Whistle-Controlled Drone: Real-Time Human-Drone Interaction via Audio & AI

Semester B 2025 – Kfir Messika, Sigal Arabov
Advisor: Mr. Idan Tobis

This project explores a novel way to control a drone using only human whistles, inspired by the fictional control of Yondus arrow in Guardians of the Galaxy. Instead of using a handheld controller, the pilot produces whistle sounds that are processed in real time and translated into drone movement commands.

Project Overview & Motivation

Our primary objective was to engineer a drone control system that operates exclusively through whistling, eliminating the need for any physical controllers. This concept was directly inspired by the unique sound-directed object control depicted in Guardians of the Galaxy, prompting us to investigate its real-world applicability for drone operation.

Intuitive, Hands-Free Control

Developing a seamless and natural interface for drone operation without physical contact.

Accessibility

Enabling individuals with mobility impairments to control drones, expanding user access.

Robotics Research

Contributing to the study of alternative control modalities within the field of robotics.

Human–Computer Interaction (HCI)

Exploring innovative, non-traditional methods for intuitive human-computer interaction.

Main Technical Challenges

The development of a whistle-controlled drone system presented several significant technical hurdles that required innovative solutions.

Control Language

Translating simple whistle tones into complex 3D flight commands (forward, rotation, altitude) with precision and intuition.

Noise Filtering

Ensuring the system reacts exclusively to whistles, effectively ignoring speech, background noise, and other environmental sounds.

Pilot Identification

The drone must respond solely to the intended pilot's whistle, even in environments with multiple individuals whistling.

Control Approaches Explored

We investigated two distinct methods for translating whistle inputs into drone movement commands, ultimately selecting the most effective approach for real-time control.

A. Continuous Mapping (Chosen Method)

In this approach, the drone's movement is controlled continuously based on live audio features extracted from the pilot's whistle.
Instead of predefined commands, the system measures the **pitch**, **volume**, and **pitch change over time** to directly adjust the drone's flight parameters in real time.
This creates an intuitive and fluid control experience, giving the pilot precise and immediate command over speed, altitude, and rotation.

Forward speed	**Pitch** • Lower pitch→ Slower forward movement • Higher pitch → Faster forward movement
Altitude	**Volume** • Softer volume → Drone descends • Louder volume → Drone ascends
Rotation	**Pitch Change** • Low pich to high pich → Rotate right • High pich to low pich → Rotate left

B. Pattern Matching (Not Used)

In this approach, each drone command was assigned a unique whistle pattern — for example, a short melody or a sequence of pitch changes. To recognize these commands, the system first converted the incoming whistle into a numerical representation called an **embedding**. This embedding captured the essential pitch and timing characteristics of the whistle, enabling comparison against stored reference patterns. When the pilot whistled, the system would search for the closest match among the stored embeddings and trigger the corresponding command.

Why We Abandoned It:

- **False Activations:** Many whistle patterns shared overlapping segments or similar pitch shapes. This often caused the system to mistake one pattern for another, triggering incorrect commands.
- **Lack of Precision:** Continuous whistles could inadvertently contain partial patterns, causing unintentional actions.
- **Limited Adaptability:** Unlike continuous mapping, pattern matching struggled with natural variations in pitch and timing, making it less robust in real-world, noisy environments.

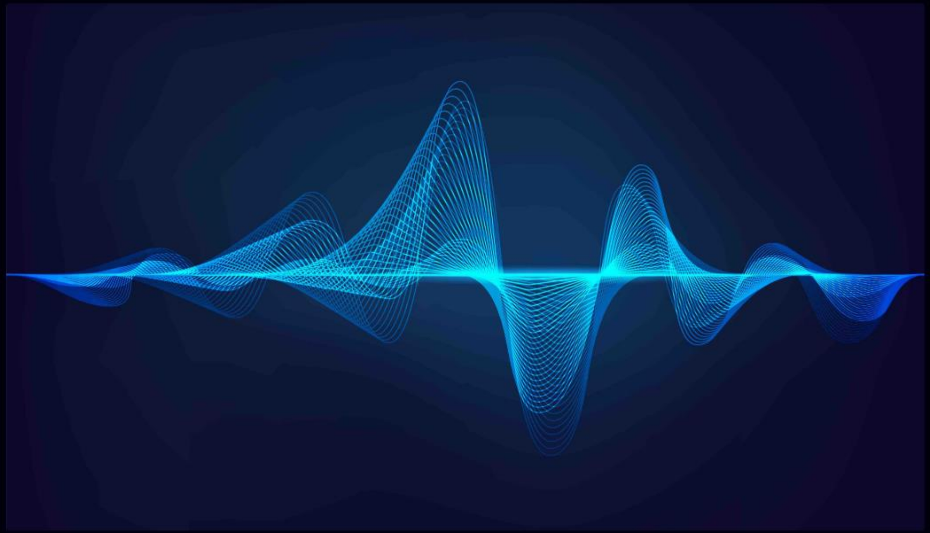
Ultimately, continuous mapping proved more reliable, smoother, and better suited for real-time drone control.

