**Capstone Proposal Final Report**

Group: 24-013

Seven Hills Adaptive Drone

Sponsor Staff: Steven Kessler, Ryan Beckett

Mentor: Professor Jean-Francois Millithaler (ECE)

Team Members: Himadri Saha (CpE), Abumere Okhihan (CpE), Tejas Patil (EE), Daniel Diep (EE)

Date: December 13th, 2024

Course: EECE.3991

**Project:** Adaptive Drone

**Sponsor:** Seven Hills: Steven Kessler, Ryan Beckett

**Mentor:** Professor Jean-Francois Millithaler (ECE)

**Team Members:** Himadri Saha (CpE), Abumere Okhihan (CpE), Tejas Patil (EE), Daniel Diep (EE)

**Who:** This project is created for Seven Hills, a company committed to offering innovative support to individuals living with disabilities.

**Problem:** The problem at hand is that the wide population of people who live their lives dealing with a disability are not able to experience certain aspects of life. Seven Hills requires an adaptive drone that can be controlled by someone living with a disability.

**Solution:** The solution is to create a computer-based application that utilizes facial recognition technology to allow a user to control a drone. Tracking face movements allows someone living with a disability, such as quadriplegia, the ability to do something they may have never had the experience of doing before.

**Benefits:** This solution not only addresses the effects of quadriplegia but also supports individuals with other disabilities. Its primary advantage lies in promoting user autonomy by enabling independent photography without requiring physical input. Using facial expression recognition via webcam, it is non-intrusive and does not depend on wearable devices or specialized hardware. The app’s adaptability enhances accessibility, with MediaPipe’s algorithms ensuring reliable performance across diverse face shapes and lighting conditions. Additionally, the reliance on MediaPipe and OpenCV makes the solution cost-effective, eliminating the need for extra hardware—a crucial benefit for users with limited resources. As an open-source platform, it invites developers to customize and expand its functionality, enabling broader applications for different drones and diverse user needs while fostering innovation.

**Our Progress:** So far, our team has been working on getting the face tracking software to work more efficiently and accurately. The DJI Tello drone has been lent to us to research and work on ways to control the drone using software commands utilizing the Tello’s built in SDK (the drone’s specific command set) rather than the physical remote controller. The next step is to create our own application that can be used as an interface between the user and the drone. Then we will have to obtain feedback from the drone’s camera so that the user can see what the drone sees through the app.[

**I. Introduction:**

The aim of the Seven Hills Adaptive Drone Project is to create a new solution that allows people of physical disabilities, especially those with quadriplegia, to control a photography drone for the purpose of enabling more interactions with their environment which otherwise would be impossible. It is important to ensure that anyone can control the drone since people with quadriplegia – people who suffer from paralysis in both arms and legs – are limited in performing basic activities and enjoying leisure. Advances in assistive technologies are opening new doors to these people, giving them the ability to do more for themselves. The aim of this project is to devise a system where users would be able to navigate a drone through non-invasive means, such as head movements or facial recognition.

The sponsor, Seven Hills Foundation, is highly committed to offering innovative and comprehensive health services to people facing disabilities in their lives. For more than 60 years they have been committed to improving the quality of life for people. Seven Hills has a wide range of services, from educational to clinical health support. They aim to develop solutions that are person-centered, enabling users to achieve control and dignity in their daily lives. Having experience in supporting disabled individuals, they were the perfect fit for this pioneering project.

The following report will outline the primary problem our project aims to address, and the corresponding objective designed to provide a solution. It will also include details about the sponsor, an overview of relevant research, and a comprehensive analysis and design of our proposed solution, explaining how each component of the system will function. Throughout this report, we will discuss the project’s background information and the benefits our future system will deliver. Additionally, the report will explore potential solutions considered during the research phase. Lastly, we will outline the specific requirements that our chosen solution must meet and detail the deliverables expected by the project deadline.

OpenCV can be used to implement face recognition or gesture-based control for a completely hands-free interface to the drone. It uses very modern technologies to deliver ease for the quadriplegic to have the solution implemented, using a minimum amount of physical input while maximizing his independence.

**II. Company and Client Information:**

Company Info

Seven Hills Foundation is a human service organization committed to promoting and providing comprehensive health and human services for individuals with disabilities and life challenges. From its humble beginning more than 60 years ago, Seven Hills has grown to become a leading full-service provider of clinical, educational, and behavioral health supports, with person-centered practices promoting empowerment and dignity in all aspects of service delivery. From its New England base and international programs, the foundation continues to expand its scope by delivering tailored care to people of any age. The sponsors Steven Kessler and Ryan Beckett of Seven Hills Foundation, Middleton MA, both of whom are specialists in assistive technology, have instructed the group of the project to create an adaptive drone controller so that those with disabilities without the use of limbs can control and take photographs with a photography drone.

Client Info

Although there is no available client for the project, due to issues with contact earlier in development, the sponsors have assured us that it should be for those with quadriplegia. A physical disability where people cannot move their limbs from the neck down. The system will allow drone control for users with this condition, potentially benefiting others with varying physical disabilities. Due to no specific location listed for a client, the final product will be delivered to Seven Hills Foundation, Middleton, MA, so that it may be used for those who need the system to interact with the environment later.

**III. Project Information:**

Initial Description

This project addresses the need for a non-invasive, accessible solution enabling quadriplegic individuals to control a photography drone. Although there is not a specific client, a clinician will be with the user during the operation of the system. Recognizing the challenges quadriplegic individuals face with traditional two-stick controllers, Seven Hills requested a system that empowers users to operate a drone independently, requiring only minimal assistance for setup. The solution should also be versatile, with no additional customization needed to accommodate a diverse range of users. While the ability of each user will be different, the user must not need to use any of their limbs.

