence with the launch of the Magic 6 Pro smartphone, which has an innovative eye-tracking AI feature. This cutting-edge technology lets users control and navigate their cars remotely by gazing at their phone screens[23]. This report served as inspiration to adopt the method.

According to research, Sign language detection on smartphones offers better opportunities for hearing and speech-disabled persons. [24]
Suharjito et al. conducted a thorough examination of application systems for recognizing sign language used by individuals with hearing loss or speech disorders. The study applied an input-process-output framework to assess dif-

ferent approaches to sign language recognition, determining the most efficient method. Furthermore, the research delved into diverse acquisition methods and classification techniques, outlining their strengths and weaknesses. This extensive analysis provides valuable perspectives for researchers aiming to enhance sign language recognition systems [25].

A proposed review of intelligent gloves for converting sign language into speech for the mute community lacked comparisons across different research papers. The focus was primarily on a single method, specifically gesture recognition using glove-based technology [26].

Similarly, The article of Galvan-Ruiz also discusses the viewpoint and development of gesture recognition in sign language. It evaluates various gesture recognition tools over time, highlighting key attributes and successful recognition rates. The study suggests that Leap Motion is a cost-effective, user-friendly option with accurate hand detection for sign language applications [27].

Smartphones with multiple cameras have led researchers to investigate their use in recognizing sign language through visual cues. In this approach, the smartphone's camera captures hand gestures which are then processed to identify signs and produce text or speech output. However, using vision-based methods may involve trade-offs in accuracy compared to sensor-based approaches. This is due to challenges such as variations in lighting conditions, sensitivity to skin colour, and complex backgrounds in images. This study examines various vision-based approaches and the associated datasets [28].

The paper of RA Ziar et al. presents a comprehensive review of academic literature, focusing on the utilization of smartphones for detecting and interpreting sign language as assistive tools for individuals with speech disorders. The study highlights the potential advantages of creating a universal sign language to simplify translation processes and reduce associated complexities and costs. It proposes a shift towards using affordable devices such as smartphones that are socially acceptable, in place of more expensive or cumbersome wearable technologies. Ultimately, this paper aims to guide future research and development aimed at improving real-time translation, ensuring privacy during translation, and enhancing gesture recognition under various lighting conditions. It offers a framework for advancements in accessibility technology catering to individuals with speech disabilities [24].

In a study, Evaluating the accuracy and generalization capabilities of four distinct machine learning models—Decision Tree, Gradient Boosting Decision Tree (GBDT), AlexNet, and LeNet—in classifying various sitting postures. Utilizing pressure data captured by a flexible sensing fabric, which records the distribution of human body pressure in different sitting positions, these models serve as classifiers to process and interpret the input data. The predictive outcomes from these classifiers are then transmitted to control a vehicle in a simulation environment via the TCP/IP protocol. This innovative approach not only demonstrates the potential of machine learning algorithms in understanding and utilizing human posture data but also explores the practical application of such data in real-time vehicle control, highlighting the intersection of ergonomic design and automated systems. [29]

Eye tracking and blink recognition algorithms are used in various applications on mobile devices, such as protecting against spoofing attempts in facial recognition systems. [30].

The Viola-Jones algorithm is created to detect objects using cascade classifiers based on Haar features. These classifiers efficiently scan eye images by utilizing specific Haar characteristics, typically 2 or 3 rectangles with dark areas indicating the eyes. The cascading process reduces computational burden and allows for focused detection calculations within subwindows that are likely to contain the eye image. Initially, a rapid sliding window identifies a subwindow using a two-feature classifier, followed by the inclusion of additional features to enhance classification within that subwindow. OpenCV is widely recognized for integrating the popular implementation of the Viola-Jones algorithm [31].

This master's thesis fills these gaps by combining iris recognition and hand gesture commands to create a control system that prioritizes security, accuracy, and accessibility. Unlike other methods, this approach directly addresses the need for an intuitive yet precise control interface adaptable to users' capabilities and requirements, including those with physical disabilities. A new approach is presented to address the usual constraints of UAV control systems, leading to enhanced usability and expanded functionality of drones for a greater variety of applications in different industries. This method is deemed most suitable as it aptly synthesizes the strengths of AI, computer vision, and ergonomic design into a unified system poised to tackle previously identified primary challenges effectively. This comprehensive background sets the stage for understanding the current landscape and future potential of UAV technology. By examining the progression from basic drones to highly sophisticated autonomous and swarm systems, this section not only contextualizes the research within broader technological trends but also aligns it with cutting-edge developments in computer vision and AI. This foundation is crucial for appreciating the subsequent discussions on innovative drone control methodologies and their implications for future applications in smart cities, cognitive science, and beyond.

Table 2.1: Summary of Literature Review

Study	Method	Result	Relation to Proposed Work
Graells Pina [19]	Swarm architecture, communication protocols, navigation, trajectory planning, work allocation	Highlighted the efficiency, reliability, and complexity management benefits of UAV swarms.	Provides foundational insights into swarm control mechanisms, relevant for integrating advanced control systems in UAVs.
Hamann [20]	Theoretical study on swarm robotics	Discussed the collaborative potential and teamwork ideal in swarm robotics.	Underlines the importance of collaboration in swarm systems, applicable to UAV swarm control for enhanced teamwork and task distribution.
Subramanian et al. [21]	Computer vision, photogrammetry, ORB, A* algorithm	Improved precision and reliability of UAV landings in static environments.	Illustrates advanced navigation techniques that can be integrated into the UAV control system for precision.
Chen et al. [22]	Deep learning for PPE detection	Real-time approach for monitoring PPE usage on construction sites.	Suggests the potential of real-time, vision-based monitoring systems that could enhance drone operational safety and compliance.
Honor [23]	Eye-tracking AI technology	Enabled remote car control via eye-tracking on smartphones.	Inspires the incorporation of eye-tracking technology for intuitive UAV control interfaces.
Alam [24]	Sign language detection on smartphones	Enhanced communication options for hearing and speech-disabled persons.	Points to the accessibility features that can be integrated into UAV systems for better user interaction.
Zhong et al. [29]	Machine learning models (Decision Tree, GBDT, AlexNet, LeNet)	Classified various sitting postures using pressure data for vehicle control.	Demonstrates the application of machine learning to interpret human physical data, directly relevant to gesture-based UAV control.

Suharjito et al. [25]	Input-process-output framework	Evaluated various sign language recognition systems.	Highlights the importance of efficient data processing methods, applicable in UAV control systems for improved user interaction.
Sohel Rana [26]	Gesture recognition using gloves	Lacked broad comparison but focused on glove-based recognition.	Encourages exploring other gesture recognition technologies that could enhance UAV control flexibility.
Galvan-Ruiz [27]	Evaluation of gesture recognition devices	Leap Motion identified as effective for sign language.	Suggests potential use of Leap Motion or similar devices for accurate hand gesture detection in UAV control.
Sharma et al. [28]	Vision-based approach using smartphones	Explored trade-offs in accuracy for sign recognition.	Demonstrates challenges and solutions in vision-based recognition that could be mirrored in UAV systems.
RA Ziar et al. [24]	Review of smartphone use for sign language	Advocated for a universal sign language system using smartphones.	Supports the use of universal and intuitive communication methods, which can be integrated into UAV control.
Zhong et al. [29]	Machine learning models with pressure sensors	Classified sitting postures for vehicle control.	Provides insight into using sensory data for precise control, directly applicable to UAV gesture commands.
Pan et al. [30]	Eye tracking and blink recognition algorithms	Utilized for security in mobile device facial recognition systems.	Emphasizes the importance of eye-tracking for secure and intuitive user interfaces in UAV controls.
Viola-Jones [31]	Object detection with Haar features	Optimized eye detection for efficient processing.	Underlines efficient processing techniques for real-time image analysis in UAV systems.

3 Methodology

This section of the study is designed to tackle the identified challenges and objectives highlighted in the literature review. It focuses on improving the control and interaction functionalities of UAV swarms for individuals with physical disabilities, utilizing advanced computer vision and machine learning methods. To address the diverse tasks performed by UAV swarms, a combined approach is suggested, integrating eye-tracking and hand gesture recognition systems to create a more user-friendly control interface. These systems utilize algorithms for real-time feature detection and machine-learning models for gesture classification, as mentioned in previous research. This strategy aims to overcome the limitations of traditional control systems by combining sensor-based and vision-based technologies to ensure efficient functioning under different environmental conditions while meeting various user needs. The subsequent sections provide detailed information about specific technologies, procedural steps involved in developing this innovative UAV control framework, integration strategies for these technologies, and anticipated improvements in both UAV swarm operability and user interaction capabilities.

3.1 Tools

The study utilized a mix of Python, MATLAB, and Blender for the development and evaluation of UAV control systems. Python was leveraged for scripting and automating data processing procedures, while MATLAB offered powerful tools for algorithm creation and numerical computation essential to modelling and simulations. Blender, an open-source 3D creation suite, played a pivotal role in producing intricate visual simulations and animations depicting the UAVs' flight paths. This combination of software enabled a thorough method to design, analyze, and visualize the intricate dynamics present in UAV control systems.

3.1.1 Computational Tools and Framework

This section outlines several essential libraries and models that are invaluable for various applications in computer vision and machine learning. These tools offer a range of functionalities, from image and video processing to machine learning and multimedia application development.

- **CVzone HandTracking Module:** This module specializes in detecting and tracking hand landmarks in real-time, utilizing advanced computer vision techniques. It is particularly beneficial for applications involving gesture recognition, sign language interpretation, and human-computer interaction. The module is designed for easy integration into projects without the need for implementing complex algorithms[32].

- **CVzone Classification Module:** The Classification Module in CVzone is adept at performing image classification tasks using pre-trained deep learning models. It streamlines the process of loading models, preprocessing images, making predictions, and interpreting results, thereby enabling efficient classification of objects, scenes, or patterns within images [33].

- **cv2 (OpenCV):** OpenCV is an open-source computer vision library that is pivotal for image and video processing. It supports a wide array of computer vision tasks and offers easy-to-use APIs. OpenCV is extensively utilized in both academic and industrial settings for developing cutting-edge computer vision applications [34].

- **TensorFlow.keras:** TensorFlow.keras is a high-level neural networks API designed for constructing, training, and deploying deep learning models. It provides a user-friendly interface, enabling users to build various types of neural networks for applications such as image classification, object detection, and natural language processing [35].

- **Pyglet:** Pyglet is a cross-platform library suited for developing multimedia applications and games. It supports graphics rendering, audio playback, user input handling, and resource management, making it an excellent choice for creating interactive multimedia projects in Python [36].

- **SciPy Spatial Distance:** As part of the SciPy library, the Spatial Distance module calculates distances between points in multi-dimensional space. This functionality is particularly useful in machine learning for clustering and other distance-based algorithms [37].

- **Dlib:** Dlib is a C++ library with Python bindings that focuses on machine learning and computer vision. It is commonly employed for tasks such as facial recognition, featuring robust and efficient algorithms [38].

- **Matplotlib:** Matplotlib is a Python library for creating a wide range of visualizations, including static, interactive, and animated charts in both 2D and 3D. It is highly regarded for its versatility and ease of use in visual data representation [39].

