AI In The Wild Group Project

Tello Drone

March 2025

Michael Cullen C00261635 & Aaron Doyle C00272515

Introduction

This group project aims to implement some intelligent functionality to our drone and its movement. We are using a DJI Tello drone for our implementation of this task. This drone is coded in Python and that is how we control the drone and train it for its intelligent functionality.

This document aims to outline our findings, learnings and processes that we learned throughout completing this project.

Our Process

Initial Steps

The first thing that we wanted to do was to figure out how to fly the drone manually, we wanted to do this because understanding its movements and how it flies would give us a better understanding of how to code it later on. We downloaded the Tello drone mobile application and this acts as a controller for the drone with the directional buttons to fly it, once we connected it over wifi we were then able to fly the drone.

Our next step after this was to try and get the drone to fly using some code. For this, we decided to read through the DJI Tello documentation that can be found here. (https://djitellopy.readthedocs.io/en/latest/tello/) . We initially ran into some trouble getting the commands to run in our code. We were trying to get the drone to fly forward, turn around, fly back to us and then land. Initially, we were running into issues where it would just take off and land straight back down, but after coding some commands to make the drone sleep in the code so it had enough time to register each command, we were able to get this executing as expected.

Gesture Recognition

Our next step for this project was to get the gesture recognition that was given to us in the project specification working. This functionality uses the camera on the drone to recognise gestures that are made with your right hand. So, for example, if the drone detects that the person in the camera is gesturing upwards, the drone will fly up, and if it gives the ok gesture, the drone will land. We cloned down this repo and tried to get it working, but we ran into some initial issues as the drone wouldn't take off and would throw an exception in the code. This was because the code was getting the drone to fly down straight away, but we were starting it from the floor, so once we removed this, it started working. We then got the gesture recognition through the camera on the drone working as expected as we got it to fly up, down, left, right and land all through using our hands.

Our Plan

Our plan for our functionality for the drone is to get it to detect letters, and then, based on what it detects, it can fly in the path of the letter that it detected. For example, if the letter 'O is detected, it will fly in a circle, and if an 'L' is detected, it will fly down and then right. We want to use a machine learning model to recognise the letter that the drone sees through its camera.

Flight

Our initial concern was if the drone can move diagonally or in a curve to make certain letters possible, for example 'O' as mentioned. After some research through the Tello docs, we stumbled across a function. "send_rc_control()" that allowed us to control the bot's movement in all directions. So after discovering we can move diagonally our next step was to start thinking about moving in a curved motion, we could break it down into small parts. We decided that it would be best to break it into quarters with three movements in each one, so it is a lot of small movements so overall it imitates a circular motion.

We ran into an issue with this, however as the drone begins to struggle when it receives too many commands too quickly. It can end up getting stuck on a command, and this caused us a big issue. We were testing our movement with a circular motion, and it got stuck on going up and right, and because it couldn't reach the command to tell it to come back down or stop, it hit the ceiling. This made us realise that we need to implement more safety features, such as a 'kill switch' to make the drone land straight away.

First Model

After some research into potential datasets, the one we went with was called “Chars74k”, which can be found here (https://info-ee.surrey.ac.uk/CVSSP/demos/chars74k/). After using it to train a CNN model using tensor flow, we had undesirable results and letters we were getting misclassifications. There was a lot of pre-processing in this as this dataset contained fonts, handwritten and real world letter images. It also contained numbers which we didn’t need as we thought it would throw our model's accuracy off. We combined the datasets and filtered them for just the datasets that we needed. We trained the model and were getting very good accuracy ratings in the mid-nineties however, when we went to test it using the drone's camera and live images of letters, it seemed wildly inaccurate, so we wanted to explore other options for building the model.

Second Model

We then decided to attempt to use a website from a previous lab where we built a Tiny Sorter. We wanted to train the model on our images, so we remembered that that was possible by using Teachable Machine, which can be found here (https://teachablemachine.withgoogle.com/train). It works by holding up a notebook with a hand-drawn letter and taking roughly three hundred photos of ten random letters along with three hundred photos of a background with no notebook. We trained ten letters to save time rather than using all twenty-six. This ended up working quite well as it was custom training data that we would be using in the project.

Model Implementation

Our next step for this project was to try and implement the model into the drone, We had some help from ChatGPT as we were struggling to connect the model to the Tello drone and restructure the image size so that the drone's camera would match the image format for the model. We got this to connect to the camera and then overlay the prediction with the confidence rating on the screen over the camera.

Once we were confident that the model was being executed as expected on the drone, our next step was to implement the model whilst it was flying and link it to the movements based on its predictions. We used the logic that we got in the last step to load the model and connect to the Tello drone camera stream. There was initially an issue of the model being extremely delayed on the live feed, but we figured out this was because we were storing every frame that the camera was getting, which was slowing the process down massively. Once we stopped storing them, this issue was immediately fixed. We did also run into some version issues with TensorFlow but these were resolved quite quickly and didn't pose too much of an issue as we were using a python virtual environment. We implemented keyboard controls as an extra safety measure so everything is user initiated. For example, these are the keys we implemented:

- T was used for the takeoff of the drone
- M was used to activate the model and start predicting
- Q was to quit the model and program
- Space was an emergency landing button just in case anything did go wrong
- L was also implemented as a normal land button for the drone

We wanted to implement it like this as we felt it was a safer way to handle the drone in case anything did go wrong whilst running the model and movements.

Movement Logic

The next and final step for us was to now implement the movement logic based off of the predictions. Once the model detected the letter with 70% confidence or above for 3 consecutive seconds it would then execute the movement of the drone. We did this as it does tend to fluctuate whilst detecting letters so we wanted to make sure it was absolutely sure of the letter before it started to carry out the movement logic for the letter.

One thing we were working on and had been working to a good extent was getting the drone to fly in curved shapes. We were still tweaking this but didn't get a chance to implement it as we needed to focus more on implementing the letter flight logic as this was a primary focus for the functionality of the drone.

We unfortunately didn't get time to implement the logic for all 10 letters, we would have been able without time restraints but unfortunately we only got the time to implement the following letters: X, T, E, S, Z and L.

We were happy to get X and Z working as getting the diagonal movements was quite tricky but we did manage to get it moving smoothly in the end.

Results & Conclusion

In conclusion, we managed to implement a machine learning model with the drone that when activated is able to detect hand drawn letters on paper and once it is confident of the letters it will then execute a flight path of the letter that it just detected.

Overall, this was a successful project, as the logic and model all worked as expected. If we had a little bit more time we would have liked to smooth out the curved flight paths and implement more letters for the model to detect and carry out. We were very focused on the first model and tried to get that working initially, but once we used the Teachable Machine model, we got a lot better results.

Some of the key achievements that we managed to obtain throughout this include finding the best model to use for our machine learning, executing smooth diagonal movements for the drone, implementing safety measures for the drone, integrating the model with the Tello camera and executing flight paths based off of the predictions that the model was making.

This project helped us with time management, collaboration, prioritisation of tasks and integrating software with hardware. We are overall very happy with the logic that we managed to implement to the drone.