Expected Deliverables

The primary deliverable is a fully functional app capable of detecting and interpreting ten distinct facial expressions, with each expression mapped to a specific command for controlling a DJI Tello drone in photography. The app will leverage facial recognition technology to ensure accurate expression detection across various face shapes and lighting conditions. Alongside the app, an Autodesk Instructable guide will be provided to support setup and operation for users with minimal technical background. This guide will clearly outline steps for installation, setup, and app use, ensuring accessibility for all users. Developer documentation, in the form of a comprehensive Readme file, will also be included. This documentation will cover the app’s architecture and the integration of OpenCV, MediaPipe, and the Tello SDK, along with guidance for customizing the codebase for different drones and commands. Such documentation will allow developers to modify and expand the app for broader control and compatibility. To extend the app’s compatibility to other drones, a USB RF transmitter will be optionally available, enabling control over standard RF drones that lack the Tello SDK.

Benefits

The primary advantage of a facial expression-controlled solution is that it promotes user autonomy by enabling independent photography, eliminating the need for any physical input during operation. By utilizing facial expression recognition via webcam, the solution is non-intrusive and comfortable, requiring no wearable devices or specialized hardware. Additionally, the app’s adaptability across users enhances its accessibility. MediaPipe’s algorithms create a face mesh that functions effectively across different face shapes and lighting conditions, ensuring the app can be used by a broad range of individuals without customization. The use of MediaPipe and OpenCV for facial recognition also makes this solution highly cost-effective, as it eliminates the need for additional hardware. This affordability is particularly beneficial for users with limited resources or technical expertise. Lastly, the open-source nature of the software expands its impact. The flexibility for developers to modify and enhance the app means it can be adapted for various drones and tailored to support users with diverse needs. This adaptability broadens the technology’s application, making it beneficial for a wider audience and fostering further innovation.

Expected Success Criteria

The expected success criteria for the face-expression-controlled photography drone project encompasses three primary aspects: functional drone control, photography capabilities, and integrated safety features. First, the drone must be controllable using distinct facial expressions, ensuring intuitive and reliable operation. Success will be measured by the drone’s responsiveness to predefined facial expressions. Each control must be triggered accurately with minimal delay and must maintain stability during operation.

Second, the drone must effectively capture photographs using the appropriate facial command. The photo-capturing mechanism should function seamlessly within the control interface, with clear feedback indicating successful image capture. Image quality will be assessed based on clarity, resolution, and the absence of distortions caused by movement or environmental conditions. The integration of this feature should enhance user experience by allowing hands-free operation while ensuring high-quality photographic results.

Lastly, the project must incorporate safety features to prevent risks from potential malfunctions. These features include automatic emergency land functions triggered by abnormal system behavior or communication failures. Additionally, the system should alert users in case of critical errors, providing real-time feedback for troubleshooting. These criteria define the operational, functional, and safety benchmarks necessary to consider the project a success.

**IV. Research**

Facial Recognition and Expression Detection Technology overview

Facial recognition technology is a method of identifying or verifying a person's identity by analyzing and comparing patterns based on their facial features [1]. A facial recognition model uses algorithms to map facial features (landmarks) such as the eyes nose and mouth and converts them into a mathematical model. Artificial neural networks are often used to process these models because many mathematical operations needs to simultaneously occur [1]. Facial expression detection technologies analyze these mathematical models to infer emotions. Using computer vision techniques, these facial landmarks are tracked and parameters such as distance and angles are calculated [2]. All these technologies generally follow these phases:

Data Collection: Images or videos of faces are captured using cameras. In applications where the user could potentially be moving, the face expression model will use individual frames from a video feed (a webcam for example). Image processing will also occur here, with some face landmarking models requiring a black and white image overlayed with a color image.

Face detection and feature extraction: From each image, a face detection algorithm will detect the presence of a face by identifying facial features. Calculations of facial landmarks will also occur here.

Expression classification: The extracted landmarks are then processed by machine learning algorithms trained to classify different expressions. In general, these algorithms map the facial features to specific emotional states such as happiness, sadness, surprise and fear [3]. These algorithms are trained on image and 3D facial scan datasets that contain many different examples of human faces. While these datasets are quite extensive, they occasionally have issues with a lack of diversity, making the algorithms less efficient and impacting performance negatively. Some of these datasets are available to the public such as Extended FER2019. Others may be private and only used by certain technologies such as Face++ used by Megvii and NEC [1].

With improvements and increased accessibility of artificial intelligence technology recently, facial detection technology has also improved. This technology is found in many different applications, from robotics, healthcare and human-computer interaction [4]. Facial expression technology has already become integrated in NAO robotics. These are small humanoid robots that are equipped with cameras that capture facial images through a video feed (Data collection phase), process the video frame as images (Face detection phase) to identify the individual it is looking at (expression classification). The NAO robots use the identity of the individual to adjust its actions [5]. In healthcare, facial recognition technology can be used to detect facial palsy, which is a condition that causes weakness or paralysis of the facial muscles. In this application, the same facial recognition phases are followed. However, in the expression classification phase, the facial movements itself are being tracked using the facial landmarks. For example, a system might analyze the movement of eyebrows when a person smiles. If the eyebrows do not move normally (which is dictated by a dataset), then this could be flagged as a sign of facial palsy. Because of the computationally intensive nature of these systems, the analysis time is reduced by prioritizing specific time intervals where an emergency sign is detected [6].

Facial Landmark Calculations

Understanding how to make calculations with these facial landmarks is essential for the classification stage of these systems. Each facial landmark is mapped using a set of (x,y) coordinates on the 2D frame from the input phase. This creates a structured grid that represents the users’ unique facial features. The face detection phase begins with a landmark detection algorithm that identifies these landmarks.