Whistle Filtering & Recognition

To ensure robust and accurate drone control, we implemented a hybrid approach combining traditional Digital Signal Processing (DSP) with an AI classification model. This two-stage system effectively isolates and identifies the intended pilot's whistle while rejecting unwanted noise and other sounds.

DSP Audio Processing

- **Band-pass Filter:** Applied between 100–3,000 Hz to isolate the relevant whistle frequency range.
- **Real-time Extraction:** Utilized the Aubio library to extract pitch and volume data in real time.
- **Whistle-like Frames:** Only audio frames exhibiting whistle- like characteristics are passed to the next stage.



AI Whistle Classifier

- **Neural Network:** A small neural network developed in Google Teachable Machine and exported to TensorFlow Lite.
- **Audio Classification:** Classifies 1-second audio segments into three categories:
  - Class 0: Background noise
  - Class 1: Pilot’ s whistle
  - Class 2: Someone else’ s whistle
- **Acceptance Criteria:** Only Class 1 (pilot’ s whistle) is accepted for further processing.

The combined DSP and AI approach significantly reduces false triggers, with any rare occurrences resulting in minimal or no drone movement due to DSP smoothing.

System Pipeline: From Whistle to Flight

The whistle-controlled drone system operates through a meticulously designed pipeline, ensuring seamless translation of audio input into precise flight commands.

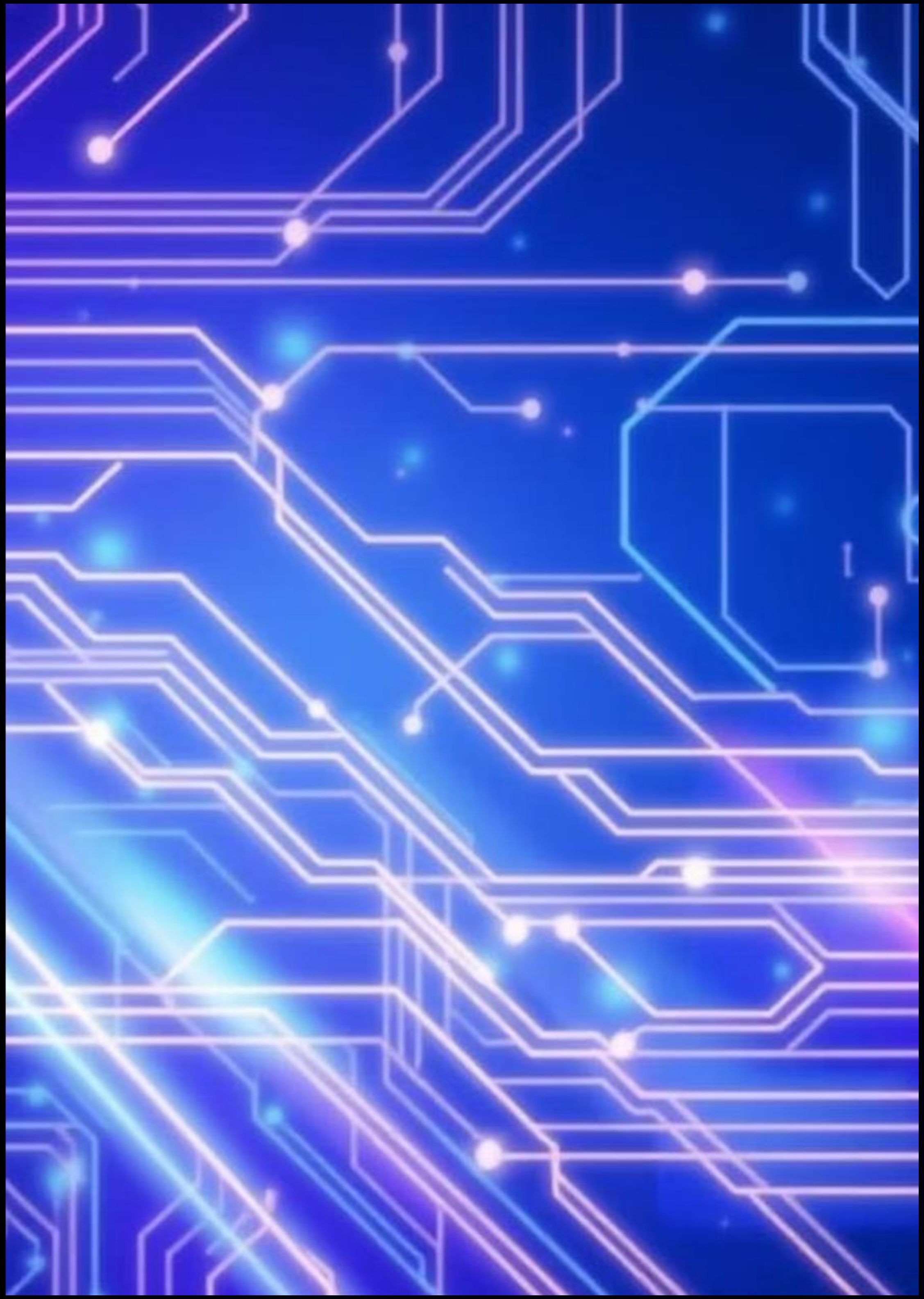
The system works as follows:

