Gesture Control Of Drone Using A Leap Motion Controller

Navjot Sandhu | Ruthvik Penumatcha | Ajithan Urutheran

Ryerson University

Objective

The leap motion controller is a small device that tracks hand and finger movements using infrared LEDs. This allows the user to input gesture commands in the place of sophisticated keyboard and mouse inputs. The objective of the project is to create two different solutions: Non-AI and AI Leap Motion Controller in order to control the Parrot Mambo Drone. Improvements were made to the original solution by using an accessible webcam as an input controller rather than the Leap Motion Controller. The Non-AI solution will be created using the Google MediaPipe API and the AI solution will utilize a gesture-recognition CNN Model.

Motivation

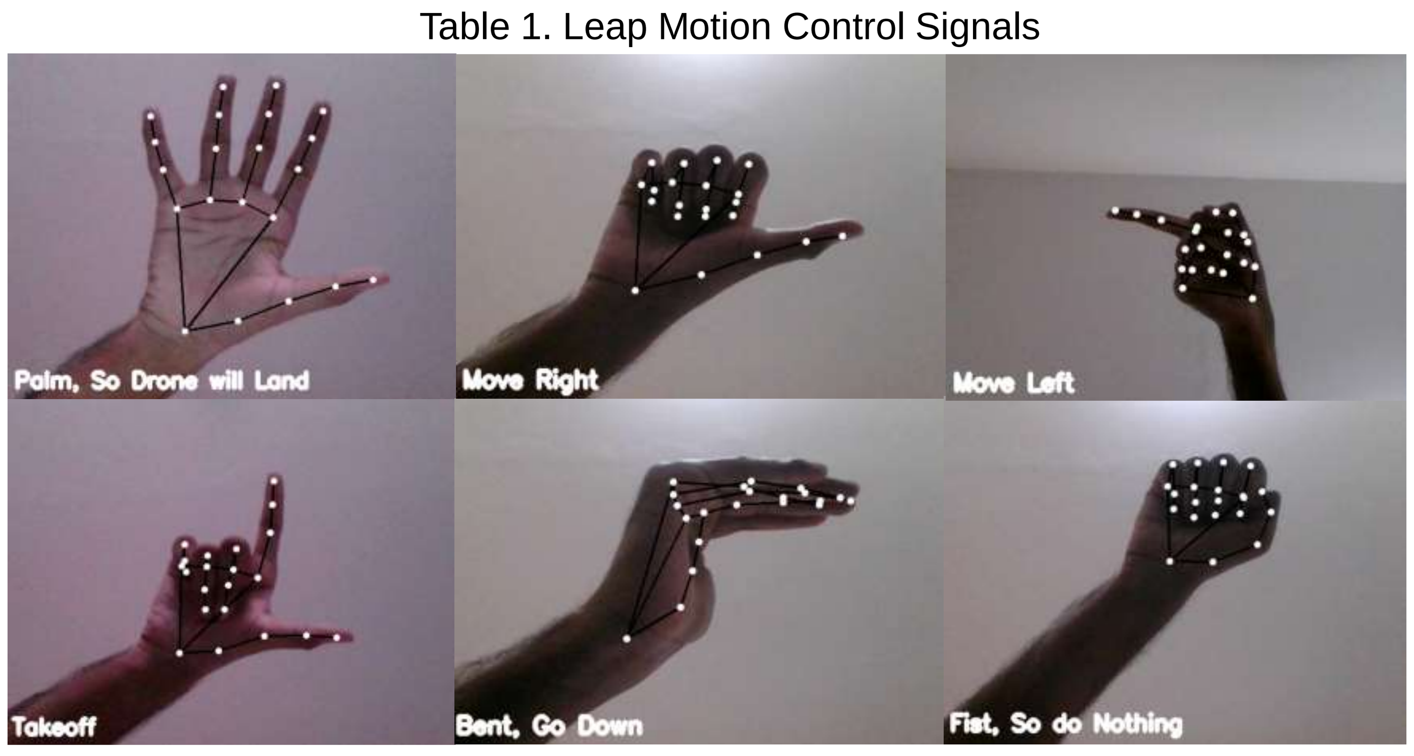
Drones are becoming increasingly prevalent in today's world. There has been a spike in drone industry growth with several new use cases in many industries. Some of these industries include Insurance, Construction, Agriculture and Emergency Medical Services; each of which have their own commercial drone applications [1]. The issue is the severe learning gap between a professional drone pilot and the average layman. There is extensive training required to control a drone using a hand-held controller. Rather than have companies undergo these expenses for training, the complex drone functionalities and maneuvers can be mapped to simple hand gestures. An example of the significant use cases of a drone include delivering food, medical services, and relief packages to remote communities. In the case that traditional delivery methods (road transportation) are blocked due to landslides, drones can mitigate these obstacles.