A common algorithm is Dlib’s Facial Landmark Detector. Dlib’s uses a set of regression trees to detect 68 specific landmarks [7]. Before detection of the user’s face, Dlib will first detect the face in the image using a centralized neural network. This focuses the algorithm on just the users face in the image, making it more effective. Dlib’s starts by estimating an initial “average” face shape based on the average position of landmarks across all training images in a dataset. The initial face shape serves as a reference to adjust the landmarks as the face moves. The regression trees will then iteratively refine each landmark position by analyzing the pixel information around each point. For each iteration of the algorithm, the positions become increasingly accurate. After several iterations, the final landmarks will be laid out for the user to see. In a video application where the users face is being tracked by a webcam for example, the initial interactions of the algorithm might be completed in a couple of seconds. The speed and efficiency of Dlib’s is the main reason it is so popular is face recognition systems [7]. However, due to Dlib’s approach with analyzing local pixel information around each initial landmark, it can be sensitive to lighting and extreme face angles.

Another commonly used algorithm is MediaPipe. It uses a deep learning-based approach for its facial landmark detection rather than using regression trees. MediaPipe’s algorithm consists of 4 stages: detection, landmark localization and tracking [8]. To detect the users face, MediaPipe’s lightweight neural network will use the same method that Dlib’s employs. Then it creates a bounding box around the detected face, isolating the image to focus only on relevant areas for processing. Using the detected face, the system lays 468 facial landmarks, covering finer details compared to Dlib’s. As multiple frames are inputted to the system, MediaPipe tracks landmarks with temporal consistency. This means MediaPipe will maintain landmark positions over multiple frames, only recalculating landmarks if there is a significant change [8]. This reduces any ‘jittering’ effects for a more stable output.

A key feature that is not in Dlib’s algorithm is that MediaPipe only recalculates the location of landmarks that made slight changes rather than recalculating all landmarks for every frame. This improves the speed of the algorithm significantly, making it suitable for applications that require fast processing. In addition, MediaPipe’s speed and efficiency, it also allows for 3D face citations, making it suitable for applications where the user’s face might be at an extreme angle from the input device. When comparing MediaPipe and Dlib’s efficiency in general applications, MediaPipe has a clear advantage. Although MediaPipe’s effectiveness is sensitive to extreme lighting conditions, it is no less than other algorithms like Dlib’s. Despite being computationally expensive compared to Dlib’s, MediaPipe’s extensive optimizations make it well worth it.

Facial-emotion-Recognition Example

A diagram of a process

Description automatically generated with medium confidence

*Figure 2: Software Architecture for Facial motion recognition (Accessed [9])*

A project on GitHub by Manish Tiwari uses OpenCV and DeepFace library to create a real time facial emotion detector [9]. This system uses the video feed of a webcam as then input. It detects faces and predicts the emotions with each facial expression. It uses both computer vision techniques and deep learning algorithms to classify emotions in a very simple way. To provide the pre trained algorithms, the library DeepFace is used. It uses TensorFlow to simplify the backend computations that these algorithms would use. OpenCV is used to handle the video capture and image processing function. Most notably, it provides the function to create a bounding box around the detected users face to focus the DeepFace algorithms on. Just like many facial recognition systems, it follows the same phases: Data Collection, Face detection and feature extraction, and Expression classification.

First, the system uses OpenCV’s Haar Cascade algorithm to detect the user face from a grayscale frame in the video feed. This algorithm “can detect objects in images, irrespective of their scale in image and location” [10]. Haar Cascade’s is also optimized for real time processing, so it works effectively in this application where the input is a high frame rate video feed. Next, the system isolates the face region (Region of Interest or ROI) and uses DeepFace’s “analyze” function to process the users face. DeepFace’s analyze function will then predict the associated emotion of the user. Finally, using OpenCV, a bounding box is drawn around the ROI and is overlayed for the user to visualize, along with the predicated emotion.

This system is efficient and compact, achieving real time emotion prediction with relatively minimal code. Because these libraries are also prebuilt with face detection models, developers do not need to program their own algorithm making it easier to focus on other areas of development. Because of this, however, the issues with lighting and face angles will be difficult to address without changing these library functions themselves. It also relies on one specific algorithm and dataset, potentially causing issues with users faces that are less represented in the tested dataset. Despite these drawbacks, this codebase proves a strong example of how one would go about developing a facial recognition system. Most notably, its use of OpenCV is very standard among many facial recognition systems.

Drone Control Systems

Quadcopter Drones rely on control systems to manage flight paths, stability and functionality. These systems use a combination of hardware and software to interpret user inputs. The main components of a drone control system are the onboard flight controllers, sensors, and input mechanisms [11].

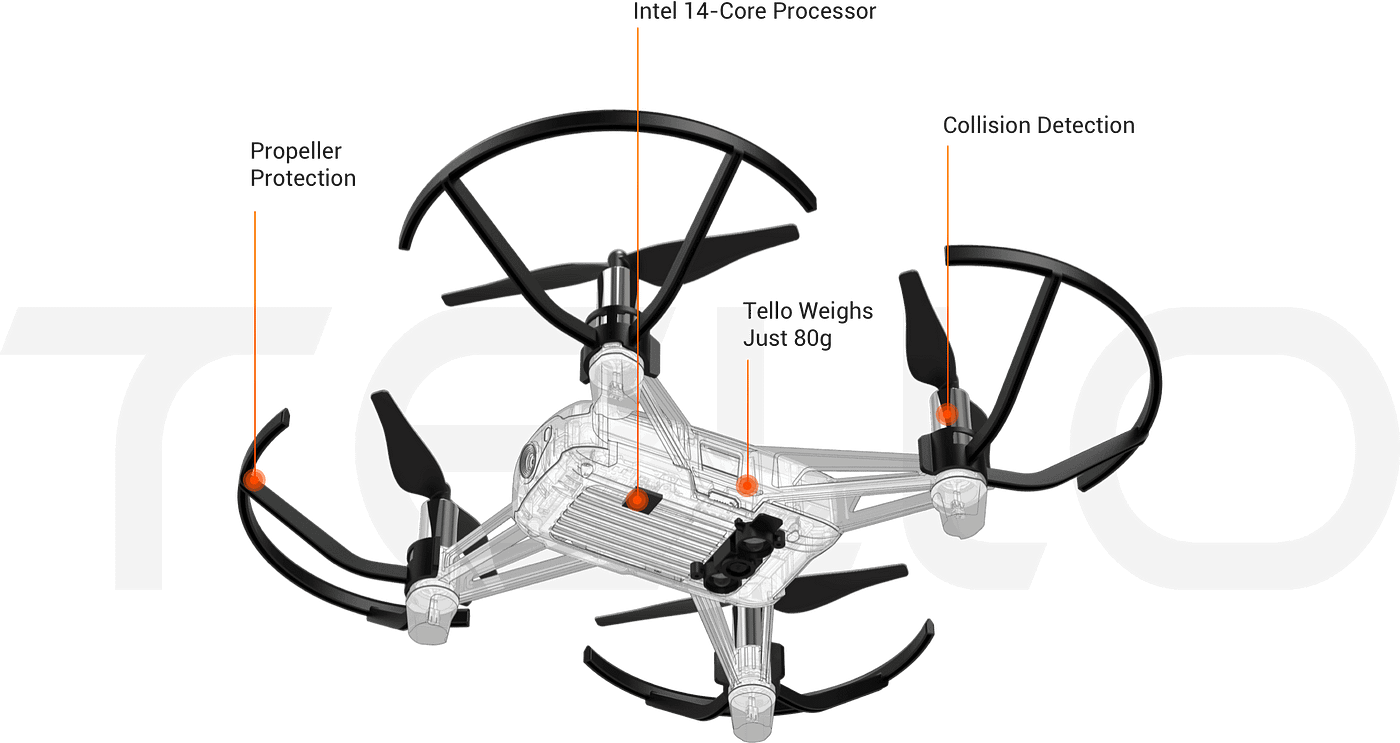
The onboard flight controller acts as the brain of a drone, managing all drone systems and processing input commands to make the desired flight paths [11]. Modern controllers are integrated with various sensors to maintain altitude, control speed, and adjust orientation. Onboard sensors can include accelerometers, gyroscopes, magnetometers, and GPS modules.

For quadcopter drones, users must be able to control movement along different axes. More specifically, forward/backward, left/right, up/down, and yaw are the primary movements a drone should be able to perform [12]. Modern drones may also have a combination of real time user control and autonomous systems for safety features. For example, many drones have a land feature. When triggered, the users’ commands are ignored, and the drone will safely fly down until a sensor can identify that the drone has landed.

When drones are controlled using physical input systems, such as remote controls or mobile applications, specific commands corresponding to the drone’s primary movements are transmitted to the drone [12]. Traditional two-stick controllers are commonly used due to their ability to provide precise manual control. While effective, these controllers are not accessible to individuals with mobility challenges. However, they offer a fundamental understanding of how user inputs translate to drone movements.

For instance, the left stick on a two-stick controller typically controls the drone’s altitude (up/down) and yaw (left/right rotation). The altitude function does not recenter to a neutral position, allowing the user to focus on other movements without continuously maintaining the desired altitude. Conversely, the right stick, which controls the drone’s forward/backward and left/right lateral movements, automatically returns to a neutral position when released. This means that movements such as turning left or right should not require active maintenance by the user, regardless of the control system.

Tello Software Development Kit



*Figure 3: DJI Tello Outline (Accessed [14])*

The DJI Tello is a small, lightweight quadcopter drone that is generally used by hobbyists and in educational cases [12]. It also has an onboard 5-megapixel camera that can take photos and record videos. The key features of Tello are being programmable with the SDK of the Tello drone. The Tello SDK is one of the most interesting tools, which lets the users stop operating drones and get more from it. So, we can think that Tello SDK is like a translator that makes code understandable for the Tello. It has a set of libraries and functions that give you full command of drone hardware [12]. That means developers are no longer constrained to pre-programmed flight modes but can design custom flight patterns, automated maneuvers, and include data from sensors of the drone in your programs.

The SDK utilizes Wi-Fi to establish communication between your computer and the Tello drone. It uses the UDP protocol, which prioritizes speed, making it suitable for real-time control. Developers don't need to worry about the technicalities of UDP communication as the SDK handles the details of packaging data, sending commands, and receiving responses [14]. The Tello SDK is designed to work seamlessly with Python, a popular programming language known for its readability and extensive libraries [13]. Several Python libraries simplify the use of the SDK even further such as Tello-Python, DJITelloPY, TelloPy, and easyTello. Tello-Python is the official library from DJI, offering a comprehensive set of functions of the drone’s control. DJITelloPY is the community developed library that offers even more user-friendly interfaces such as prebuilt flight maneuvers and telemetry data access [12].

Overall, this SDK is the key to unlocking DJI Tello’s programmable capabilities, allowing developers more fine-grained control over its flight and access to sensor data, without having to program everything from scratch. The Tello SDK’s application extends beyond basic control, allowing developers to enable computer vision onboard the drone, allow for autonomous navigation, and integrate artificial intelligence models [12].

**V. Problem Statement**

The project aims to create an adaptive drone control for people with physical disabilities who are not able to interact with their environment. People with physical disabilities have major difficulties connecting with their surroundings such as those with forms of quadriplegia who cannot use their limbs below the neck. Our project will ensure they can control a drone without limbs to explore the world and take photos that they never could have done before.

*People with physical disabilities such as quadriplegia have difficulties interacting with the environment. It is desired to have an adaptive photography drone controller that will provide that experience without requiring the use of limbs.*

**VI. Objective Statement**

*The objective is to create a set of controls for a photography drone that a person with physical disabilities can use by May 2025.*

**VII. Requirements**

[1] A facial recognition system used to control a photography drone.

Rationale: Requested by the sponsors, a photography drone that can be controlled by a quadriplegic is desired.

[2] OpenCV and MediaPipe based facial detection system: The facial detection system will detect facial movements as input commands to control the photography drone.

Rationale: This facial detection system is the best solution for the problem without the user needing to use their limbs.

[3] App Design: There will be a PC based application that works best using a DJI Tello photography drone.

Rationale: The DJI Tello photography drone is easy to use and is the easiest drone to operate with the application.

[4] Flight Controller System: The photography drone will also be able to be controlled using inputs such as keys from a keyboard.

Rationale: This is a secondary way of controlling the drone that does not require facial recognition.

[5] Onboard Drone Camera: The PC based application will be able to obtain video feed from the Tello drone so the user can see what the drone sees.

Rationale: The ability to see the drone’s perspective will allow the user to interact with the environment.

[6] Instructions and Safety: The application will come with an Autodesk Instructables guide on proper setup and operation to ensure accessibility for all users.

Rationale: Proper setup for the system is vital to the product working as intended.

[7] Return to home: In the event the drone cannot detect the users face after ten seconds, the drone should land safely.

Rationale: To ensure safe flight and minimal damage to the drone, this safety feature prevents any unwanted errors when the user is unable to control the drone.

[8] Drone Maintenance: The drone should be fully charged before use and left to be cooled down after use.

Rationale: The drone model given works best when fully charged and tends to heat up after use.

[9] Project Budget: The money spent on the development and research for the system will not surpass the $500 limit.

Rationale: The budget provided by the University of Massachusetts Lowell is $500.

[10] Open-source code: In the case of users not having the Tello drone, there may be open-sourced code to access other drones with guided instructions.

Rationale: Different drones have flight control systems different from the Tello, so steps will be provided where others can have the same opportunity to create a similar system.

[11] Labeling the System: The application and Autodesk Instructables guide will include our names, capstone group number, and UML.

Rationale: To credit the creators of the system and the University.

**VIII. Technical Details and Analysis**

Analysis of Multiple Design Choices

A description of two hands-free control solutions can be applied to increase accessibility and comfort in operating the DJI Tello by persons having various levels of mobility impairments. These are the following:

Face Detection Controller Solution Analysis

The first proposed solution is the development of a python based mobile or PC-based app that allows users through facial and hand expressions to give commands to the controller. In this application, a live FPV display, and media storage management are integrated, including Deep Face for face detection, all to enable compatibility with the DJI Tello and other drones for scalability. This uses the Google MediaPipe framework for improving expression detection. MediaPipe can detect up to 468 facial landmarks for explicit recognition of expressions that can be mapped onto specific drone commands. Stitching Deep Face for basic face identification, to MediaPipe high precision tracking of facial expressions, the implementation of the face detection controller approach can enable an intuitive hands-free experience that can be very useful for users with disabilities by enabling features such as the camera roll for managing media. Real-time processing of expression input may bring in some latency, which would reduce the smoothness of control. A drone model called the DJI Tello communicates using the Tello SDK via a Wi-Fi UDP protocol, which though simple and connectionless, can easily result in packet loss and hence unreliable commands. Besides, the limitation of Wi-Fi itself may constrain the effective range of control, though error-handling mechanisms within the SDK improve reliability during operation.

VIVE Headset Solution Analysis

The second solution explored was the use of a VIVE Headset for drone control. The VIVE headset provides a solid, immersive experience by allowing users to control the drone via head movements. This can potentially enable hands-free control through mapping these movements to specific drone commands by leveraging inbuilt sensors in the headset that detect orientation and motion. However, although this solution is high in precision and immersion, it was not chosen because of a couple of practical limitations that come with it. First and most importantly, this would be since, to date, the VIVE headset must be connected to a computer, making it a bit less portable or practical for outdoor use regarding drone operation. Additionally, VIVE systems are more expensive and demand high-end hardware, further limiting accessibility to those who have no access to VR equipment. Considering the above-mentioned disadvantages, the VIVE headset was not considered to be an appropriate tool for a hands-free drone control solution with broader accessibility in mind.

Chosen Solution Justification

After weighing both solutions, Face Detection Controller was chosen to be the go-to approach. This makes much more sense as it has lower power consumption than compared to the VIVE headset approach and will provide a smooth, seamless experience on a wide array of devices. The facial detection expression controller is handled using MediaPipe which offers more customization in comparison to the VIVE headset. Also, the immersive capability of the VIVE headset itself presents limitations in portability and cost that are counter to the goal of accessibility embraced in this project. Therefore, the Face Detection Controller finds a balance between usability, scalability, and compatibility, making it an attainable option for users with low-end hardware.

In addition, further optimization of the Face Detection Controller system will be oriented to real-time responsiveness, enhancing the accuracy of expression detection, and compatibility across devices with different computational capabilities. These improvements will culminate in an efficient and user-friendly interface that will provide an effortless hands-free experience for end-users, especially those with limited mobility.