> (1) Microphone → (2) DSP Whistle Detector → (3) AI User Classifier → (4) Command Mapping → (5) DJI Tello Drone

1. Audio Capture: Microphone picks up sound continuously.
2. DSP Filtering: Band-pass filter, pitch detection, volume measurement.
3. AI Classification: Checks if whistle is from the pilot.
4. Command Mapping: Converts pitch/volume into forward speed, altitude, rotation.
5. Flight Execution: Sends commands to DJI Tello.

Safety: Auto-hover on silence, auto-land on audio loss - always active for protection.

This integrated pipeline ensures a responsive and safe control experience, with built-in failsafes for audio loss or silence.



Problems Encountered & Solutions

During the development process, we encountered several technical challenges that required specific solutions to ensure the system's reliability and performance.

Pattern method unreliable	Switched to continuous mapping for improved precision.
DSP let noise through and couldn't classify pilot whistle	Added AI pilot classifier for robust filtering and identification.
Small whistle dataset	Used lightweight platform/model suited to limited data, ensuring efficiency.
control instabilities, particularly during turns.	Tuned parameters, implemented PID (Proportional–Integral–Derivative) control, and applied smoothing algorithms to ensure stable flight.

Testing & Results

Rigorous testing, both in simulation and real-world environments, validated the effectiveness and reliability of the whistle-controlled drone system.

80%

Pilot Distinction Accuracy

The system achieved approximately 80% accuracy in distinguishing the intended pilot's whistle from others.

100%

Whistle-Controlled Flight

The drone successfully flew entirely via whistle commands in real- world tests.

0

Dangerous Movements

No dangerous or erratic movements were observed, even in noisy lab environments.

The combination of DSP and AI significantly reduced false triggers, contributing to a safe and controlled flight experience.

Limitations & Future Improvements

While the project successfully demonstrated the feasibility of whistle- controlled drone interaction, certain limitations were identified, paving the way for future enhancements.

1

Dataset Size

Recognition accuracy is currently limited by the relatively small and less varied whistle dataset used for AI training.

2

Future Improvement: Expanded Dataset

Collecting a larger and more diverse whistle dataset will enhance the AI's recognition capabilities and overall system robustness.

Further research could explore advanced machine learning techniques and real-time adaptive algorithms to continuously improve performance.

Acknowledgements

This project was made possible through the invaluable contributions of various individuals and the utilization of open-source tools and libraries.

Open-Source Tools

We utilized the following open-source libraries in our project: **djitellopy**, **Aubio**, **SciPy**, **Sounddevice**, **NumPy**, **TensorFlow**, **TensorFlow Lite**, **json**, **time**, and **threading**.

Google Teachable Machine

The AI whistle classifier was effectively trained using **Google Teachable Machine**, a powerful and accessible platform.

Advisor & Volunteers

Special thanks to our advisor, Mr. Idan Tobis, all whistle data volunteers, and the lab support team for their dedicated contributions.