**Phase 1 Report Guidelines:** For Phase 1, submit a report showcasing that your project is at most halfway completed. This should include evidence of progress, such as a video or detailed report, confirming that the simulation, programming, and environment setup are operational. This phase is crucial to ensure you have a solid foundation for Phase 2 and are on track to complete the project on time. Any deviations from the initial setup should be avoided to maintain continuity for Phase 2.

**Project Overview**

The goal of this project is to develop autonomous capabilities for the DJI Tello drone, enabling it to perform surveillance and target acquisition tasks.

* Implement gesture/command recognition
* Enable autonomous navigation and obstacle avoidance
* Develop target identification and tracking behaviors

We will utilize machine learning techniques including neural networks for classification and reinforcement learning for autonomous behaviors. The Webots robot simulation environment will be used for initial development and testing.

**Technical Approach**

Gesture/Command Recognition:

- Develop a neural network classifier using PyTorch

- Train on image dataset of gestures/commands

- Integrate with drone's video stream using OpenCV

Autonomous Navigation:

- Implement deep Q-learning algorithm

- Train in Webots simulation environment

- Develop reward function for navigation and obstacle avoidance

Target Acquisition:

- Extend neural network classifier for object/target recognition

- Implement search and tracking behaviors using reinforcement learning

Software Stack:

- Python programming language

- DJI Tello SDK for drone control

- PyTorch for machine learning

- OpenCV for image processing

- Webots for simulation

**Phase 1 Report: Autonomous Surveillance and Target Acquisition Behaviors with DJI Tello**

Development Plan

**1. Set up Webots simulation environment with DJI Tello model**

We began by downloading and installing Webots, an open-source robotics simulation platform. After installation, we connected Python to develop our code and manage necessary packages. We also downloaded the Deep Bots library to enhance our simulation capabilities. This initial setup allowed us to create a robust environment for simulating drone behavior and control.

**2. Implement basic drone control using Tello SDK - VIDEO ATTACHED 🙂**

Next, we focused on implementing basic drone control through the Tello SDK. We successfully configured the keyboard for manual movement of the drone, allowing us to maneuver it in the simulation environment. During this phase, we utilized the Crazyflie model in Webots, which closely resembles the DJI Tello, ensuring that our testing was relevant and applicable to our target drone. We have discretized the movements of the drone within the simulation to mimic Tello’s SDK.

**3. Develop and train gesture recognition neural network**

We progressed to developing a neural network aimed at gesture recognition. Our current focus is on training the network to differentiate between simple shapes, specifically rectangles and spheres. This step is crucial as it lays the groundwork for more complex gesture recognition tasks that will enable intuitive control of the drone based on user gestures.

**4. Implement deep Q-learning for autonomous navigation**

*Pending…*

We are planning on moving forward with implementing our deep Q-learning model this week, building upon our discretized controller in Webots.

5. Extend to include target acquisition behaviors

6. Integrate and test full system in simulation

7. Transfer to physical drone and real-world testing

Potential Challenges and Mitigations

- Simulation fidelity: Will validate sim-to-real transfer early

- Model performance: Will start with simple behaviors and incrementally increase complexity

- Real-world robustness: Extensive testing in varied environments

Next Steps

- Finalize simulation environment setup

- Begin implementation of gesture recognition network

- Start development of reinforcement learning navigation

We are on track to demonstrate basic autonomous capabilities by the project deadline. The team will provide regular progress updates as development continues.