Chosen solution

As stated previously, the chosen solution will be to create a Python-based PC application that will detect user’s facial expressions and translate them into drone commands for the DJI Tello drone. The application will use OpenCV to detect the users face within the webcam live feed, and MediaPipe’s facial mesh technology to analyze the finer details of the face to detect the facial expressions. The systems architecture consists of four main modules: the facial detection module, the DJI drone controller module, the app architecture module and the expansion/compatibility module. The expansion/compatibility module is to allow for future scalability, and while it is not a requirement for the project’s objective, it will allow for this system to be used for other drones. The focus of this project is on the first three modules, and if time allows, the fourth module may be developed. Additionally, the approach to develop facial detection system explained below will be reviewed for computational efficiency, and major changes are expected during the development process.

Face Detection Module

The face detection module is responsible for interpreting 11 distinct facial expressions in real time. Eight will be for controlling the drone (forwards, backwards, left, right, ascend, descend, rotate right, rotate left), two for controlling the camera (capture photo, capture video) and one multipurpose command that can be mapped for development purposes or further user personalization. In addition to facial commands, a keyboard-based control feature will also be enabled so that an assistant could take control of the drone. This is a safety feature in the event of an emergency as well as a feature to streamline testing the system. Upon first time use, the user will calibrate the facial detection system during the “setup phase” with the help of an assistant. The application will prompt the user to make a facial expression for a corresponding drone command. Then the assistant will press the save key to store the facial recognition data in a Python dictionary, with the keys being a corresponding drone command symbol (characters A-K). This repeats all 11 expressions. Expression data is collected with the function get\_expression\_data which calculates the distance between each landmark from the tip of the user’s node. Each distance is stored in an array, which represents the expression data.

After calibration, the live phase will start where the user controls the drone. The system tracks the users face in real time using a webcam and overlays the MediaPipe face mesh. For every frame, collects the expression data and compares it to each saved expression using a similarity score calculator. Each saved expression also has a corresponding threshold assigned to it. If the most similar saved expression compared to the user’s current expression has a similarity score greater than its corresponding threshold, then the system can detect that a valid expression has been made, and the corresponding drone control command will be sent to the drone. Having a threshold will ensure that expressions that might be similar do not overlap with output commands. For example, if the system is inaccurate in differentiating between an open mouth and a smiling face, a small threshold value representing the distance of between a user’s lips will be placed to ensure that a user must clearly open their mouth for an open face expression to be detected. To calculate the similarity between any two expressions, the algorithm takes two lists of facial landmark distances: the live distances detected in real time and the calibrated distances obtained during the setup phase. For each corresponding pair of distances, the algorithm calculates the absolute difference, representing how far apart the two measurements are. These differences are summed and then normalized by dividing the total difference by the number of measured landmarks. This produces a normalized similarity score, which reflects the overall deviation between the live and calibrated distances.

A diagram of a algorithm

Description automatically generated

*Figure 4: Live Phase flow chart*

*A diagram of a program

Description automatically generated*

*Figure 5: Similarity score calculation flow chart*

DJI Tello Control Module

The DJI Tello module serves as the central control interface for the drone in our project, using the DJI Tello SDK for seamless command execution. Currently, the module processes keyboard inputs to issue drone commands, though these will eventually be replaced by outputs from the facial detection module. The system operates through three main communication channels: command control, state monitoring, and video streaming. The command control channel sends operational instructions such as takeoff, landing, movement, and rotation. The state monitoring channel continuously receives telemetry data, including battery status, altitude, and motor activity, enabling real-time status updates. The video streaming channel captures and displays the drone's first-person view (FPV) feed.

The module’s internal structure ensures robust drone management through socket communication. Background threads listen for both command responses and telemetry data, ensuring uninterrupted control. Commands like takeoff, land, and move are issued through encoded UDP packets, with retries built in for fault tolerance. For enhanced situational awareness, the FPV feed is managed through the PyAV library, streaming video frames in real time. In the future, keyboard inputs will be replaced by corresponding drone commands derived from facial movement detections, enabling fully hands-free drone operation. This modular design ensures scalability, allowing future integration of advanced control features such as waypoint navigation and object tracking.

A diagram of a company

Description automatically generated

*Figure 6: DJI Tello Control Module process for controlling drone*

App architecture module

The Homepage is the most important part of the DJI Tello drone app because it's the starting point from which users can access all features in the app. It has been designed to make things easy and simple to handle for anyone. The basic idea of this application is that it opens to a homepage from where the user can select one of two sections: Main Control Interface and Settings.

The user would enter the Main Control Interface to control the drone directly. It includes the features Facial Tracking, which would enable the drone to follow the person's face, and Camera Feed to show live video from the drone's camera. These features were designed to be basic and functional so that everyone can make use of them. Settings: The settings include setting up the drone for flight. This includes the Drone Movement Mapping, where a user can alter controls like assigning different buttons to different actions. It also has a Calibration tool intended for facial tracking. These make the flying of the drone much more stable and easier to control.

A diagram of a device

Description automatically generated With everything set up, users can return to the Main Control Interface and take flight. They can enable Facial Tracking, watch live video from their drone, or try out the custom controls. That is just about the basic idea behind this app-the ease with which anyone can fly it for fun.

*Figure 7: DJI Tello Graphic User Interface (GUI) Diagram*

**IX. Project Reflection**

Budget Estimate

Although Professor Alessandro Sabato, of the Mechanical and Industrial Engineering department, has generously provided the DJI Tello drone model for this project, there used to be a possibility that the drone may need to be replaced, as will be discussed in the report’s section on challenges and critical issues. The DJI Tello model has been out of stock at DJI since the Summer of 2024, and the store has yet to restock it. Although the listed price on the DJI site is $99, limited availability since August 15, 2024, has driven prices up to between $146 and $157, according to Google trends, from other sellers. To mend one of the issues, an Alfa AWUS036ACH Wi-Fi adapter alongside two 10dBi RP-SMA Male antenna (15.5 in) attachments will be able to remedy an issue presented in the challenges. The Alfa AWUS036ACH Wi-Fi adapter will be able to pick up the Wi-Fi of the drone and extend it further than 300 meters. On the original supplier site, Rokland, the Alfa Wi-Fi adapter is $64.97, and the antennas compatible with the adapter are $9.99 on the site Amazon. In addition to Amazon, the same model for the adapter goes for $59.99. Since we are monitoring the drone’s physical performance, the budget for the adapter and antenna is $69.98, which stays within the budget of $500. If the drone ever needs to be replaced, it is estimated that if the stock is back up for DJI Tello drones it will go for $169.98, and without the stock being back it’d be $215.98 to $226.98. Although it stays within the budget, it’d take a significant portion.

Challenges

Initially mentioned in the midterm report that the DJI Tello’s Wi-Fi connection protocol range was limited to 300 meters (1000 feet) a way to extend it is by using a Wi-Fi adapter. Although it won’t completely solve the issue, it will allow the drone to have more freedom in exploring the surroundings of the controller.

The system needs to account for visual obstruction, such as lighting, that may interfere with facial recognition.

To ensure the control system will be ideal for using facial recognition, without having multiple commands, the system will have safety features to prevent the drone from crashing or bumping into objects which will take time to fix.

Critical Issues

As mentioned in the cost estimate section, there is a mention of possibly replacing the DJI Tello drone with a new one. The reason is that the drone tends to heat up when used, which can pose issues in the long run when testing. A cause for heating is the lack of a cooling fan for air to flow into the heat sink due to the size. The vent may not be adequate meaning a new baseplate should be created to optimize airflow into the heat sink. A STL file is available to be 3D printed to create a new baseplate to prevent overheating by MS Aerospace CCA site listed under solution for the problem by Marcus.C & Shreyas.S.

Predelivery Hazards

The DJI Tello drone depends on a firm Wi-Fi connection from the app to the drone for commands such as taking off, landing, or changing direction. If the Wi-Fi system is not tested well enough before delivery, then users are likely to face signal drops or delays. For instance, it could lose its connection to a flying drone and become unresponsive; this may lead to a crash or fly away. It is important to ensure the Wi-Fi works reliably over various ranges and environments before the product release.

When the DJI Tello drone flies, its motors and onboard systems work hard to keep it stable, especially if it's moving quickly or flying in windy conditions. This work creates heat. If the drone is flown for too long without a way to cool down, it can get too hot. When this happens, the drone might not fly as well, could lose stability, or might even shut off in the middle of a flight to prevent damage.

Postdelivery Hazards

The Tello drone must handle signals from things like phones and Wi-Fi routers. It may lose its connection to an app if there is too much interference, such as around a crowded park or city, and stop responding or just fly away. For this, the users should be warned, and the application could give a warning if they are flying in a zone that has high interference.

The Tello can only fly safely within a certain distance from the controller; it might lose connection and stop responding if it goes too far. For example, flying across a large field might seem fine, but it may crash or get lost when it is out of range. Add range warnings in the app and teach users about the limits to prevent this.

Ethical Considerations

To ensure ethical project development, the team engaged with Seven Hills and potential clients to understand the real-world implications of the proposed drone controller application. Through these discussions, it became clear that Seven Hills could not provide a client who met the exact initial requirements outlined in the project’s objectives. In response, the team collaborated with Seven Hills to adjust the project requirements. This modification expanded the application's usability, allowing multiple clients with varying needs to benefit from the drone controller. This adaptive approach ensured that the project remained client-focused while maintaining accessibility and broader applicability.

Intellectual Property Considerations

The development of this project used several open-source libraries and frameworks. Specifically, the team utilized OpenCV for computer vision tasks, MediaPipe for facial landmark detection, and DJITelloPy for drone communication and control. By using these open-source tools, the team ensured compliance while accelerating development. Proper compliance with the terms of use of each library were maintained throughout the project to respect intellectual property rights and promote ethical software development practices.

Standards Utilized

**Standards utilized by the supplier:**

‌Title 47 CFR Part 15: “Radio Frequency Devices”

This standard outlines the regulations for unlicensed radio frequency devices such as the DJI Tello drone. Since the drone communicates via Wi-Fi, it must operate without causing harm through interference with licensed communication systems.

[17] “47 CFR Part 15 -- Radio Frequency Devices,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15> (Accessed: 25 November 2024).

Title 14 CFR Part 91: “General Operating and Flight Rules”

A small part of this standard (91.119) outlines the rules for the minimum safe altitude and applies for both manned and unmanned aircraft such as the DJI Tello.

[18] “14 CFR Part 91 -- General Operating and Flight Rules” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-91> (Accessed: 25 November 2024).

**Standards utilized by our team:**

Title 14 CFR Part 107: “Small Unmanned Aircraft Systems”

This standard provides the regulations set for the operation of small drones weighing less than 55 pounds for commercial and non-recreational purposes in the United States. These rules are enforced by the Federal Aviation Administration (FAA).

[15] “14 CFR Part 107 -- Small Unmanned Aircraft Systems,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107> (Accessed: 25 November 2024).

Title 14 CFR Part 48: “Registration and Marking Requirements for Small Unmanned Aircraft”

This standard provides the regulation set that drones must be marked with identifying information. This only applies to drones that weigh 0.55 pounds or less and are only effective during commercial use, not recreational.

[16] “14 CFR Part 48 -- Registration and Marking Requirements for Small Unmanned Aircraft,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-48> (Accessed: 25 November 2024).

‌

PEP 8 – Style Guide for Python Code

This standard is part of the Python Enhancement Proposals. It lists the guidelines required to write code in a readable manner using consistent styling. Python is the programming language that is used in our product.

[19] G. van Rossum, B. Warsaw, and N. Coghlan, “PEP 8 – Style Guide for Python Code,” *peps.python.org*, Jul. 05, 2001. <https://peps.python.org/pep-0008/> (Accessed: 25 November 2024).

**X. Summary**

The project for Seven Hills aims to design a hands-free control interface that will enable users with a different level of physical disability, such as quadriplegia, to interact with their environment for photo capture using the DJI Tello drone. The goal is to design an adaptive controller that is intuitive and accessible, especially to users who cannot use traditional remote controllers. This report discusses the problem, research, and objectives that were completed in order to fulfill the requirements for this project. First, the Face Detection Controller uses face expressions to control the drone, while secondly, the MediaPipe Controller uses the Google MediaPipe framework, which can detect high-precision expressions. Both solutions make use of the Tello SDK over a Wi-Fi UDP connection and introduce considerations about both reliability and range. Comparing these alternatives, Face Detection Controller was selected because it requires less computational power and thus would be more accessible on different devices, especially those that have lower processing power. Further steps include the optimization of the face detection controller for real-time responsiveness, more accurate expression detection, and device compatibility. This will create a firm and easily accessible interface to assist in creating a seamless, hands-free user experience for persons with any mobility impairments, whether in the operation of the drone or taking of a photo. The next step is our aim to deliver a completed working project to Seven Hills by May 2025.

**REFRENCES**

[1] Recfaces, “Understanding Facial Recognition Algorithms,” *RecFaces*, Mar. 25, 2021. https://recfaces.com/articles/facial-recognition-algorithms

[2] S. Xu, J. Wang, W. Shou, T. Ngo, A.-M. Sadick, and X. Wang, “Computer Vision Techniques in Construction: A Critical Review,” *Archives of Computational Methods in Engineering*, vol. 28, no. 5, pp. 3383–3397, Oct. 2020, doi: https://doi.org/10.1007/s11831-020-09504-3.

[3] A. Bhattacharyya, S. Chatterjee, S. Sen, A. Sinitca, D. Kaplun, and R. Sarkar, “A deep learning model for classifying human facial expressions from infrared thermal images,” *Scientific Reports*, vol. 11, no. 1, Oct. 2021, doi: https://doi.org/10.1038/s41598-021-99998-z.

[4] Computer vision engineer, “Emotion detection with Python and OpenCV | Computer vision tutorial,” *YouTube*, Sep. 14, 2022. https://www.youtube.com/watch?v=Vq\_01gFG2vk&list=TLGGt9PGnW-ostwxNTExMjAyNA (accessed Nov. 15, 2024).

[5] “Home Page - PROVEN Robotics,” Mar. 02, 2022. https://provenrobotics.ai/

[6] T. Altameem and A. Altameem, “Facial expression recognition using human machine interaction and multi-modal visualization analysis for healthcare applications,” *Image and Vision Computing*, vol. 103, p. 104044, Nov. 2020, doi: https://doi.org/10.1016/j.imavis.2020.104044.

[7] “Facial landmarks with dlib, OpenCV, and Python,” *PyImageSearch*, Apr. 03, 2017. https://pyimagesearch.com/2017/04/03/facial-landmarks-dlib-opencv-python/

[8] G. Boesch, “MediaPipe: Google’s Open Source Framework for ML solutions (2023 guide),” *viso.ai*, Jan. 02, 2023. https://viso.ai/computer-vision/mediapipe/

[9] manish-9245, “Facial-Emotion-Recognition-using-OpenCV-and-Deepface/emotion.py at main · manish-9245/Facial-Emotion-Recognition-using-OpenCV-and-Deepface,” *GitHub*, 2023. https://github.com/manish-9245/Facial-Emotion-Recognition-using-OpenCV-and-Deepface/blob/main/emotion.py

[10] “Object Detection Using Haar Cascade: OpenCV,” *Analytics Vidhya*, Apr. 01, 2022. https://www.analyticsvidhya.com/blog/2022/04/object-detection-using-haar-cascade-opencv/

[11] Drone Mesh (Drone Reviews), “Ultimate Flight Controller Design -- Master Class [EP1] // INTRO,” *YouTube*, May 23, 2020. https://www.youtube.com/watch?v=asWNNfc5AKg&list=TLGG-gblAgFzvPYxNTExMjAyNA (accessed Nov. 15, 2024).

[12] G. Ghazi and J. Voyer, “Use of a DJI Tello Drone as an Educational Platform in the Field of Control Engineering,” *Proceedings of the Canadian Engineering Education Association (CEEA)*, Mar. 2024, doi: https://doi.org/10.24908/pceea.2023.17061.

[13] CloudFlare, “What is UDP? | Cloudflare,” *Cloudflare*, 2022. Available: https://www.cloudflare.com/learning/ddos/glossary/user-datagram-protocol-udp/

[14] “SDK 2.0 User Guide 2,” 2018.

[15] “14 CFR Part 107 -- Small Unmanned Aircraft Systems,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-107> (Accessed: 25 November 2024).

[16] “14 CFR Part 48 -- Registration and Marking Requirements for Small Unmanned Aircraft,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-C/part-48> (Accessed: 25 November 2024).

[17] “47 CFR Part 15 -- Radio Frequency Devices,” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15> (Accessed: 25 November 2024).

[18] “14 CFR Part 91 -- General Operating and Flight Rules” *www.ecfr.gov*. <https://www.ecfr.gov/current/title-14/chapter-I/subchapter-F/part-91> (Accessed: 25 November 2024).

[19] G. van Rossum, B. Warsaw, and N. Coghlan, “PEP 8 – Style Guide for Python Code,” *peps.python.org*, Jul. 05, 2001. <https://peps.python.org/pep-0008/> (Accessed: 25 November 2024).