Approach

Initially, research was completed on forming a connection between the Leap Motion Controller (LMC) and the Parrot Mambo Drone (PMD). There were python incompatibility issues between the LMC and PMD software development kits. They were overcome through creating a client-server pipe. The client was hosted on the LMC and formatted the user's hand directional inputs to drone movements signals. The server on the PMD read and executed the drone movement signals. These issues became exacerbated when implementing an AI solution. Instead of the LMC, a webcam was utilized to capture hand-image inputs and trained a multi-classification CNN model to recognize gestures. These gestures would be classified and send drone movement signals to the PMD.

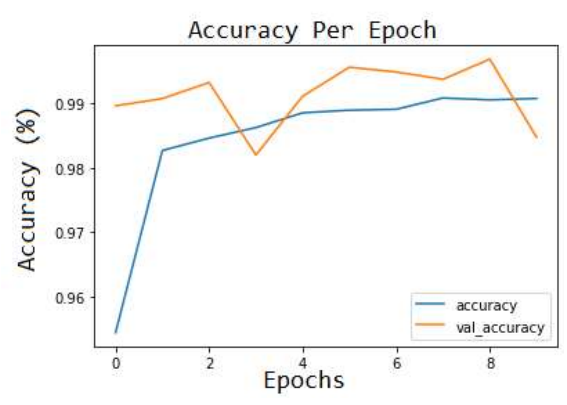
Gesture	Control Signal Flag	Data
Takeoff	1	1 Takeoff
Direction Vector	2	2 {roll} {yaw} {pitch} {0}
Landing	3	3 Landing
Stationary	4	4 0000

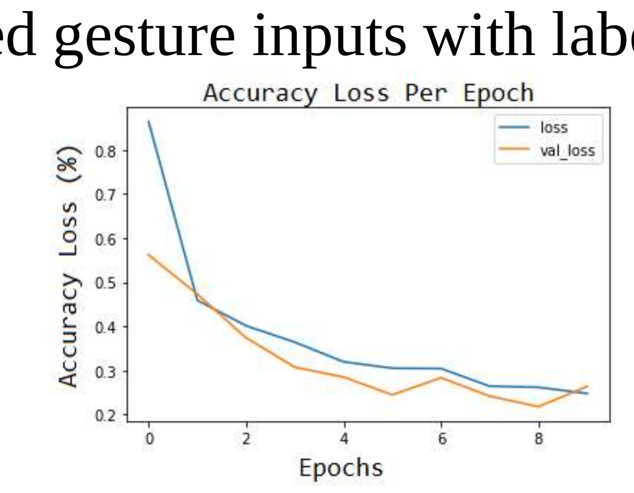
Table 1. Leap Motion Control Signals



Testing

Testing for the drone was completed by determining a list of gestures to implement and creating a large dataset containing these gestures in a split of 5000:1000 for training and validation, respectively. Once this data was obtained, a python file was created that imports the CNN model and identifies unclassified gesture inputs with labels.