These tools collectively facilitate a broad spectrum of applications in machine learning and computer vision, underscoring their importance in advancing technology and research in the fields of drones. The integration of these advanced libraries and models has significantly enhanced the computational framework of this project, particularly in computer vision and machine learning. The utilization of CVzone for real-time hand tracking, gaze detection and image classification, combined with OpenCV's strong image processing capabilities, has substantially improved the accuracy and efficiency of gesture recognition mechanisms essential to UAV control. TensorFlow.keras has supported the development of advanced neural network models that enhance the adaptiveness and responsiveness of the control systems. Additionally, incorporating Pyglet and Matplotlib has enabled effective visualization and interaction with simulation environments, providing clear real-time feedback necessary for refining UAV operations. Together, these tools have not only streamlined complex computational processes but also advanced the project towards achieving more nuanced and reliable UAV control systems, showcasing a significant impact on overall success and innovative research outcomes.

3.1.2 Matlab Simulation and Blender Visualization

The MATLAB environment was configured to simulate the flight dynamics of a drone, considering aerodynamic forces, control system responses, and environmental conditions. This table 3.1outlines the essential simulation parameters configured in the MATLAB environment to model the flight dynamics of a UAV. Each element plays a critical role in replicating a realistic flight scenario, considering aerodynamic forces, control system responses, and environmental conditions. The careful configuration of these parameters ensures that the simulation provides valuable insights into UAV behaviour, allowing for the detailed analysis and optimization of its performance and control strategies. The simulation outputs, including trajectory data and motor responses, are critical for assessing the efficacy of the control systems and for visualizing flight paths in real-time or post-processing environments like Blender.

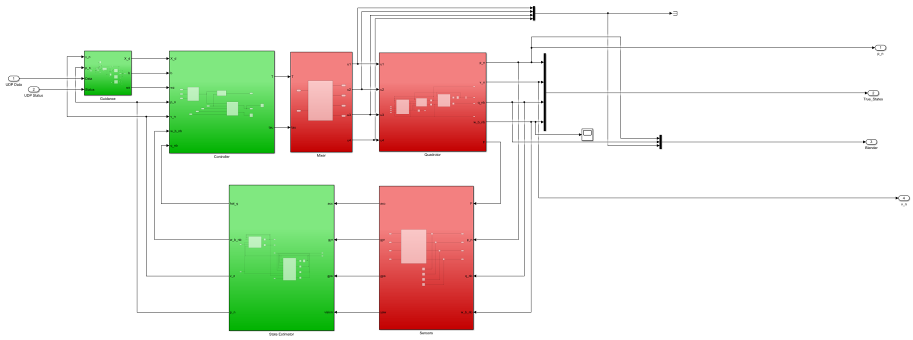


Figure 3.1: The figure illustrates the Model of the drone as shown in the table.3.1

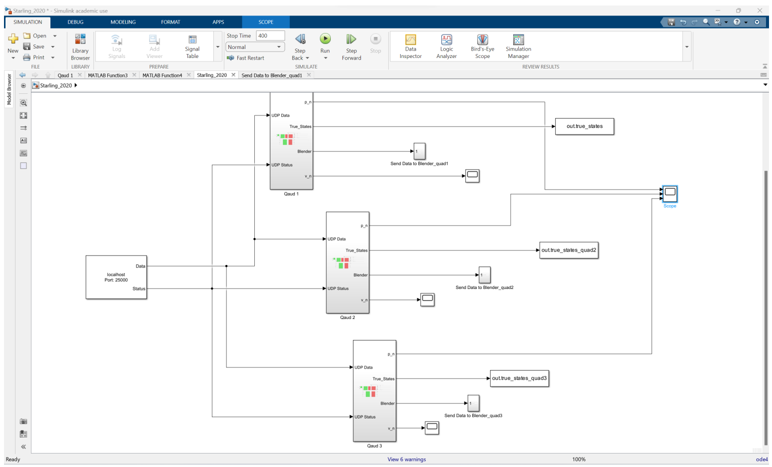
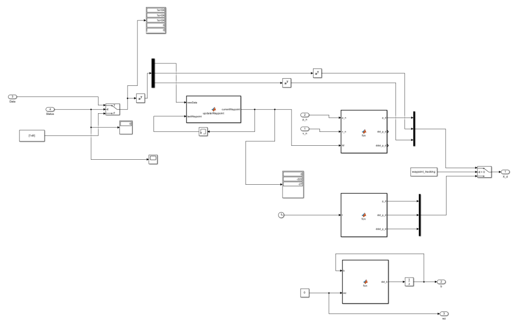


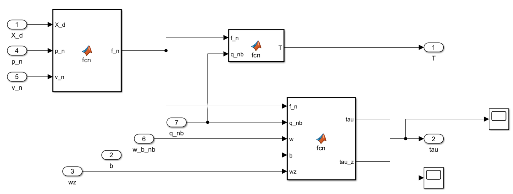
Figure 3.2: The figure presents UDP Port connection for data transfer to the swarm of drones in Matlab

Table 3.1: Mathematical Modelling of Drones

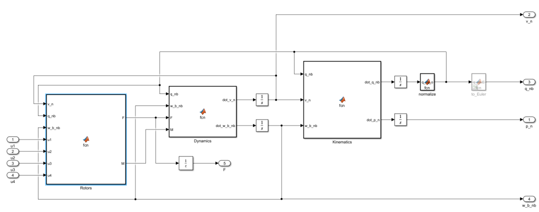
Parameters	Description
Position (p_n0)	Initial position vector of the drone, are critical for starting navigation and trajectory mapping.
Velocity (v_n0)	Starting velocity vector, set to zero to initiate from a static state.
Gravity (g)	Gravitational acceleration constant, are essential for simulating realistic drone dynamics under the influence of gravity.
Mass (m)	Total mass of the drone, affecting its inertial properties and flight dynamics.
Inertia (J)	Diagonal inertia matrix [Jx Jy Jz], fundamental for calculating rotational dynamics and responses to control inputs.
Motor Constants (C_T, C_Q)	Thrust constant (C_T) and torque constant (C_Q) are used for motor output calculations, essential for propelling and manoeuvring the drone.
Rotor Configuration (pr)	Position vectors of the rotors relative to the drone's centre, are crucial for defining the force and torque generation dynamics.
Control Gains (k_p, k_d)	Proportional and derivative gains for the translational and attitude controllers, configuring the responsiveness and stability of the control system.
Guidance System	Defines trajectory tracking settings, influencing how the drone follows waypoints or a defined path.
State Estimator	Initializes the estimated state with positions and orientations, integral for feedback control where precise state information is critical.



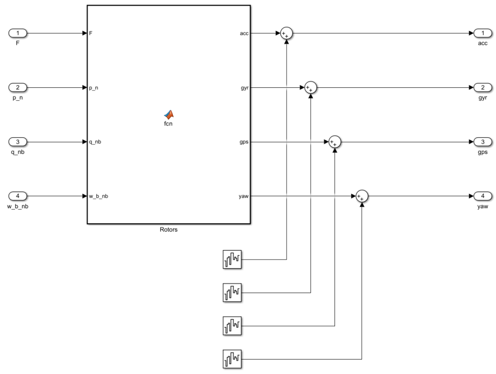
(a) Guidance



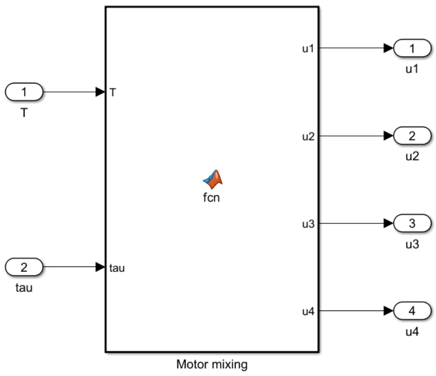
(b) Controller



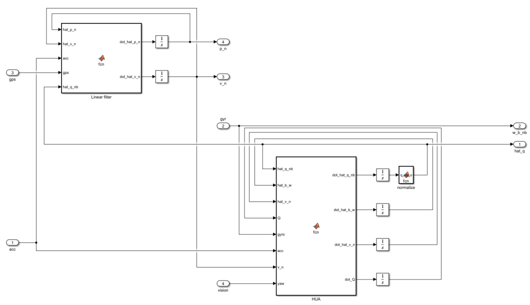
(c) Quadrotor



(d) Sensor



(e) Mixer



(f) State Estimator

Figure 3.3: The figure shows a Matlab Simulink simulation developed based on the UAV mathematical models from the studies of Mahony [40] and Pounds [41]. The control system used in this simulator is based on the reactive control strategies by Tom Stian Andersen [42], enabling the simulation to accurately demonstrate UAV behaviour and control dynamics under various scenarios.

Algorithm 1 Simulation Configuration for UAV Flight Dynamics

Clear MATLAB environment
addpath("math library/")
Initialize simulation time: $t_0 = 0$, $t_{end} = 150$, $t_{step} = 0.001$
Define unit vectors $e1 = [1, 0, 0]'$, $e2 = [0, 1, 0]'$, $e3 = [0, 0, 1]'$
Configure gravitational acceleration $g = 9.81$
Set mass $m = 1.27$
Setup UDP ports for Blender $udp\_port = [300, 310, 340]$
Initialize inertia matrix $J = \text{diag}([Jx, Jy, Jz])$
Define motor parameters and mixing matrix M
Initialize initial conditions and state estimators
Configure control systems parameters
Run simulator
Plot results and export to Blender

Blender was utilized to convert the raw simulation data into a three-dimensional animated model. A custom Python script automated the parsing and application of mathematical data to the drone model within Blender. The key frame-based animation was implemented to accurately reflect the drone's motion over time. The provided algorithm.1 and 2 highlights how position and orientation data from MATLAB were used to animate the drone model in Blender, showing the application of data to generate realistic flight paths. This section discusses the implementation of a quadrotor simulation using MATLAB. The aim is to model the dynamics, design a controller, and simulate the flight of a quadrotor. The dynamics of the quadrotor are modelled considering both translational and rotational movements.

The figure. 3.3 presents a simulation interface developed using Matlab Simulink, designed to reflect the dynamic behaviours and control strategies of unmanned aerial vehicles (UAVs). This simulation environment is built upon the mathematical models articulated in the research conducted by Mahony [40] and Pounds [41], which provide foundational frameworks for multirotor dynamics and UAV physical modelling, respectively. Additionally, the control system implemented within the simulator draws on the reactive control strategies studied by Tom Stian Andersen [42], integrating advanced algorithms to ensure responsive and stable UAV operations under varying conditions. This integration of sophisticated modelling and control systems demonstrates the simulator's capability to accurately replicate and analyze UAV behaviour, serving as a critical tool for testing and refining UAV control techniques.

Algorithm 2 Real-Time Animation of Quadrotors in Blender

```
Remove all existing grease pencil objects
Define maximum rotor angular velocity
Setup UDP sockets for each quadrotor
for each quadrotor do
    Create a non-blocking UDP socket
    Bind to a predefined local port
end for
Define update function for quadrotor position and propeller animation
Start the real-time animation operator
while true do
    Listen for incoming UDP packets on each socket
    if data received then
        Update each quadrotor's position and rotation
        Adjust propellers' rotation based on received data
    else
        Continue listening
    end if
    Exit on user interrupt
end while
Clean up: unregister operator and close sockets
```

3.1.3 Google's Teachable Machine

Google's Teachable Machine is a user-friendly tool that enables users to create machine learning models without any coding experience. This web-based resource allows for the quick and easy training of models using simple drag-and-drop functionality. For this project on UAV control systems, the Teachable Machine proved instrumental in developing and refining the AI-driven control mechanisms necessary for interpreting iris recognition and hand gesture commands.

The tool provided immediate feedback on model performance, allowing for quick adjustments and improvements. This hands-on approach facilitated a deeper understanding of how different inputs affected model accuracy and response times, which are crucial for the real-time operation of UAVs.

In this research, the Teachable Machine served as a critical tool for training datasets efficiently, significantly reducing the time from concept to testing. This capability was especially valuable in demonstrating the practical viability of using advanced AI techniques in UAV control systems.

3.2 Eye Controlled Drone Swarm

Eye movement control for drones is particularly groundbreaking because it taps into natural and ubiquitous human behaviour. Unlike head or finger movements, which require conscious effort and can sometimes be cumbersome or imprecise, eye movements are swift, effortless, and incredibly precise. Our eyes are constantly in motion, reacting instantaneously to our thoughts and surroundings. By harnessing this innate capability, eye movement control offers a direct and fluid interface for drone navigation, significantly reducing the learning curve and making drone operation more accessible to a wider audience.

Computer vision is integral to the methodology described for controlling interfaces through facial interactions, playing a pivotal role in accurately interpreting user gestures as commands within a virtual environment. This process begins with the use of a camera to capture real-time video footage, from which facial landmarks are detected and analyzed using computer vision techniques and algorithms, such as those provided by libraries like OpenCV and Dlib.

The primary task of computer vision in this context is to detect and track facial landmarks—specific points on a user's face, such as the edges of the eyes or the contour of the mouth. This detection is critical for determining subtle changes in facial expressions, such as blinks or gaze direction. For example, by calculating the ratio of distances between certain facial landmarks around the eyes, the system can determine whether the eyes are closed (indicating a blink) or where the user is looking. These metrics are then translated into specific commands ("Up", "Down", "Left", "Right") based on predefined thresholds and mappings.

Moreover, computer vision algorithms help in the stabilization and noise reduction of the input video stream, ensuring that involuntary movements or changes in ambient lighting do not trigger unintended commands. The robustness of facial landmark detection directly affects the system's responsiveness and accuracy, making advanced computer vision techniques crucial for enhancing user experience and interaction quality.

In addition to real-time interaction, computer vision also supports calibration and customization of the control system to individual users. It can dynamically adjust parameters such as the sensitivity of gesture detection based on the user's typical blink duration or gaze stability, further tailoring the interface for optimal usability.

3.2.1 Virtual Keyboard Method

The algorithm.3 processes video input to interact with a virtual keyboard via facial gestures, utilizing detected facial landmarks to command drone swarm simulation through TCP/IP-based control signals. The main operations include

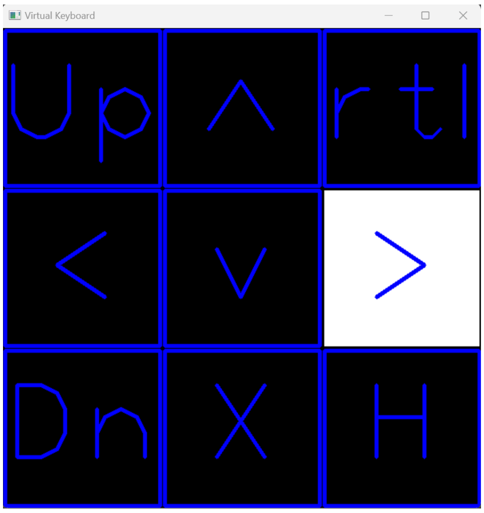


Figure 3.4: This virtual keyboard showcases real-time interaction via eye blinking, where users control the swarm of drones simulation by triggering commands.

gesture recognition through blinking and eye movement, and dynamic keyboard interaction. The facial interaction-based control system is designed to map predefined textual commands (e.g., "Up", "Down", "<", ">") to specific three-dimensional vectors that represent control actions in a virtual environment. Each command corresponds to a unique vector indicating movement direction or action within this space. The function retrieves these vectors from a dictionary based on the input text, printing the type of the retrieved object to ensure data consistency, which is critical for debugging and validation of the command processing pipeline. This simple yet effective mapping and validation mechanism facilitates the translation of facial gestures into precise control signals in real-time applications for drones.

3.3 Eye tracking Technology using datasets

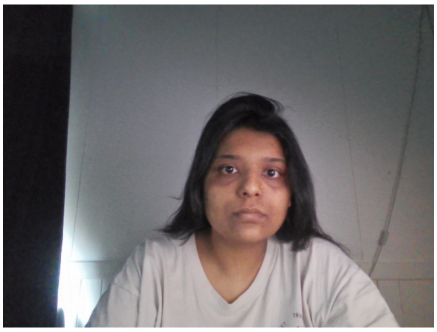
The system is designed to interpret eye movement data to control drones, utilizing a comprehensive model of eye movement behaviour. By comparing real-time captured data against a pre-established model dataset, the system determines appropriate commands for drone navigation and control.

The algorithm.4 and figure.3.5 outlines how the system processes eye movement data to facilitate drone control.

Algorithm 3 Facial Interaction using Virtual Keyboard Control System

```
initialize camera, facial detector, and predictor
load sound file for feedback
create virtual keyboard and command set
define server address and port for TCP/IP communication
while camera is open do
    capture frame and convert to grayscale
    detect faces in the frame
    for each face in detected faces do
        predict facial landmarks
        calculate blinking ratio
        if blinking ratio threshold is exceeded then
            play sound and update command based on active key
            send command via UDP using the current posture data
        end if
        calculate gaze ratio for gaze-based control
    end for
    update virtual keyboard display
    show frame and keyboard
    if escape key is pressed then
        break
    end if
end while
release camera
close all windows
```

(a) Down



(b) Forward

(c) Left

(d) Up

Figure 3.5: The figure presents the types of eye Movement datasets for the swarm of drones

The development and deployment of an advanced facial gesture-controlled interface for drone command initiation incorporates the integration of sophisticated eye-tracking technology and machine-learning algorithms. The system's core functionality is predicated on capturing user eye movements through a high-resolution camera, which is part of a comprehensive eye-tracking apparatus. This setup is crucial for accurately detecting subtle variations in eye position and movement, ensuring precise input interpretation under varied environmental lighting conditions.
To facilitate the effective translation of these eye movements into actionable drone commands, the interface employs a robust machine-learning framework developed using TensorFlow Keras. The model is trained on a diverse dataset comprising approximately 90 images captured from different angles and under varying lighting conditions—ranging from brightly lit environments to dimly illuminated settings. This training approach enhances the model's ability to generalize across different real-world scenarios, significantly reducing the likelihood of performance degradation due to environmental variances.
The training process leverages Google's Teachable Machine, an accessible platform that simplifies the creation of machine learning models without requiring extensive programming expertise. This tool allows for rapid prototyping and it-

Algorithm 4 Drone Control through Eye Movement Analysis

```
initialize camera, eye tracking system
load model dataset of eye movements
define drone control commands and UDP settings for communication
while camera is operational do
    capture eye movement data in real-time
    compare captured data with model dataset
    if match found with specific model pattern then
        identify corresponding drone command
        send command to drone via UDP
    end if
    monitor for any emergency stop signals
    if stop signal detected then
        send stop command to drone
        break
    end if
end while
disconnect camera
terminate communication
```

erative testing of the model with real-time feedback, facilitating the refinement of the system's accuracy in interpreting eye movements as specific commands. Once a particular eye movement pattern is recognized by the system, it is immediately mapped to a predefined drone command, such as looking left or looking down. These commands are then promptly dispatched over a UDP connection, enabling real-time control of the drone with minimal latency. This real-time processing capability is critical for applications requiring immediate responsive actions from the drone, such as navigation through complex environments or precise manoeuvring in response to dynamic external factors.
Overall, the system epitomizes the integration of advanced eye-tracking technologies with cutting-edge machine-learning techniques to create a highly effective, real-time interface for drone control. This setup not only enhances the accessibility and usability of drone technology for a broader user base, including individuals with physical disabilities but also pushes the boundaries of what can be achieved in the realm of interactive and autonomous aerial vehicles.

3.4 Hand Gesture Controlled Drone Swarm

The comprehensive methodology employed in Algorithm.5 for real-time drone control via hand gestures utilizes state-of-the-art computer vision and machine learning technologies.

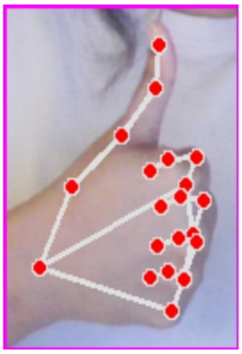
Algorithm 5 Main Processing Loop of Hand Gesture

while true **do**
  Capture a frame from the webcam.
  Detect the presence of any hands in the frame.
  if a hand is detected **then**
    Crop the image around the detected hand.
    Resize and possibly pad the image to fit the input requirements of the classifier.
    Classify the hand gesture.
    Retrieve and send the corresponding command to the drone via UDP.
  end if
  Display the frame and any relevant overlays to the user.
end while

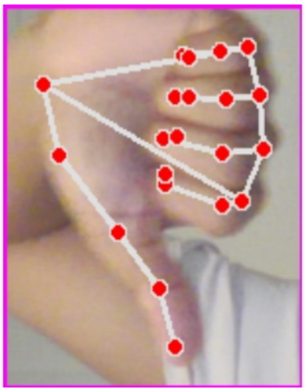
This system is particularly designed to capture, process, and interpret human hand gestures from a live video feed, allowing for intuitive interaction with drones. Steps are:

- **Image Capture and Processing:** Utilizing OpenCV, the system captures live video through a high-resolution webcam. This setup is essential for obtaining clear and precise visual data necessary for accurate gesture recognition. To ensure robustness against various operational challenges such as different lighting conditions and diverse skin textures, the model is trained on a dataset of approximately 1570 images featuring hands from multiple individuals in various lighting scenarios as shown in the figure. 3.6.

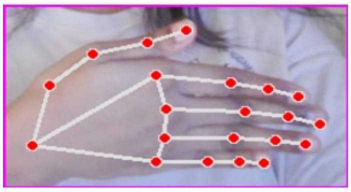
- **Hand Detection and Gesture Recognition:** At the heart of the system lies the HandDetector module, a critical component of the CVzone library, which is adept at detecting hands within video frames. This module utilizes sophisticated algorithms to identify hand regions accurately, despite variations in background or hand orientation. Once a hand is detected, the image is cropped around the detected hand, resized, and possibly padded to meet the input specifications of the gesture recognition model. The TensorFlow Keras model, trained using Google's Teachable Machine, classifies the hand gesture into predefined categories such as "Up", "Down", "Left", "Right", "Land", "Hover", "Front", "Back", etc.



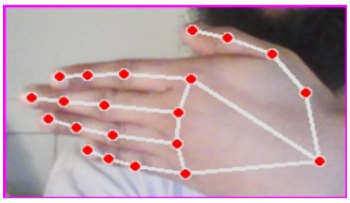
(a) Up



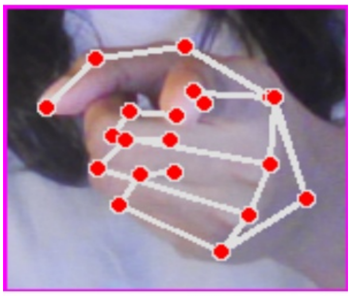
(b) Down



(c) Left



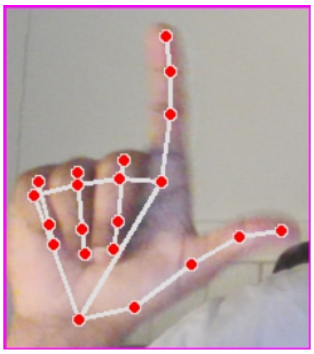
(d) Right



(e) Front

(f) Back

(g) Hover



(h) Land

Figure 3.6: The figure presents the types of sign language datasets for the swarm of drones

This classification process is based on patterns learned from the training dataset, which enables the model to map specific gestures to drone commands effectively.

- **Real-Time Communication:** After recognizing a gesture, the system converts this information into a drone command which is then transmitted via UDP (User Datagram Protocol) to minimize communication delay. This rapid transmission is crucial for real-time applications where any delay could disrupt drone operation.

- **User Interface and Feedback:** The user interface enhances interaction by displaying the live video feed along with overlays that mark detected hands and recognized gestures. This immediate visual feedback is vital for users to adjust their gestures dynamically, thereby fine-tuning their control over the drone movements.

Through the integration of the HandDetector module and advanced image processing techniques, the system achieves a high level of accuracy and responsiveness. This method not only facilitates effective drone control via natural human gestures but also exemplifies the potential of advanced computer vision and machine learning techniques in creating more interactive and accessible technology interfaces. The use of these innovative approaches promises to revolutionize the way to interact with machines, making sophisticated drone operations accessible to a broader audience.

4 Results

This research aimed to revolutionize the control mechanisms of UAV swarms by integrating artificial intelligence with iris recognition and hand gesture command systems. This section presents the findings from three experimental setups designed to assess different methods of UAV control: direct interaction through a virtual keyboard via gaze detection, hand gesture recognition from datasets, and eye movement recognition from datasets. Each method was tested under controlled conditions to ensure the reliability and reproducibility of the results.

4.1 Result using the virtual keyboard with gaze detection

This segment of the results illustrates the interaction of a user with a virtual keyboard, marked by the appearance of the word "BLINKING" in blue. This indicates the system's capability to detect eye blinks as commands to navigate through the virtual keyboard. The keyboard's white light cycles through each key sequentially, pausing for one second on each. For instance, when the light reaches a desired key, such as "Up", the system registers an eye blink as a selection command, subsequently transmitting the three-dimensional command vector [0, 0, 1] to the mathematical model in MATLAB via a UDP port, as detailed in Figure 4.1. Each key corresponds to predefined vector

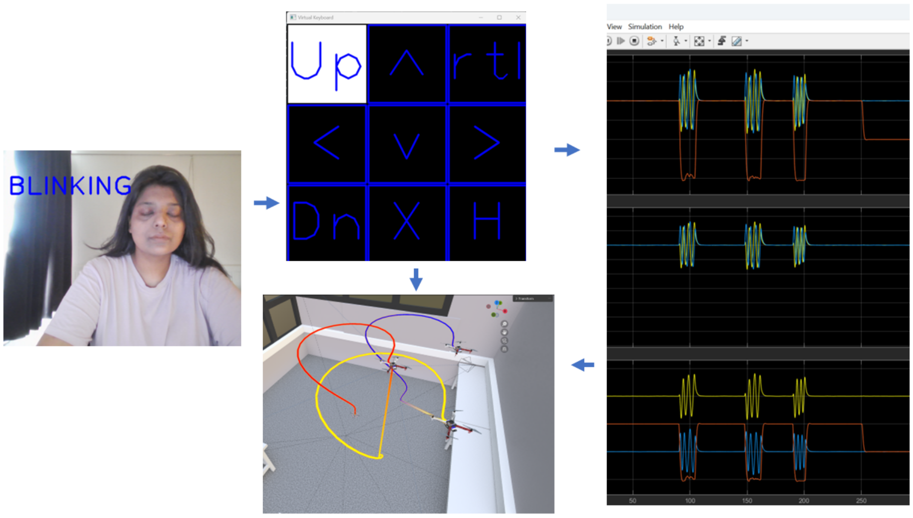


Figure 4.1: Result using virtual keyboard and gaze detection

values.

$$\begin{array}{rlrlrl} \text{"Up"}: & [0,0,-1], & \text{"^"}: & [1,0,0], & \text{"RTL"}: & [0,0,0] \\ \text{"<"}: & [0,-1,0], & \text{"v"}: & [-1,0,0], & \text{">"}: & [0,1,0] \\ \text{"Dn"}: & [0,0,1], & \text{"X"}: & [0,0,0], & \text{"H"}: & [0,0,-1] \end{array}$$

The MATLAB plot displays the drone's movement trajectory along three principal axes: x, y, and z. In this visualization:

- The yellow line represents the x-axis,
- The blue line represents the y-axis,
- The red line represents the z-axis.

These axes are critical for understanding the spatial orientation and movement of the drone in response to the input commands. Each key on the virtual keyboard is intricately mapped to a specific vector that commands the drone's movement in three-dimensional space. The axes are colour-coded to enhance clarity and allow for easy distinction among the movements along different planes. This illustrates the precise path followed by the drone based on the real-time interpretation of user inputs from eye movements as shown in Figure.4.1. The full procedure of each key is given below:

- **Up:** The system interprets the command "Up" as a vector of [0,0,1]. This vector denotes a movement along the negative z-axis, which typically corresponds to movement as Figure. 4.2 in the context of drone operations. The command effectively instructs the drone to move upwards in its operational space, responding directly to the user's input via the virtual keyboard.

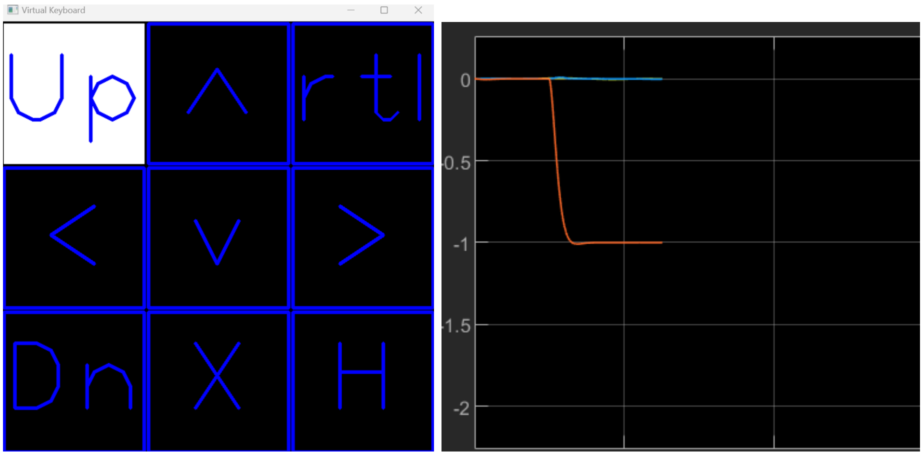


Figure 4.2: The "Up" command, shown in Figure, uses the vector [0,0,-1] to direct the drone upwards along the negative z-axis via the virtual keyboard.

- **Front:** The virtual keyboard is represented in the Figure. 4.3, allows users to input commands such as "Front," which correspond to pre-defined movement vectors. For instance, the "Front" key correlates with the vector [1,0,0], directing the drone to move one unit forward along the x-axis.

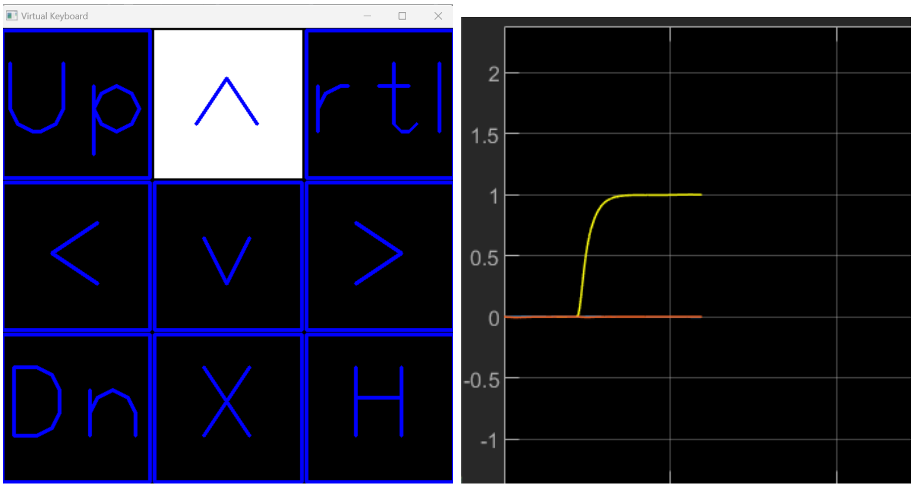


Figure 4.3: The Figure illustrates a virtual keyboard where the "Front" key, associated with the vector [1,0,0], directs the drone forward.

- **Return to Launch:** In the context of drone swarm management, the "Return to Launch" (RTL) command is crucial for ensuring the safety and reliability of operations, especially in scenarios requiring an emergency or planned return of drones to their initial positions.

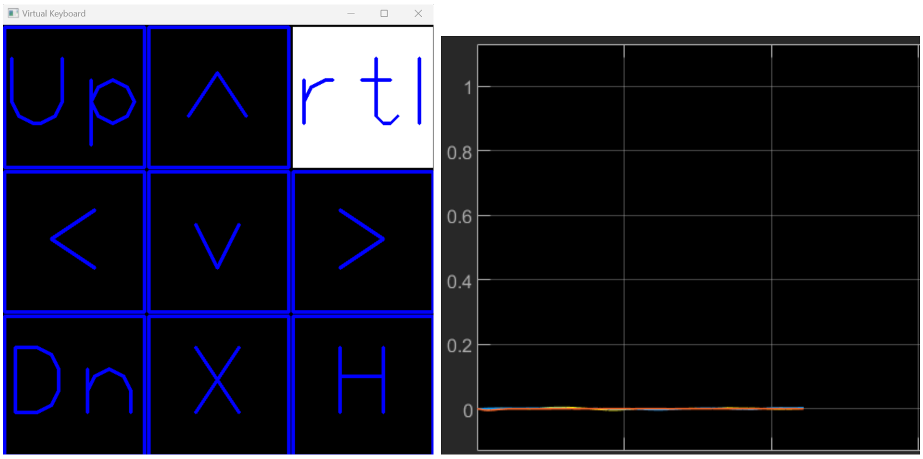


Figure 4.4: The figure represents the command "RTL" means a return to launch for the drone swarms

- **Left:** The diagram illustrates the virtual keyboard, as shown in Figure 4.5, enabling users to enter instructions like "Left," which are associated with predetermined movement vectors. For example, blinking the "<" key corresponds to the vector [0,-1,0], indicating that the drone should move one unit forward along the y-axis.

