Modifly: A Multimodal Interaction Simulator for Drone Operation via Gesture Recognition and Voice Commands

PASQUALE CELANI, Sapienza, Italy

Drones are increasingly prevalent in various fields, but their control systems often rely on traditional, joystick-based interfaces, which can suffer from high production costs, physical fragility, and a lack of intuitive interaction. To address these limitations, the ModiFly project introduces a multimodal human-drone interaction (HDI) system that leverages natural communication modalities such as gestures and voice commands. This work details the design and implementation of a simulator built on the Godot game engine, which provides a safe and cost-effective platform to test this new paradigm. Within this framework, gestures are used for continuous drone movement, while voice commands are reserved for discrete, supportive functions. The system employs a decision-level fusion strategy for gesture recognition, combining an integrated MediaPipe classifier with a landmark-based approach for enhanced reliability. Speech recognition is handled by an external API to minimize computational load on the system. The ModiFly project's approach aims to provide a more natural and asynchronous user experience, which preliminary empirical test it indicate a promising direction for future HDI research on this topic.

CCS Concepts: • **Human-centered computing → Mixed / augmented reality**; **User centered design**; **Interface design prototyping**; **Systems and tools for interaction design**; **Accessibility technologies**.

Additional Key Words and Phrases: Drone Simulation, Gesture Recognition, Voice Control, Multimodal Interface, Multimodal Interaction,Natural User Interface, Virtual Environment

1 Introduction

Drones have become increasingly prevalent across a multitude of human endeavors. Beyond their growing presence in cinematic production and entertainment, drones are playing a particularly significant and evolving role in contemporary military conflicts worldwide. A prominent example of their impact can be observed in the ongoing Russian-Ukrainian conflict, where their deployment has redefined battlefield dynamics and tactics. While the recent prominence of drones in conflicts like the Russian-Ukrainian war might suggest a novel development their integration into military operations is not a new phenomenon. As documented by [Gilli and Gilli 2016], drones were first employed in warfare as early as the 1990s. However, their role has expanded dramatically in recent years, largely driven by significant advancements in Information and Communication Technologies (ICT). This technological progress, coupled with the decreasing cost of production due to the utilization of common, commercially available components, has made drones increasingly accessible and ubiquitous on the modern battlefield.

Current drone control systems often rely on human machine interfaces that mimic those found in consumer electronics. The most common of these are joysticks, similar in design and function to those used with popular video game consoles such as the PlayStation or Xbox. This method of interaction provides a direct link between the user and the drone, enabling rapid and precise manipulation, particularly for fine grained movements. However, this "traditional" control paradigm, while effective, presents several significant limitations that must be addressed for future applications:

- Higher production costs;
- Physical Fragility;
- Risk of Loss;
- Unnatural Interaction.

Although the low cost of some consumer grade joysticks may appear to mitigate the issue of production expense, this is a misconception when considering large-scale manufacturing. Even minor increases in component costs

Author's Contact Information: Pasquale Celani, Sapienza, Rome, Lazio, Italy, celani.1839634@studenti.uniroma1.it.

when multiplied across a massive production run can lead to significant financial losses. Furthermore, these inexpensive joysticks are often highly susceptible to damage and wear making them unreliable in the field. More durable military-grade controllers, while built to last, are prohibitively expensive and still remain vulnerable to physical damage or degradation over time. Beyond durability, the use of external control devices introduces several points of failure and vulnerability. The physical separation of the controller from the user means that it can easily be lost or stolen, effectively disabling the drone.

Most importantly, the abstract nature of joystick commands presents a significant cognitive barrier to intuitive operation. For example, assigning the "X" button to a forward movement creates an arbitrary link between action and command. This lack of a natural, or embodied, connection requires a user to learn and memorize a specific control scheme, which can be difficult and slow, especially in high-stress situations. This stands in stark contrast to more natural human-computer interfaces that leverage gestures or speech, which are inherently more intuitive and require less cognitive load.

This project aims to introduce a more intuitive and natural human drone interaction model by moving beyond abstract button-based interfaces. The aim is to take advantage of natural human communication modalities, specifically gestures and speech, to create a more seamless and direct control experience. To facilitate the development and evaluation of this new paradigm, a simulated environment has been developed for drone control. This simulator provides a safe and cost-effective platform to test and refine our natural interaction modalities before transitioning to real-world drone applications, an approach that is both practical and essential to ensure system safety and reliability.

