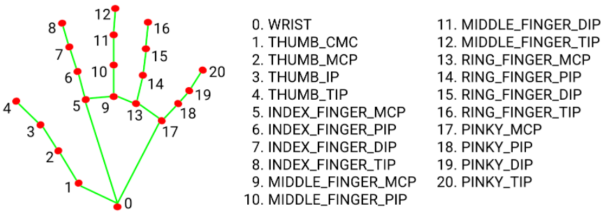
Controlling Drone with Hand Gestures

The ability to perceive the shape and motion of hands can be a vital component in improving the user experience across a variety of technological domains and platforms, and can form the basis for sign language understanding and hand gesture control.

In this assignment, we try to control a drone using our hand gestures.

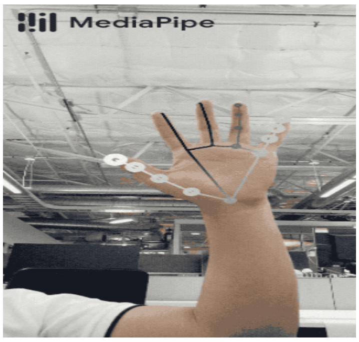
MediaPipe Hands has been implemented to detect 21 3D landmarks.
2 stage detection:
A palm detection - returns an oriented hand bounding box.
A hand landmark model - returns high-fidelity 3D hand key points



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

1) **Let's start with implementing media pipes hand key point detection model using webcam or static image.**

   https://google.github.io/mediapipe/solutions/hands.html



For each frame processed we have an array of 3d coordinates of 21 key points. We use these key-points for gesture recognition

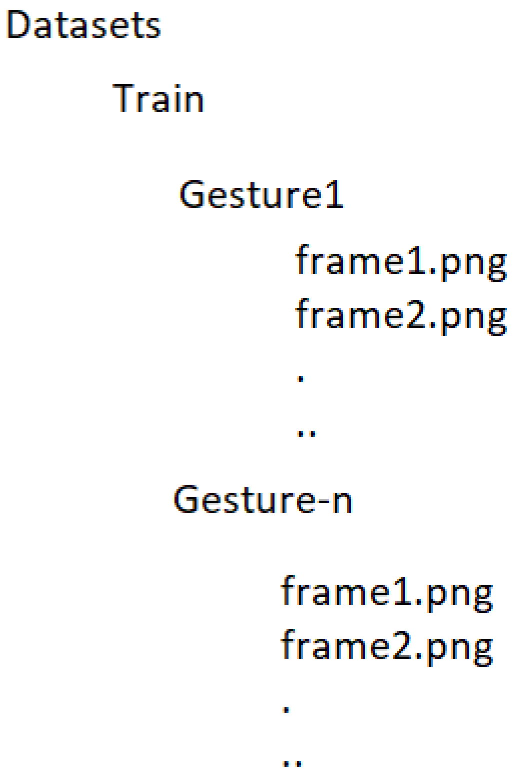
To just use pretrained model refer to ***gesture_test.py***

You can use the pretrained model for gesture recognition or create your own dataset, or you can just

- The dataset can be created by running hand key point detection for certain time and save different gestures.

2)You can also first save your gesture as video and convert video into frames using the *videos2frame.py* script

It is recommended we save the dataset in the below format for training.



```
Datasets

     Train

          Gesture1

               frame1.png
               frame2.png
               .
               ..

          Gesture-n

               frame1.png
               frame2.png
               .
               ..
```

3)For each frame we run the hand key point detection and append it to the CSV file. You can refer *my_pose_train.py* script to finish the task.

At the end of step3 you should have a CSV with information about key points and its label.

4)Now we are ready with our dataset, we now build a KNN classifier to recognize our gestures.

Train_X = N * [(x0,y0), (x1,y1) …… (x20,y20)]
Train_Y = N*label for each hand gesture

You can refer to *knncvs.py* to train the KNN classifier and save model as pickle file.

By the end of step 4 we should have a pickle file saved in your directory.
5) Let's test the classifier on real time. You can refer *PoseModule.py* to test our classifier out.

You should be able to perform hand key point detection and gesture recognition in real-time.

6)Now all is left to implement drone control using our gestures. You can refer to *drone_pose_control.py*

You can take this further to make the drone do some cool actions or trajectories with your pose.