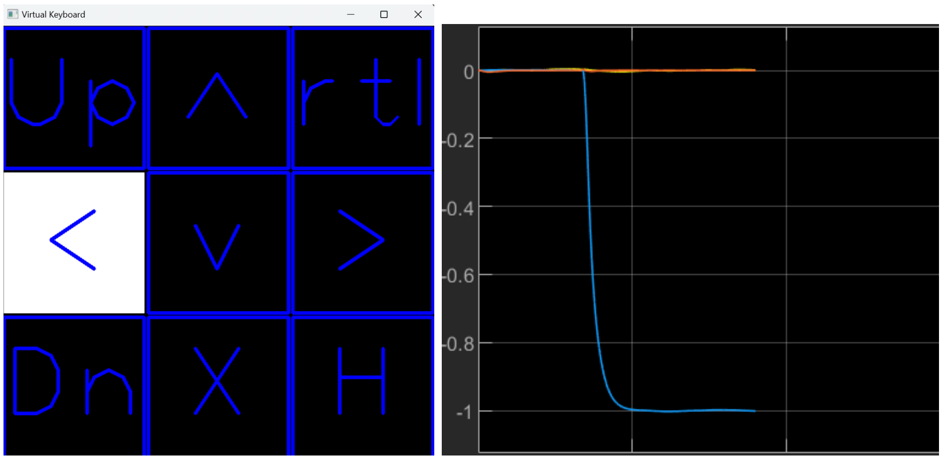


Figure 4.5: The Figure shows a virtual keyboard where blinking the "<" key for the "Left" command corresponds to the vector [0,-1,0].

- **Back:** This illustration presents the virtual keyboard, as depicted in Figure 4.6, allowing users to input commands such as "Back," which are linked with pre-determined motion vectors. For instance, blinking the "v" key is related to the vector [-1,0,0], signalling that the drone should shift one unit forward along the x-axis.

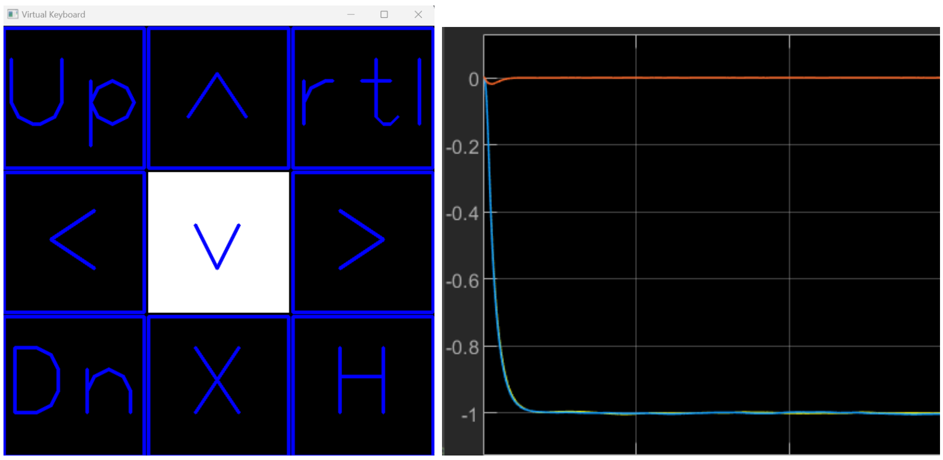


Figure 4.6: The Figure displays a virtual keyboard where the input command is "Back," with the "v" key corresponding to the vector [-1,0,0]

- **Right:** This illustration depicts the virtual keyboard, as depicted in Figure 4.7, that allows users to input commands such as "Right" linked to pre-determined motion vectors. For instance, blinking the ">" key with an eye blink corresponds to the vector which signifies that the drone should advance by one unit along the y-axis.

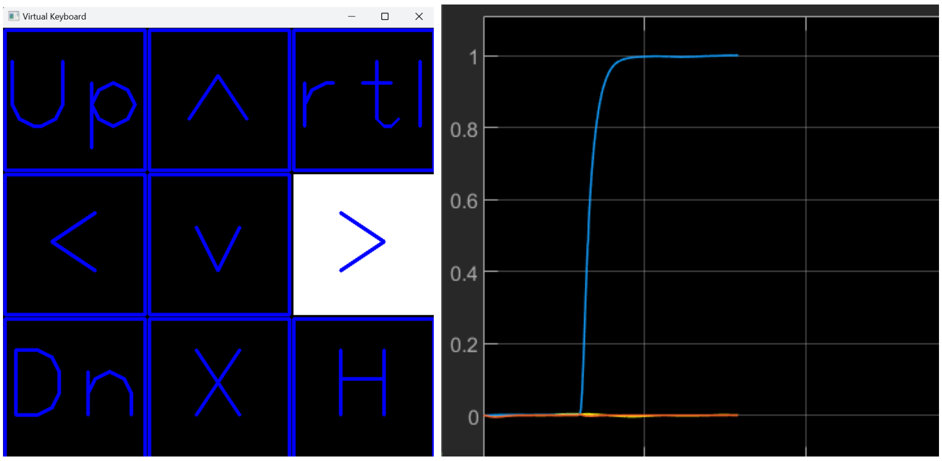


Figure 4.7: The figure presents the Right key and the simulations

- **Down:** This image shows the virtual keyboard, as shown in Figure 4.8, which enables users to enter instructions like "Down," connected to predetermined motion vectors. For example, the eye blinking the "Dn" key corresponds to the vector, indicating that the drone should move forward by one unit along the z-axis.

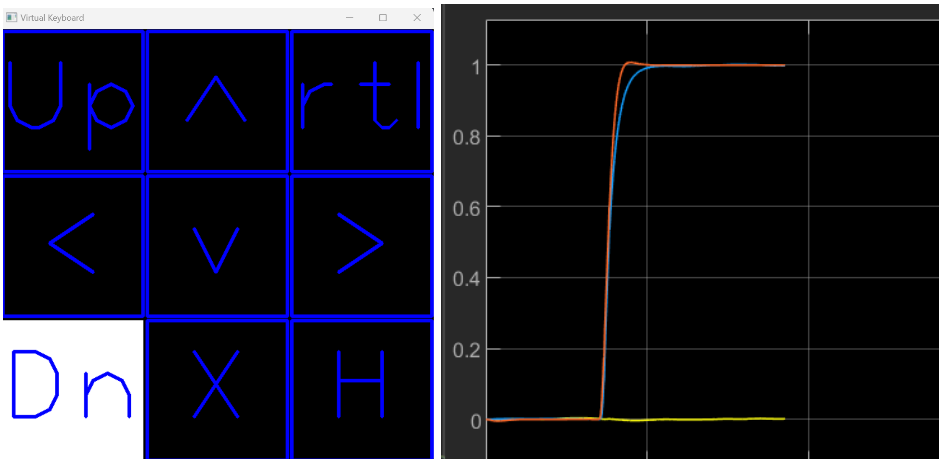


Figure 4.8: The Figure displays a virtual keyboard that allows users to input commands "Down," linked to specific motion vectors.

- **X or Cyclic trajectory:** The displayed graph (Figure 4.9), generated from the MATLAB environment, illustrates the drone's trajectory when the "X" command is executed. This trajectory is defined by a set of parametric equations that modulate the drone's position over time. The blue line represents the drone's sinusoidal movements along the y-axis, indicating lateral shifts. The yellow line traces a sinusoidal pattern along the x-axis, signifying forward and backward movements. The red line remains steady, denoting no change in the z-axis, implying that the altitude is constant during this manoeuvre. When the "X" key is activated, the drone is programmed to follow a cyclic trajectory defined by:

$$\mathrm{p}(t) = [c_1 \cos(c_2 t), c_1 \sin(c_2 t), -2] \tag{4.1}$$

- **First derivative of position vector $\dot{\mathrm{p}}(t)$ (velocity)**:

$$\dot{\mathrm{p}}(t) = [-c_1 c_2 \sin(c_2 t), c_1 c_2 \cos(c_2 t), 0] \tag{4.2}$$

- **Second derivative of position vector $\ddot{\mathrm{p}}(t)$ (acceleration)**:

$$\ddot{\mathrm{p}}(t) = \left[-c_1 c_2^2 \cos(c_2 t), -c_1 c_2^2 \sin(c_2 t), 0\right] \tag{4.3}$$

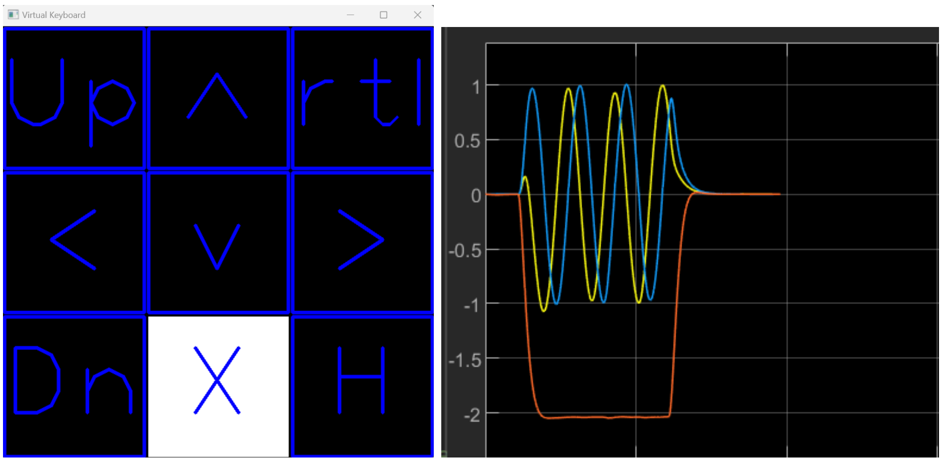


Figure 4.9: The graph in Figure produced using MATLAB, shows the drone's path following the "X" command, defined by parametric equations that dictate its movement over time.

These equations provide a comprehensive description of the drone's kinematic behaviour over time, accounting for the cyclical movements in the horizontal plane and a constant altitude change.

- **Hover:** The diagram depicts the virtual keyboard, represented in Figure 4.10, which allows users to input commands such as "Hover." These commands are linked to predefined movement vectors. For instance, detecting eye blinking in the "H" key corresponds to the vector, signifying that the drone should remain stationary without any movement.

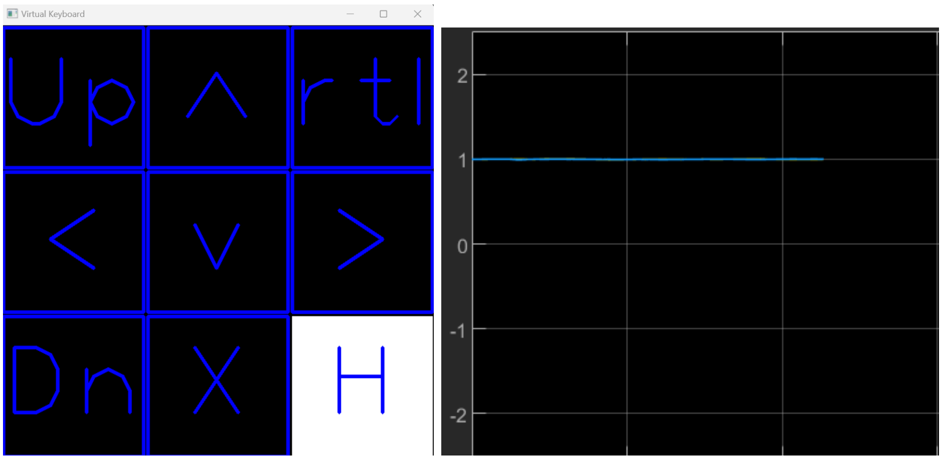


Figure 4.10: The diagram illustrates the virtual keyboard where users can issue commands "Hover" through eye blinking on specific keys such as "H," which corresponds to [0,0,0] vector, maintaining the drone's position without movement.

4.2 Result using datasets of Hand Gesture

This section presents a series of waveform graphs as seen in Figure. 4.11, which depicts the signal or data analysis correlating to the user's hand gestures. These graphs are crucial in showing how gestures are detected, interpreted, and transformed into command signals understandable by the UAV.

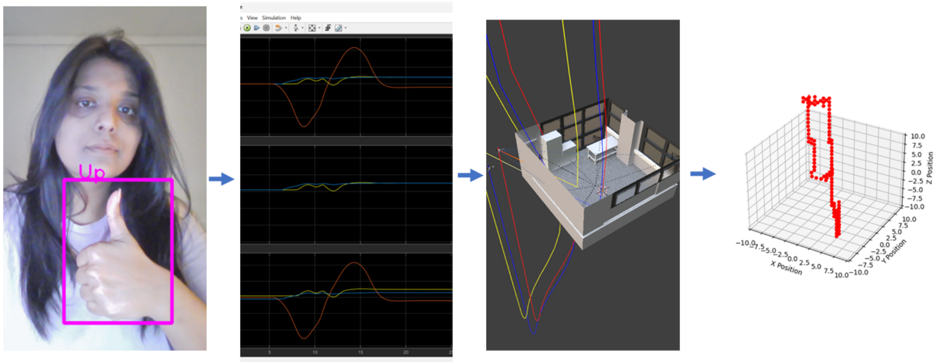


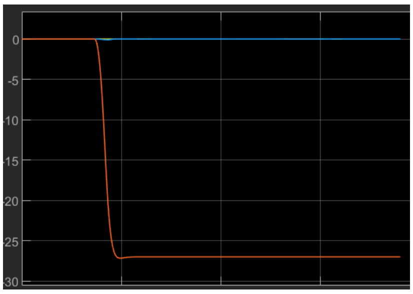
Figure 4.11: The Figure represents the flow of work and the result depending on Sign language datasets, First capture the video, detect the movement of hand, load the datasets and predict the movement, send the value to the Matlab simulator, generate the graph simulation, send the values to the blender visualization for swarm of drones

Upon recognizing the "Up", "Down", "Front", "Back", "Left", "Right", "Land" or "Hover" gestures, the system assigns a pre-defined three-dimensional vector, which denotes a movement for the swarm of drones. This vector is immediately communicated to the drone's control system. The efficacy of this process is depicted in the simulation graph as shown in the Figure. 4.12.

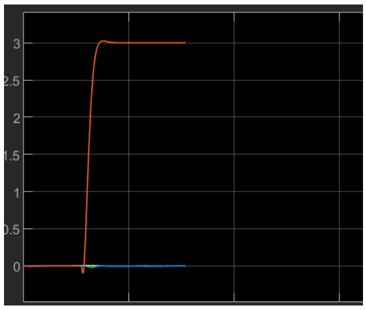
The simulation graph offers a visual representation of the drone's response to the gesture-based command. The sharp transition in the movement captured in the graph demonstrates the system's responsiveness in translating the hand gesture into a precise navigational adjustment. This graphical depiction is crucial for validating the effectiveness of the gesture recognition system and its potential application in real-world drone operations.

4.3 Result using datasets of eye movements

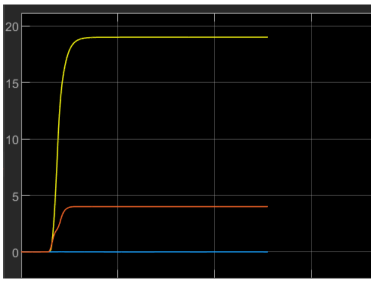
Analysis based on eye movement datasets indicates detection accuracy, such as identifying when the user looks to the left. Concurrently, drone simulation outputs graphical data that reflect real-time feedback or sensor information pertinent to the UAV's performance, responding dynamically to the issued commands. These outputs demonstrate the system's responsiveness and the



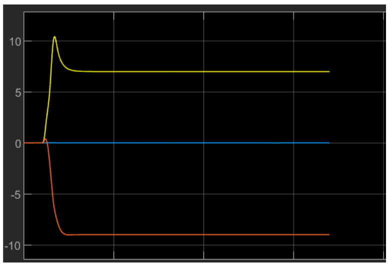
(a) Up



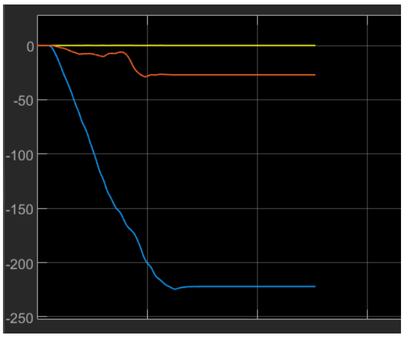
(b) Down



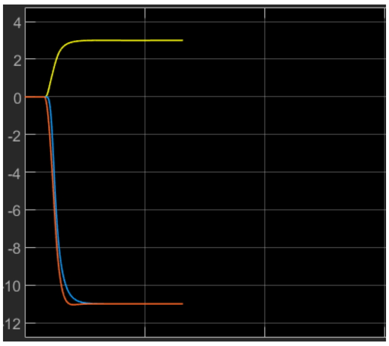
(c) Front



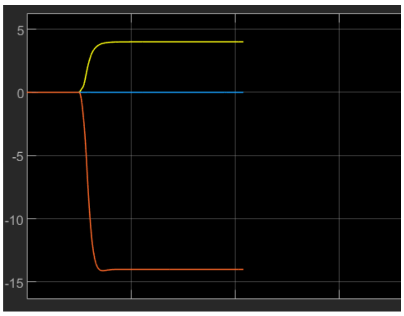
(d) Back



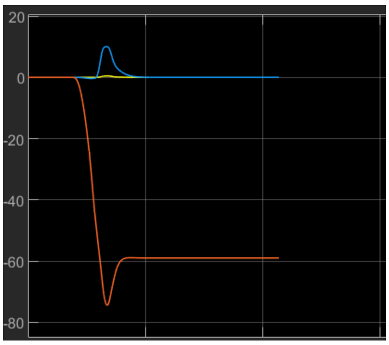
(e) Left



(f) Right



(g) Land



(h) Hover

Figure 4.12: The Figure demonstrates the simulations graph for drones depending on all the Hand Signs

effective translation of eye movements into actionable directives for UAV navigation as Figure.4.13.

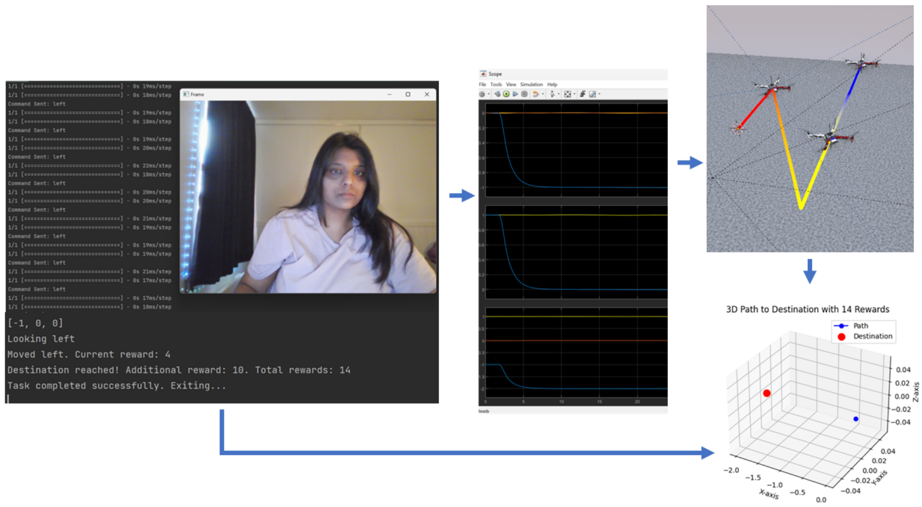


Figure 4.13: The Figure represents the flow of work and the result depending on eye movements datasets, First capture the video, detect the gaze, load the datasets and predict the movement, send the value to the Matlab simulator, generate the graph simulation, send the values to the blender visualization; One destination waypoint was pre-defined, when the drone reach that point, it saved a reward point for swarm of drones

4.4 Mathematical Representation of Drone Swarms Positions

In the context of controlling a drone swarm, the system utilizes a matrix representation to efficiently manage and update the spatial coordinates of each drone within the swarm. This matrix-based approach ensures that each command issued by a user directly translates into a coherent and synchronized adjustment of the positions of all drones simultaneously. Below is an academic explanation and mathematical formalization of this process.
Every member of the drone swarm is denoted by a vector within a three-dimensional coordinate system. The position of each drone can be denoted as:

$$p_i = [x_i, y_i, z_i] \tag{4.4}$$

where p_i represents the position vector of the i^{th} drone and x_i, y_i and z_i are the Cartesian coordinates specifying the drone's location in space. For a swarm of n drones, the positions can be encapsulated within a matrix P where each row corresponds to the position vector of a drone:

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix} \tag{4.5}$$

4.4.1 Command Vector and Position Update

When a command is issued to the swarm, it is represented by a vector c that describes the change to be applied to each drone's position:

$$c_i = [c_x, c_y, c_z] \tag{4.6}$$

where c_x, c_y and c_z represent the changes along the x, y, and z dimensions, respectively.
Upon receiving a command, the new position matrix $P^{'}$ of the drone swarm is calculated by adding the command vector to each row (i.e., each drone's position vector) of the matrix P :

$$\mathrm{P}' = \mathrm{P} + 1_n \otimes \mathrm{c}$$

where 1_n is an $n \times 1$ column vector of ones, and $\otimes$ denotes the outer product, effectively broadcasting the command vector across all drones.
For instance, consider an initial configuration of three drones positioned as:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.7}$$

and a command to move "Up" represented by the vector c = [0,0,-1], the new positions would be:

$$P' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & -1 \\ 0 & 0 & -1 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} \tag{4.8}$$

This mathematical framework facilitates a systematic and precise method for controlling each drone's position in a swarm, allowing for coherent and synchronized movements based on user commands, thereby enhancing the operational effectiveness and responsiveness of the swarm to external inputs.

4.4.2 Mathematical Formulation of the RTL Command

When the RTL command is activated, the position matrix P is set to revert to P_{init}. This operation can be mathematically described as:

$$P_{RTL} = P_{init} \tag{4.9}$$

where P_{RTL} is the position matrix post-execution of the RTL command. This simple yet effective command ensures that all drones within the swarm navigate back to their initial start points, irrespective of their current positions in the space.

In summary, this section has presented the results from three distinct experimental setups designed to assess different UAV control methods: direct interaction through a virtual keyboard via gaze detection, hand gesture recognition from datasets, and eye movement recognition from datasets. Each control method was evaluated under controlled conditions to ensure the reliability of the findings.

5

Discussion

This section of the report delves into the findings derived from the deployment of three distinct interaction techniques used for controlling drone swarms: virtual keyboard via eye interaction, hand gesture recognition, and eye movement datasets. Each method was critically analyzed to identify its effectiveness, practical challenges, and potential areas for improvement. The insights gathered not only shed light on the current capabilities of these interaction systems but also guide future enhancements to optimize their efficiency and applicability in diverse operational contexts.

5.1 Findings, Limitations and Future Work

The virtual keyboard system facilitated by eye interaction revealed a high level of precision in command execution when the timing of the user's blinks aligned perfectly with the illumination of the desired command keys. However, the system's dependency on exact timing posed challenges in user experience, indicating a need for more forgiving input recognition mechanisms.

The hand gesture recognition system demonstrated a notable potential for intuitive control but was hampered by inaccuracies due to an unbalanced training dataset and the absence of a neutral or rest state. These issues underscore the importance of enhancing dataset diversity and integrating system pauses to improve recognition accuracy and user comfort.

The eye movement-based control system exhibited biases in command recognition, primarily due to uneven data representation. This insight points to the necessity for a more balanced approach to training dataset compilation. Additionally, the prospective development of an eye movement-driven control system for extended reality applications highlights innovative directions for future research.

5.1.1 Findings on Eye Interaction Using a Virtual Keyboard

The study examined the effectiveness of using eye interaction through a virtual keyboard for drone command and control. The system operates by highlighting each key sequentially, with a light indicator that remains on each key for one second. Users issue commands by blinking when the desired key is illuminated. This method ensures that commands are transmitted accurately if the user's blink coincides with the activation of the key light. This synchronization between user input (blinking) and system response (key illumination) is critical for the correct execution of commands as presented in the figure.4.5 or figure.4.6.
However, the timing mechanism presents challenges. If the user fails to blink at the precise moment a key is illuminated, the system cycles through the entire set of keys before returning to the missed key. This delay can impede the responsiveness of the system, potentially affecting the operational efficiency of the drone swarm. Although the system successfully sends accurate, real-time data to the drone swarm upon receiving a correct command, the dependence on precise timing for blinking can complicate user interaction.
From the perspective of user-friendliness, there is a significant opportunity for improvement. The current interface requires users to wait for the key to reappear in the cycle, which can lead to inefficiencies, particularly in scenarios where timely command execution is crucial. Enhancing the interface to allow for more immediate reselection of missed keys or implementing a more forgiving input recognition might reduce the chance of missed commands and improve the overall user experience.
These findings highlight the need for further refinement of the system to enhance its accessibility and usability, ensuring that drone control via eye interaction is both effective and user-friendly.

5.1.2 Findings on Using Hand Gesture Datasets for Drone Swarm Control

The utilization of hand gesture recognition datasets for controlling drone swarms presents several challenges and areas for improvement as identified during the study. The system's current configuration and training methodologies exhibit limitations that affect both the accuracy and functionality of gesture-based drone command input.

- **Inaccuracy in Gesture Prediction:** One of the critical issues observed is the system's occasional failure to accurately predict gestures. Notably, gestures intended as "Left" are sometimes misrecognized as "Right as shown in the figure. 5.1." This misrecognition stems from inadequacies within the training dataset, which may not sufficiently represent the variety or subtlety of natural human gestures. Enhancing the diversity and quality of the dataset is essential to improve the accuracy of gesture recognition.

Figure 5.1: The figure represents the misrecognized data, User shows "back" as a command, however, the system recognized "left" as command

- **Lack of Garbage Data Training:** The current system has not been trained with "garbage" data — non-gestural, irrelevant motions that should not trigger any drone commands. This absence leads to the system's inability to dismiss non-command gestures, often interpreting them as valid commands. Integrating a dataset that includes such non-intentional or miscellaneous gestures could help the system distinguish between deliberate commands and random or irrelevant movements.

- **Continuous Prediction Without Rest:** The gesture recognition system is configured to continuously predict labels, such as up, down, left, right, front, back, land, and hover, even when no deliberate gesture is made by the user. This persistent data processing and command sending result in unnecessary system strain and potential misoperations, as it doesn't allow for a neutral or idle state.

- **Non-Stop Data Transmission Issues:** Due to the system's continuous prediction approach, it transmits data incessantly without breaks, which can lead to inefficiencies and errors in command execution. This relentless data flow makes it difficult to maintain systematic control, especially in a multi-drone management context where precision and synchronization are crucial.

- **Systematic Management Challenges:** The persistent issues with gesture recognition accuracy and non-stop command output pose significant challenges in maintaining a systematic and reliable control environment for a swarm of drones. The full setup, therefore, requires substantial improvements to ensure it can effectively manage the nuanced demands of drone swarm operations.

These findings underscore the necessity for comprehensive improvements in the hand gesture recognition system used for drone control. Future work should focus on enhancing the training dataset, incorporating garbage data training, and refining the system's ability to enter a neutral state when no gestures are detected. Moreover, developing algorithms that can more effectively manage periods of inactivity and reduce erroneous data transmission will be crucial in achieving a more reliable and efficient control system for drone swarms. Such advancements will not only improve the accuracy and responsiveness of the gesture-based control system but also enhance its applicability in complex operational scenarios.

5.1.3 Findings on Using Eye Movement Datasets for Drone Swarm Control

The investigation into the use of eye movement datasets for controlling drone swarms has revealed several critical insights and areas for potential enhancement. These findings emphasize the impact of data training biases and the need for improved system design to enhance the functionality and applicability of eye movement-based control systems.

- **Bias in Gesture Recognition:** A significant issue detected in the current setup is the biased recognition of eye movements, where the system disproportionately identifies the "Left" command. This bias arises because the dataset used for training contains more instances of left-looking eye movements than in other directions as shown in the figure. 4.13. As a result, even when looking forward or down, the system erroneously indicates that the user is looking left. This skew in data leads to inaccuracies in command execution and can severely limit the system's effectiveness in real-world applications.

- **Continuous Data Collection without Breaks:** Similar to the hand gesture system, the eye movement-based control system continuously processes data without any breaks. This non-stop data collection can lead to user fatigue and system overload, diminishing efficiency and increasing the potential for errors. Implementing the break system or idle state recognition could significantly improve user experience and system reliability by providing necessary pauses during non-command phases.

- **Future System Enhancements for Extended Reality:** Looking ahead, there is a compelling opportunity to expand the use of eye movement data in the realm of extended reality (XR), particularly for swarm drone control. The envisioned system would utilize sophisticated eye-tracking technology to direct a swarm of drones, essentially allowing the user's gaze to guide the movement and actions of multiple drones simultaneously. Such an approach could revolutionize interaction paradigms in various applications, from immersive entertainment to complex surveillance operations.

These findings highlight the need for careful consideration of data diversity during the training phase and the importance of system downtime to prevent continuous operation fatigue. Future development efforts should focus on addressing these shortcomings by balancing the training dataset and integrating functionality that can intelligently recognize and respond to user intent based on eye movement patterns. Moreover, exploring the integration of eye movement controls in extended reality environments presents an exciting frontier for research and application, potentially setting the stage for groundbreaking advancements in how to interact with and manage technology using just gaze. The study of natural swarming behaviours—observed in entities ranging from bird flocks to insect swarms—provides invaluable insights into the development of advanced control algorithms for UAV swarms [43]. By analyzing the orderly and rapid obstacle avoidance behaviour of starling flocks, the study adapted the motion patterns—collective, evasion, and local following into a flocking control algorithm tailored for large-scale UAV operations in dynamic and unknown 3D environments [44]. This bio-inspired approach not only enhances the UAVs' ability to navigate safely and efficiently but also illuminates the broader applicational potential of such algorithms in achieving high-density, coordinated movement without centralized control. Simultaneously, the concept of marginal opacity in natural flocks, which allows for long-range information exchange and optimal density for visibility, offers a theoretical framework that could further refine UAV swarm dynamics by improving the global interaction among drones, ensuring faster and more reliable collective responses to environmental changes. This integration of biological principles into UAV technology highlights a promising direction for future research, emphasizing the synergy between natural world observations and robotic application advancements.

5.2 Research Achievements

- **Feasible Alternative Methodologies for Controlling UAVs:** Alternative methodologies for controlling UAVs that do not depend on traditional remote or sensor-based systems include the use of virtual keyboards, hand gesture recognition, and eye movement tracking. These methods leverage computer vision and machine learning to interpret human gestures or gaze as commands, providing a more intuitive and accessible way of interacting with drones. Virtual keyboards allow for precise command input through gaze detection, while gesture and eye movement systems enable users to control drones through natural actions, reducing the reliance on physical controllers and making drone technology accessible to users with physical disabilities.

- **Integration of Computer Vision Technologies in Drone Control Systems:** Computer vision technologies can be integrated into drone control systems to enhance their operational capabilities by providing advanced gesture and object recognition capabilities. For instance, using high-resolution cameras and real-time processing algorithms, drones can be programmed to recognize and interpret specific hand gestures or eye movements as commands. This integration allows for the development of 'smart' drones capable of understanding contextual commands and responding to user intentions more effectively. Additionally, computer vision can be employed to enable drones to navigate complex environments autonomously by identifying obstacles, assessing terrains, and adjusting flight paths accordingly.

- **Effective Methods for Improving Human-Drone Interaction:** The most effective methods for improving human-drone interaction include eye-tracking systems and hand gesture recognition technologies. These methods are considered effective because they allow for natural, intuitive communication between the user and the drone, miming human-human interaction paradigms. Eye-tracking systems offer precision and speed in interpreting user commands by directly correlating gaze points with control inputs, making them highly responsive. Hand gesture recognition allows users to command drones through simple movements, which can be easily learned and remembered, enhancing the user experience and reducing the cognitive load during operation.

- **Computational Techniques to Optimize Command and Control of Drone Swarms:** To optimize the command and control sequences of drone swarms and enhance their operational efficiency and accessibility, several computational techniques should be applied. Machine learning algorithms can be utilized to predict and adapt to changes in the environment or user behaviour, improving the responsiveness and accuracy of the swarm. Reinforcement learning, in particular, can be employed to dynamically adjust control strategies based on real-time feedback. Furthermore, algorithms for decentralized decision-making can enable drones to operate more autonomously, reducing latency in command execution and enhancing system reliability. For users with physical disabilities, adaptive interfaces that customize input methods based on the user's capabilities can be developed using predictive modelling and user profiling techniques.

The exploration of these three control methodologies has illuminated several critical factors essential for advancing drone swarm interaction technologies. Addressing the identified limitations through strategic dataset enhancement, system design revisions, and the integration of advanced computational algorithms will be pivotal in refining these systems. Furthermore, the potential expansion of eye movement control into extended reality scenarios opens new avenues for research and application, promising to transform user interaction paradigms across various technologies. As advance, it is crucial to continue this exploration to fully harness the capabilities of eye and gesture-based control systems, ensuring they meet the evolving demands of drone technology applications effectively.

6

Conclusion

The exploration and development of innovative control methodologies for unmanned aerial vehicles (UAVs), as discussed in this report, significantly contribute to several burgeoning fields, including cognitive science, artificial intelligence (AI), autonomous drone technology, and smart city infrastructure. These advancements offer profound implications for how drones can be integrated and utilized in complex, dynamic environments.

The integration of eye-tracking and gesture-recognition systems into drone controls aligns closely with research in cognitive science, which studies how humans interact with machines. By adapting drone operations to natural human behaviours and responses, such systems can reduce cognitive load and enhance the intuitiveness of user interfaces. This harmony between user and machine not only improves the efficiency of drone operations but also fosters a deeper understanding of human-machine interaction dynamics, which is a central theme in cognitive science. Moreover, the application of AI-based controllers that can learn and adapt to user preferences and environmental conditions represents a leap forward in creating more intelligent, responsive control systems. These controllers exemplify the application of AI principles in real-world scenarios, pushing the boundaries of what autonomous systems can achieve.

The methodologies developed also have significant implications for the advancement of self-flying drones. By employing sophisticated machine learning algorithms and computer vision techniques, drones can become fully autonomous, and capable of complex decision-making and navigation without human intervention. This autonomy is crucial for applications such as aerial surveillance, disaster management, and delivery services, where the ability to operate independently in unpredictable environments is invaluable. The research into more intuitive control systems indirectly enhances the capabilities of these drones, making them safer and more effective.

Finally, the integration of advanced drone control technologies is pivotal for the development of smart cities. Drones equipped with AI-enhanced controllers can be used for traffic management, public safety monitoring, infrastructure maintenance, and more, contributing to more efficient city management and better public services. The ability of drones to interact seamlessly with human operators and autonomously execute complex tasks makes them an essential component of the smart city ecosystem. As cities continue to grow and become more technologically integrated, drones will play a crucial role in ensuring these urban environments are not only intelligent but also adaptable and sustainable.

Moreover, the implications of this improved control system extend beyond daily convenience and accessibility. In scenarios such as pandemics, natural disasters, or other emergencies, drones controlled via intuitive gestures could play a critical role in delivering medical supplies, conducting search and rescue missions, or facilitating communication when traditional infrastructure is compromised. For isolated or sick individuals, drones could provide not only a means of receiving essential supplies without physical contact but also a way to maintain social interactions, thus reducing the psychological impact of isolation.

The concept of "Drones for All" encapsulates this vision, where the goal is to democratize the use of drone technology, making it a universally accessible tool that requires minimal training. This would significantly benefit the elderly, disabled, and those in remote or underserved areas, ensuring that everyone can leverage the potential of UAVs to enhance their quality of life and safety. These intuitive control methods, designed to enhance accessibility for individuals with physical disabilities, also offer significant usability benefits for able-bodied users, simplifying interaction and reducing the learning curve for all.

The advancements in drone control technologies discussed in this report not only push the envelope in terms of what drones are capable of but also contribute broadly to fields like cognitive science, AI, and urban planning. These technologies provide the tools necessary to realize the potential of UAVs in a variety of applications, from enhancing public safety in smart cities to advancing autonomous flight. As such, the ongoing research and development in this area are not only about improving drone technology but also about driving forward the future of how technology interacts with and shapes our world.

Bibliography

[1] A. T. Azar, A. Koubaa, N. Ali Mohamed, *et al.*, "Drone deep reinforcement learning: A review," *Electronics*, vol. 10, no. 9, p. 999, 2021.

[2] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of unmanned aerial vehicles*. Springer, 2015, vol. 1.

[3] Y. Zeng and R. Zhang, "Energy-efficient uav communication with trajectory optimization," *IEEE Transactions on wireless communications*, vol. 16, no. 6, pp. 3747–3760, 2017.

[4] S.-H. Joo, S. Manzoor, Y. G. Rocha, *et al.*, "Autonomous navigation framework for intelligent robots based on a semantic environment modeling," *Applied Sciences*, vol. 10, no. 9, p. 3219, 2020.

[5] B. Liu, X. Xiao, and P. Stone, "A lifelong learning approach to mobile robot navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1090–1096, 2021.

[6] Y. Mulgaonkar, A. Makineni, L. Guerrero-Bonilla, and V. Kumar, "Robust aerial robot swarms without collision avoidance," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 596–603, 2017.

[7] M. T. Yang, "Avian rhetoric, murmurations," Ph.D. dissertation, University of Pittsburgh, 2019.

[8] N. Ivanova, "Swarm robotics-coordination and cooperation: Exploring coordination and cooperation strategies in swarm robotics systems for achieving collective tasks," *Journal of Computational Intelligence and Robotics*, vol. 4, no. 1, pp. 1–13, 2024.

[9] I. Navarro and F. Matía, "An introduction to swarm robotics," *Isrn robotics*, vol. 2013, pp. 1–10, 2013.

[10] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[11] A. Khang, V. Abdullayev, O. Hrybiuk, and A. K. Shukla, *Computer Vision and AI-Integrated IoT Technologies in the Medical Ecosystem*. CRC Press, 2024.

[12] A. Mahboob, "You need to start learning," in *Writings on Subaltern Practice*, Springer, 2023, pp. 135–135.

[13] A. F. Villán, *Mastering OpenCV 4 with Python: a practical guide covering topics from image processing, augmented reality to deep learning with OpenCV 4 and Python 3.7*. Packt Publishing Ltd, 2019.

[14] D. Kavitha, B. R. Kiran, B. Niteesh, and S. Praveen, "Multiple object recognition using opencv," *REVISTA GEINTEC-GESTAO INOVACAO E TECNOLOGIAS*, vol. 11, no. 2, pp. 1736–1747, 2021.

[15] K. Krafka, A. Khosla, P. Kellnhofer, *et al.*, "Eye tracking for everyone," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2176–2184.

[16] B. Farnsworth. "What is eye tracking and how does it work?" Accessed: 2024-05-11. (2023), [Online]. Available: `https://imotions.com/blog/learning/best-practice/eye-tracking-work/`.

[17] *Sign language*, `https://makeitfable.com/glossary-term/sign-language/`, Accessed: 2023-05-11.

[18] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[19] E. Graells Pina, "Design of an uav swarm," M.S. thesis, Universitat Politècnica de Catalunya, 2022.

[20] H. Hamann, "Introduction to swarm robotics," in *Swarm robotics: A formal approach*, Springer, 2018, pp. 1–32.

[21] H. Ahn and S. Chung, "Deep learning-based anomaly detection for individual drone vehicles performing swarm missions," *Expert Systems with Applications*, vol. 244, p. 122 869, 2024, ISSN: 0957-4174. DOI: `https://doi.org/10.1016/j.eswa.2023.122869`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0957417423033717`.

[22] S. Chen, K. Demachi, and M. Tsunokai, "A real-time automated approach for ensuring proper use of personal protective equipment (ppe) in construction site," in *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering: ICCCBE 2020*, Springer, 2021, pp. 1115–1126.

[23] Tech Wire Asia, *Honor magic 6 pro features eye-tracking ai to control cars*, Accessed: 2024-03-27, 2024. [Online]. Available: `https://techwireasia.com/02/2024/honor-magic-6-pro-features-eye-tracking-ai-to-control-cars/#:~:text=Honor%2C%20a%20prominent%20Chinese%20tech,gazing%20at%20their%20phone%20screens.`.

[24] I. Alam, A. Hameed, R. A. Ziar, *et al.*, "Exploring sign language detection on smartphones: A systematic review of machine and deep learning approaches," *Advances in Human-Computer Interaction*, vol. 2024, 2024.

[25] R. Anderson, F. Wiryana, M. C. Ariesta, G. P. Kusuma, *et al.*, "Sign language recognition application systems for deaf-mute people: A review based on input-process-output," *Procedia computer science*, vol. 116, pp. 441–448, 2017.

[26] K. Sohelrana, S. F. Ahmed, S. Sameer, and O. Ashok, "A review on smart gloves to convert sign to speech for mute community," in *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, IEEE, 2020, pp. 1262–1264.

[27] J. Galván-Ruiz, C. M. Travieso-González, A. Tejera-Fettmilch, A. Pinan-Roescher, L. Esteban-Hernández, and L. Domínguez-Quintana, "Perspective and evolution of gesture recognition for sign language: A review," *Sensors*, vol. 20, no. 12, p. 3571, 2020.

[28] S. Sharma and S. Singh, "Vision-based sign language recognition system: A comprehensive review," in *2020 international conference on inventive computation technologies (ICICT)*, IEEE, 2020, pp. 140–144.

[29] W. Zhong, H. Xu, Y. Ke, *et al.*, "Accurate and efficient sitting posture recognition and human-machine interaction device based on fabric pressure sensor array

and neural network," *Advanced Materials Technologies*, vol. 9, no. 3, p. 2 301 579, 2024.

[30] G. Pan, L. Sun, Z. Wu, and S. Lao, "Eyeblink-based anti-spoofing in face recognition from a generic webcamera," in *2007 IEEE 11th international conference on computer vision*, IEEE, 2007, pp. 1–8.

[31] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Ieee, vol. 1, 2001, pp. I–I.

[32] T. Sravya, S. N. Bhargava, S. Shravani, R. Bodke, and N. Kulkarni, "Ai virtual hardware," in *2022 IEEE Pune Section International Conference (PuneCon)*, IEEE, 2022, pp. 1–6.

[33] L. Anjaneya, U. Mahanthesha, and K. Banumathi, "Machine learning approach to the classification and identification of hand gesture recognition using python," 2023.

[34] M. Kim, O. Wang, and N. Ng, "Convolutional neural network architectures for gaze estimation on mobile devices," *Standford Univ., Standford, CA, USA, Tech. Rep*, 2016.

[35] T. Lee, V. P. Singh, K. H. Cho, T. Lee, V. P. Singh, and K. H. Cho, "Tensorflow and keras programming for deep learning," *Deep Learning for Hydrometeorology and Environmental Science*, pp. 151–162, 2021.

[36] S. ENGLE and S. WHALEN, "Dypy: Dynamical systems in python,"

[37] P. Virtanen, R. Gommers, T. E. Oliphant, *et al.*, "Scipy/scipy: Scipy 0.19. 0," *Zenodo*, 2020.

[38] M. Xu, D. Chen, and G. Zhou, "Real-time face recognition based on dlib," in *Innovative Computing: IC 2020*, Springer, 2020, pp. 1451–1459.

[39] A. Yim, C. Chung, and A. Yu, *Matplotlib for Python Developers: Effective techniques for data visualization with Python*. Packt Publishing Ltd, 2018.

[40] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 20–32, 2012.

[41] P. Pounds, R. Mahony, and P. Corke, "Modelling and control of a quad-rotor robot," in *Proceedings of the 2006 Australasian Conference on Robotics and Automation*, Australian Robotics and Automation Association (ARAA), 2006, pp. 1–10.

[42] T. S. Andersen and R. Kristiansen, "Reactive-based position control of an under-actuated quadrotor," in *2020 European Control Conference (ECC)*, IEEE, 2020, pp. 1656–1661.

[43] D. J. G. Pearce, A. M. Miller, G. Rowlands, and M. S. Turner, "Role of projection in the control of bird flocks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 29, pp. 10 422–10 426, 2014. DOI: 10.1073/pnas.1402202111. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.1402202111. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1402202111.

[44] J. Brookshaw, J. Francioni, M. Zhang, and N. Debnath, "Warriordrone api development: Ar. drone 2.0 node. js api supporting color detection," in *The 15th Winona Computer Science Undergraduate Research Symposium*, 2015, p. 28.

Appendix

proposed project: Drone swarm communication and coordination
Abstract: The main area of interest is to control a swarm of small drones (airbits and/or plutoX drones). What AI technique to use, (agent framework, negotiations, blackboard etc.). How to manage communication and coordination in the swarm to solve a specific task. Tasks may include, but are not limited to, land surveys, disaster area mapping, and resource transportation. The students may suggest additional tasks as they see fit. How to manage range, battery life and other constraints. The student may work with real drones (6X airbits and 2X plutoX) or develop a simple simulator to demonstrate the concept.
Desired result: A swarm of drones that can cooperate and coordinate their efforts to solve a given task.
Data requirement and availability:
Tentative co-supervisor:
Andreas DJ