This document is structured to provide a comprehensive overview of my research on natural human-drone interaction. **section 2** presents a review of related work, examining existing research on drone control paradigms that utilize post-WIMP interaction methods. Following this, **section 3** details the design of the multimodal interaction framework discussing the integration of gestures and speech, as well as the methodology for combining these modalities. **section 4** provides a technical overview of the simulator's architecture and key implementation details. Finally, **section 5** offers concluding remarks, summarizing the key contributions of the project, and outlining a roadmap for future development.

2 Related Work

Human-drone interaction (HDI) is a central and increasingly active area within the field of multimodal interaction. A significant body of research is dedicated to exploring novel control paradigms that move beyond traditional remote control systems. For instance, the work by [Latif et al. 2022] proposes an approach to drone control using computer vision and deep learning. Their implementation leverages MediaPipe to detect and track hand gestures in real-time via a standard RGB camera. A pre-trained classifier is then used to recognize specific gestures, translating them into drone commands.

Beyond the work of [Latif et al. 2022] on hand-gesture-based drone control, another similar work is that of [Yun et al. 2024]. This research focuses on a vision-based Human-Gesture Recognition (HGR) approach for complex drone maneuvers, addressing the need for a large gesture vocabulary to control the four fundamental movements (roll, yaw, pitch, and throttle) and their three behaviors, ultimately suggesting a gesture set capable of expressing 81 combinations. Their goal was to create a lightweight and efficient HGR algorithm suitable for real-time operation on edge devices. Technically, their method briefly uses MediaPipe to extract landmark and joint position data from the hand, which is then passed into a Multi-Layer Perceptron (MLP), a feedforward neural network, to classify the gesture posture. Additionally, a notable aspect of this research, which aligns with the ModiFly project, is the use of a simulated environment for testing. The authors integrated their gesture recognition system into a virtual drone simulator to evaluate the effectiveness of their gesture vocabulary.

Another interesting research work on HGR is [Khaksar et al. 2023]. This work highlights a key weakness of MediaPipe Hands (MPH) in the control algorithm and demonstrated using a quad-rotor drone control: the Z-axis instability of its modeling system, which reduced the precision of the landmark from 86. 7% to 41. 5%. This work was particularly interesting for the ModiFly project as it provided confirmation of an observation made during its early stages. In the initial project stage gestures postures to control a drone were evaluated, and I arrived at a similar conclusion through empirical means: that MediaPipe landmarks were not consistently reliable for capturing 3D depth, likely due to their 2.5D nature. The findings of this paper, highlighting the Z-axis instability of the MPH system, scientifically validate this initial observation and provide a robust foundation for understanding the limitations of using such a framework for precise 3D drone control.

Beyond conventional vision-based HGR approaches, an interesting avenue in HDI research involves the use of multimodal tangible devices. [Yau et al. 2020], explore a novel one-handed, on-finger hardware interface for drone control. This device utilizes three modalities, touch, force, and an inertial measurement unit (IMU), to allow a user to control a drone with subtle thumb to index finger interactions. Panning gestures on a touch-sensitive surface are used for forward, backward, and lateral movements, while applying force to this surface activates the IMU, allowing wrist movements to control vertical and rotational movements. This device demonstrated significant performance improvements over traditional two-handed joysticks. In an experiment with 12 participants, the selected method achieved a task completion time of 16. 54% faster than the two-handed approaches. The research underscores the value of multimodal input and subtle and tangible interfaces as a promising alternative to purely vision-based systems.

The work of [Yau et al. 2020] provides an interesting counterpoint to purely vision based HGR systems by demonstrating the effectiveness of a multimodal tangible interface. However, this approach still relying on physical interaction similar to a traditional joystick to create a more efficient and compact control system with all it's related disadvantages. Instead, a different approach which aligning more closely with the multimodal strategy of the ModiFly project, is explored in the research by [Xiang et al. 2022]. This work investigates the fusion of voice and gesture modalities. The authors demonstrate that this strategy offers several advantages for HDI including:

- Interacting with immersive displays and remotely controlling UAV missions is no longer convenient;
- Gestures in combination with speech in virtual environments are easier to learn for the operators;
- Gesture-based natural interaction systems are the most intuitive system type;
- Speech-based natural interaction systems provide better system control;
- When gestures and speech are present simultaneously, it helps reduce faster task completion times and even lower error rates.

In detail, the authors describe a multimodal system that combines gesture and voice modalities for drone control. The gesture recognition is accomplished using a Leap Motion system, while an offline command word recognition SDK from the iFlytek open platform handles the voice recognition. The core of their approach is a score-level fusion strategy. This method normalizes the confidence scores from both the gesture and voice classifiers for each potential command. The final command is then selected based on the highest summed confidence score. The system was evaluated in a virtual environment. The experimental setup utilized a VR headset, Steam VR, and the Gazebo simulator within the Robot Operating System (ROS) framework. The efficacy of the system was tested by having a drone navigate a city environment to reach a designated objective, providing a practical validation of the proposed multimodal control scheme.

Beyond the academic sphere a compelling use case for HDI systems can be found in the gaming industry. As highlighted in the **section 1**, the prominent role of drones in recent global conflicts has significantly increased their visibility, which in turn has spurred interest in other sectors including gaming. Titles like *"Death From Above: A Ukrainian Drone Warstory"* exemplify this trend. While this is just one example, the existence of such games suggests that the gaming industry could serve as a valuable and dynamic platform for further research into HDI. These commercially developed simulators and games offer pre-existing, immersive virtual environments and a variety of control schemes. Researchers could leverage these platforms as a template to develop and test new HDI interfaces, providing a practical and engaging context for evaluating the usability, intuitiveness, and effectiveness of different gesture- and voice-based control systems.

3 Interaction Design

The ModiFly project as outlined in this paper is an exploration of HDI that leverages both voice and gesture modalities. Before detailing the specific command to modality mapping, it is essential to establish a clear definition of the term "modality" to avoid ambiguity. While numerous definitions exist within the multimodal interaction literature, for the purpose of this project, we will adopt the definition provided by [Bernsen and Dybkjær 2009].

Definition: A **modality**, or, more explicitly, a modality of information representation, is a way of representing information in some medium, and since the medium is linked to a particular physical carrier and to a particular kind of sensor system, a modality is defined by its medium-carrier-sensor system triplet.

Based on the provided definition from the reference, the ModiFly project can be understood as a multimodal system that utilizes two primary modalities for HDI. Specifically, the Gesture modality corresponds to the Graphics-Light-Vision modality, as the input is captured by an RGB camera. The Voice modality aligns with the Acoustics-Sound-Hearing modality, with input captured by a microphone.

In the ModiFly project, the choice of modalities was not arbitrary but was strategically selected to promote a more natural HDI, building on insights from the literature reviewed in **section 2**. Within this framework, the two modalities Graphics-Light-Vision and Acoustics-Sound-Hearing are not treated as equals but are assigned distinct, complementary roles. Gestures are designated as the primary control mechanism for continuous drone movements. Conversely, voice commands are relegated to a secondary role, primarily for issuing discrete setup or secondary commands for clarifying ambiguous gestures. This relationship can be formally described using the taxonomy of multimodal interactions outlined by [D'Ulizia 2009]. The system operates on a principle of both specialization and concurrency. Specialization is evident in how certain types of information are handled exclusively by a single modality. For example, a command like "go forward" is processed solely through the gesture modality (Graphics-Light-Vision), while a setup command is handled by the voice modality (Acoustics-Sound-Hearing). The modalities also exhibit concurrency, functioning in parallel without a mandatory merge point.

This interaction model, which leverages both specialization and concurrency, represents a major strength of the ModiFly project's multimodal approach. It enables a more naturalistic and asynchronous user experience compared to traditional unimodal interfaces. A key advantage is the ability to perform complex actions without the need for convoluted button combinations, as is often the case with a traditional joystick. For example, while an operator's hands are engaged in continuous gestural control for drone movement, a voice command like "Zoom in/out" can be issued in parallel to regulate the camera. This division of labor between modalities allows for simultaneous, parallel task execution, thereby reducing cognitive load and enhancing overall operational efficiency.

With the modalities and their respective roles now defined, attention is turned to describing the specific actions mapped to each modality. This section will begin with an in-depth look at the gesture modality (Graphics-Light-Vision). The **Figure 1** presents an overview of the six distinct gestures employed in the ModiFly project. Each

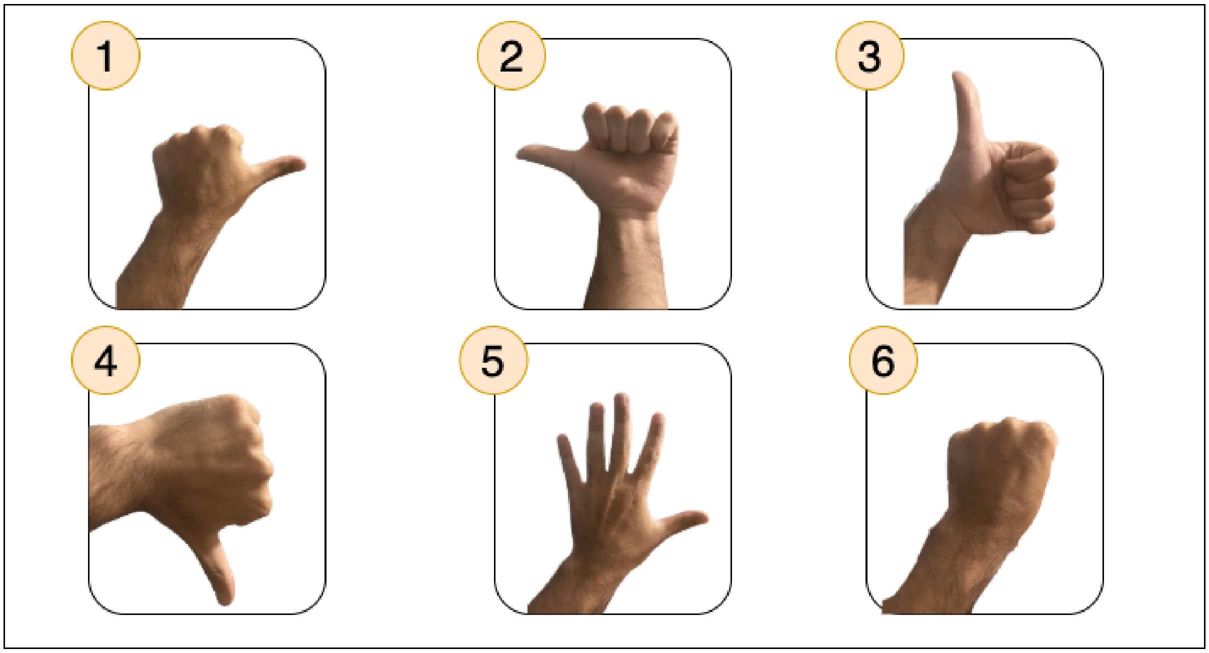

Fig. 1. Overview of the six hand gestures used in the project for the drone control mechanism. Each gesture is numbered for easy reference.

gesture has been assigned a number to facilitate a clear and unambiguous reference throughout the discussion, thereby simplifying comprehension of command-to-action mapping.

Each of the previously defined gestures is mapped to a specific action within the ModiFly control scheme. The interaction is modeled on the conceptual framework of a traditional joystick, where each hand is assigned a distinct operational role. The left hand is responsible for controlling the drone's position, while the right hand manages its rotations. Specifically, for the left hand:

- Gestures 1 and 2 correspond to lateral movement, directing the drone left and right, respectively;
- Gestures 3 and 4 control the drone's altitude, causing it to ascend and descend;
- Gestures 5 and 6 are mapped to forward and backward movement (considering acceleration and deceleration).

Conversely, the right hand uses the same directional gestures (1 and 2) to control the drone's rotational yaw. Gesture 1 turns the drone to the right, while gesture 2 symmetrically turns it to the left.

In the ModiFly simulator, each gesture is interpreted as a discrete command, meaning a specific action is triggered each time a particular gesture is detected. For example, when the gesture 3 is recognized, a precise command is issued to increase the drone's altitude by a fixed increment, such as 0.005 units. A key feature of the simulation, which is present in the majority of modern drones, is the altitude hold mechanism. This autonomous function, allows the drone to maintain a stable altitude without continuous manual input, thereby eliminating the need for a specific command when the drone is idle. This feature significantly simplifies the control scheme and reduces the operator's cognitive load.

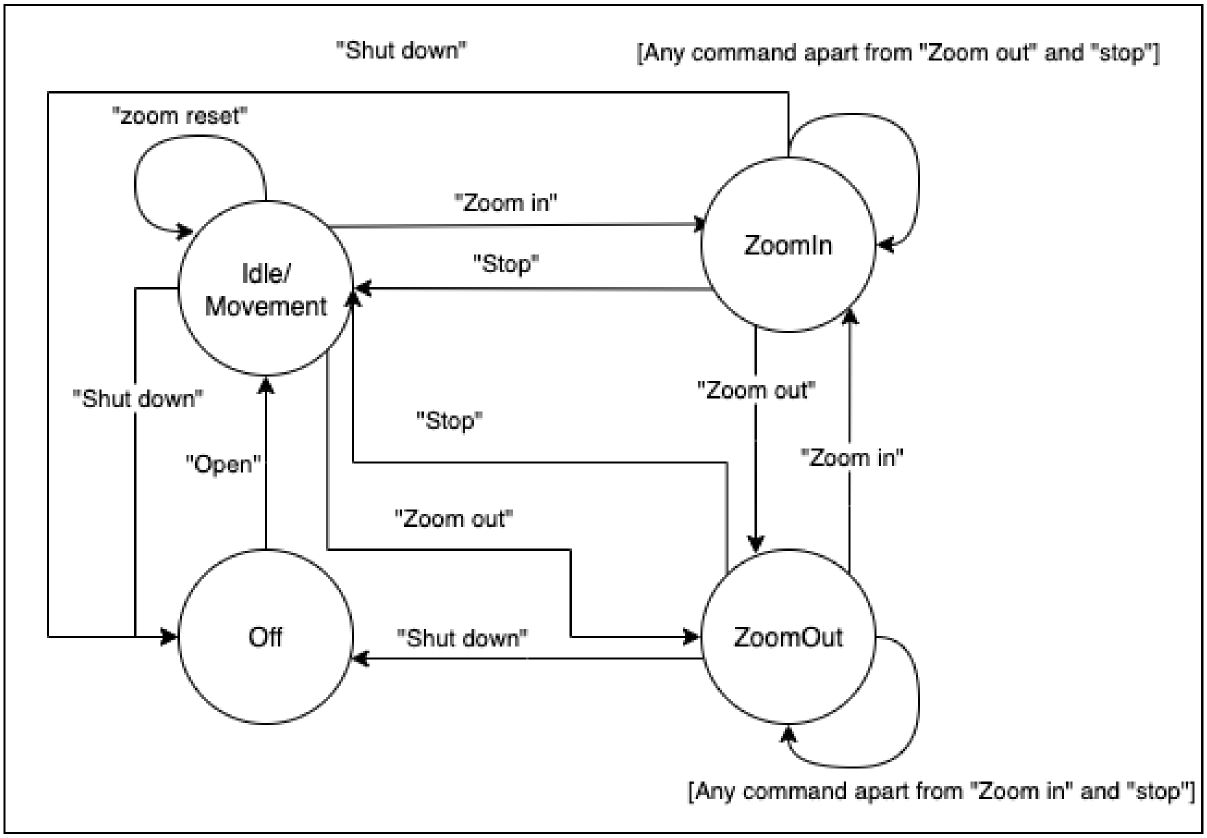

Fig. 2. State diagram illustrating the drone's system states and transitions as controlled by voice commands.

Having examined the use of gestures for continuous drone movement control, we now transition to the voice commands. As previously established, voice commands serve a supportive role in this multimodal framework. Specifically, the following voice commands are utilized in the ModiFly project:

- "open" initiates the drone's systems, enabling it for flight;
- "shut down" deactivates the drone's systems, concluding the flight session;
- "zoom in/out" to zoom in/out the drone camera;
- "stop" to stop the camera zoom in/out.

Each of the aforementioned voice commands modifies a distinct system state within the ModiFly simulator. The related state diagram, shown in **Figure 2** , illustrates how these commands trigger state transitions. The idle/movements and off states represent the drone's operational status, whether it is shut down, idle, or actively in motion. Conversely, the zoom in/out states indicate that the camera is engaged in a specific zoom function. It is important to note that the zoom out state allows for concurrent movement or idling, reflecting the asynchronous nature of the multimodal interaction using gestures. Indeed, this diagram focuses solely on the voice command states and their transitions, not on the drone's simultaneous movement, which is governed by the gestures described earlier.

4 Implementation Details

The technical implementation details of the ModiFly simulator and its multimodal control mechanisms will now be discussed. The simulator was developed using the Godot game engine, which was selected for its open-source nature and developer-friendly features. Its object-oriented design philosophy and robust scripting capabilities, facilitated by GDScript, were instrumental in implementing the required control logic.

A primary challenge in using the Godot engine was its compatibility with traditional deep learning libraries, which are typically implemented in Python rather than GDScript. To address this, a strategy inspired by multiplayer game architecture was employed. A local host server process, equipped with the necessary deep learning libraries, was established to process the recognition tasks. The commands generated by this server were then transmitted via UDP to a client component within the Godot simulator. Upon receipt, this client component would execute the corresponding drone action. The Godot side of the implementation contains no logic for command generation, it simply processes the received commands. To ensure seamless multimodal interaction, each message sent to the client was structured in JSON format, encapsulating both voice and gesture commands in a single, cohesive message.

Having detailed the Godot command communication system, we will now delve into the specific implementations of the gesture and voice control systems. Beginning with the gesture system, static hand postures are the primary form of input. These gestures are recognized using the MediaPipe library, which is employed in two distinct ways:

- **Hand landmark recognition:** MediaPipe library is used to identify key hand landmarks. The spatial relationships between these landmarks are then analyzed to classify the specific gesture being performed;
- **Gesture classifier**: The MediaPipe library also includes a built-in classifier capable of recognizing a set of predefined static gestures, such as "thumb up", "thumb down", "open palm" and etc.

Within the ModiFly project the MediaPipe classifier is utilized to recognize gestures 3, 4, 5 and 6 as depicted in **Figure 1**. However, gestures 1 and 2 are classified by analyzing the landmark points of the hand. The hand landmark recognition, on the other hand, is based on the following formulas:

- **Thumb left:** $lm[4].x < lm[3].x < lm[2].x \ \wedge \ \forall k \in \{5, 9, 13, 17\}\ lm[k+3].y > lm[k].y$;
- **Thumb right:** $lm[4].x > lm[3].x > lm[2].x \ \wedge \ \forall k \in \{5, 9, 13, 17\}\ lm[k+3].y > lm[k].y$.

Where lm is the array of hand landmarks identified by MediaPipe, and lm[i] corresponds to the i-th joint of the hand. This two-pronged approach was necessary due to the limited set of predefined gestures available within the MediaPipe classifier. Both of these approaches are used to recognize gestures, which necessitates a multimodal fusion strategy to handle their outputs. A decision-level fusion strategy was chosen, where both the gesture classifier and the landmark-based approach provide a decision, and one is selected. The specific approach implemented gives priority to the gesture classifier, as it is considered more reliable and accurate. The landmark-based approach is only consulted if the classifier fails to recognize a gesture. This prioritization is based on the superior accuracy of the integrated classifier compared to the landmark-based method. In the **Figure 3**, a summary of the concepts discussed in this paragraph is presented.

Having explored the gesture recognition system, we now turn our attention to the implementation of speech recognition in ModiFly. Speech recognition is handled by the speech recognition library, with its implementation running on a separate thread to prevent it from blocking the main application. This system continuously listens for voice commands in the background. In detail, the system's speech recognition parameters are set with a phrase recognition timeout of 2 seconds and a phrase time limit of 2 words. These parameters were carefully chosen to optimize performance. For instance, the timeout is crucial for ensuring that the system can accurately capture the intended phrases. Through empirical testing, it was determined that these values represent the best trade-off between recognition delay and accuracy, particularly given that all of our voice commands are intentionally short. Instead of local on-device processing, the system utilizes Google's Web Speech API, which requires an

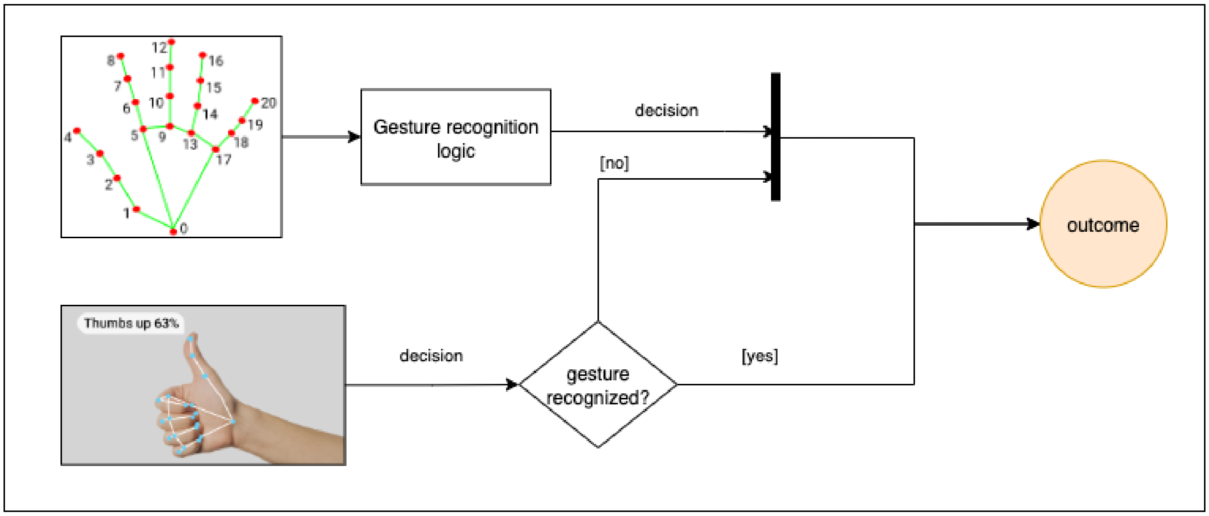

Fig. 3. A summary of the gesture recognition pipeline. The diagram illustrates the decision-level fusion strategy, where the outcomes from both the integrated MediaPipe classifier and the landmark-based classifier are combined, with priority given to the former.

internet connection. This design choice was made to mitigate the computational burden on the system, as the gesture recognition already relies on two deep learning models. Outsourcing the speech processing ensures that the system maintains a high level of performance, characterized by good speed and low latency. The transcribed text is then evaluated against a predefined set of commands such as "open", "shut down", "zoom in", "zoom out" and "stop." The system is also designed with robust error handling, allowing it to gracefully manage situations where speech is not recognized or an API error occurs, and to continue listening without interruption.

5 Conclusion

The ModiFly project successfully developed a simulator to test a novel HDI system that uses gestures as the primary control mechanism, departing from traditional joystick-based interfaces. This approach not only has the potential to reduce production costs but also to create a more natural and intuitive interaction model. A key achievement of this work is the two handed gesture system for handling drone movement, complemented by a multimodal framework that integrates voice commands for supportive functions. The results are encouraging, and an additional accomplishment of the project is its low computational resource requirement, demonstrated by its successful development on a 2019 MacBook Air with a dual-core i5 processor and 8GB of RAM. For future work, it would be valuable to test this system with a physical drone in a real operational context, similar to those discussed in the introduction, and to explore the introduction of more complex voice commands to enable a wider range of interactions. .

References

Niels Ole Bernsen and Laila Dybkjær. 2009. *Multimodal usability*. Springer Science & Business Media.

Arianna D'Ulizia. 2009. Exploring multimodal input fusion strategies. In *Multimodal Human Computer Interaction and Pervasive Services*. IGI Global Scientific Publishing, 34–57.

Andrea Gilli and Mauro Gilli. 2016. The diffusion of drone warfare? Industrial, organizational, and infrastructural constraints. *Security Studies* 25, 1 (2016), 50–84.

Siavash Khaksar, Luke Checker, Bita Borazjan, and Iain Murray. 2023. Design and evaluation of an alternative control for a quad-rotor drone using hand-gesture recognition. *Sensors* 23, 12 (2023), 5462.

Bilawal Latif, Neil Buckley, and Emanuele Lindo Secco. 2022. Hand gesture and human-drone interaction. In *Proceedings of SAI Intelligent Systems Conference*. Springer, 299–308.

Xiaojia Xiang, Qin Tan, Han Zhou, Dengqing Tang, and Jun Lai. 2022. Multimodal fusion of voice and gesture data for UAV control. *Drones* 6, 8 (2022), 201.

Yui-Pan Yau, Lik Hang Lee, Zheng Li, Tristan Braud, Yi-Hsuan Ho, and Pan Hui. 2020. How subtle can it get? a trimodal study of ring-sized interfaces for one-handed drone control. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 2 (2020), 1–29.

Guhnoo Yun, Hwykuen Kwak, and Dong Hwan Kim. 2024. Single-handed gesture recognition with RGB camera for drone motion control. *Applied Sciences* 14, 22 (2024), 10230.