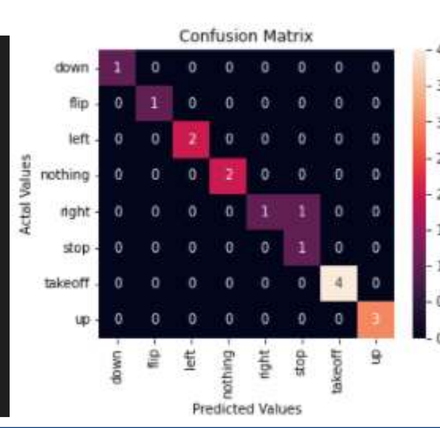


A wireless network connection was created using the 802.11ac standard to send control signals to the drone. Gesture labels were obtained by inputting the test gesture images into the CNN model. The gesture labels were sent as control signals after 20 assertions to the PMD.

```
Found 40000 images belonging to 8 classes.
Found 8000 images belonging to 8 classes.
Epoch 1/10
2500/2500 [==============================] - 342s 137ms/step - loss: 0.8644 - accuracy: 0.9545 - val_loss: 0.5621 - val_accurac
y: 0.9896
Epoch 2/10
2500/2500 [==============================] - 273s 109ms/step - loss: 0.4590 - accuracy: 0.9827 - val_loss: 0.4729 - val_accurac
y: 0.9908
Epoch 3/10
2500/2500 [==============================] - 276s 110ms/step - loss: 0.4014 - accuracy: 0.9846 - val_loss: 0.3739 - val_accurac
y: 0.9933
Epoch 4/10
2500/2500 [==============================] - 280s 112ms/step - loss: 0.3639 - accuracy: 0.9862 - val_loss: 0.3073 - val_accurac
y: 0.9820
Epoch 5/10
2500/2500 [==============================] - 278s 111ms/step - loss: 0.3195 - accuracy: 0.9886 - val_loss: 0.2850 - val_accurac
y: 0.9911
Epoch 6/10
2500/2500 [==============================] - 274s 110ms/step - loss: 0.3050 - accuracy: 0.9890 - val_loss: 0.2444 - val_accurac
y: 0.9956
Epoch 7/10
2500/2500 [==============================] - 275s 110ms/step - loss: 0.3041 - accuracy: 0.9891 - val_loss: 0.2832 - val_accurac
y: 0.9949
Epoch 8/10
2500/2500 [==============================] - 275s 110ms/step - loss: 0.2642 - accuracy: 0.9908 - val_loss: 0.2417 - val_accurac
y: 0.9937
Epoch 9/10
2500/2500 [==============================] - 279s 112ms/step - loss: 0.2616 - accuracy: 0.9905 - val_loss: 0.2178 - val_accurac
y: 0.9969
Epoch 10/10
2500/2500 [==============================] - 281s 113ms/step - loss: 0.2476 - accuracy: 0.9908 - val_loss: 0.2637 - val_accurac
y: 0.9847
```

AI CNN Controller

	precision	recall	f1-score	support
down	1.00	1.00	1.00	1
flip	1.00	1.00	1.00	1
left	1.00	1.00	1.00	2
nothing	1.00	1.00	1.00	2
right	1.00	0.50	0.67	2
stop	0.50	1.00	0.67	1
takeoff	1.00	1.00	1.00	4
up	1.00	1.00	1.00	3
accuracy			0.94	16
macro avg	0.94	0.94	0.92	16
weighted avg	0.97	0.94	0.94	16



Non-AI Leap Controller

Conclusion

Ultimately, success was achieved implementing an AI and Non-AI model to control the Parrot Mambo Drone. Gestures were recognized at an accuracy rate of nearly 99% in the multi-classification CNN model. While the Non-AI Leap Controller succeeded in controlling the movement of the drone, the AI CNN model was able to complete advanced drone maneuvers such as barrel rolls and flips. There was an increase in accuracy using the CNN model at the expense of greater latency as compared to the Non-AI model.

References and Acknowledgements

We would like to thank Dr. Cungang (Truman) Yang for guiding and supporting us throughout the capstone project. His contributions through research notes, physical technology and ideas that were brought to our group were priceless.

[1]https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts