Teoman Iscen
301 444 346
Tei1@sfu.ca

Milestone 2 Report

Project Recap:

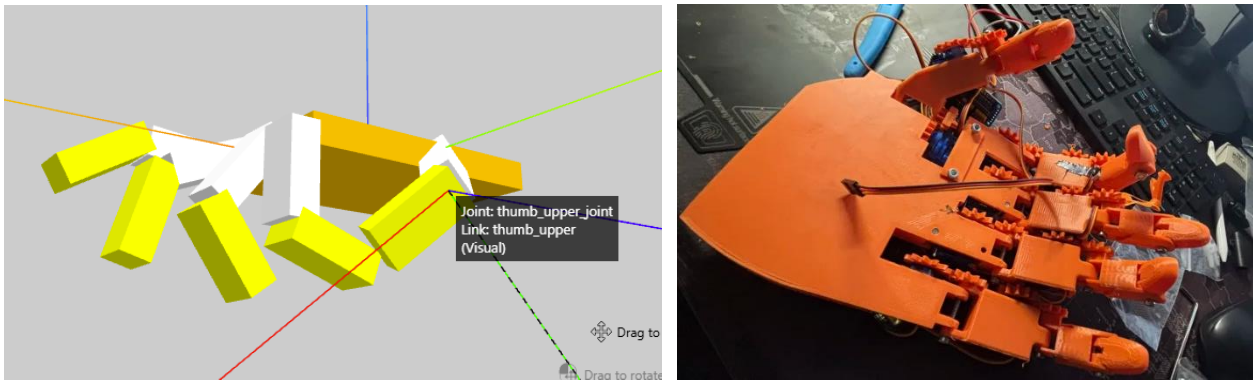
https://github.com/teiscen/Hand-Tracker

My project is about manipulating a virtual hand in real-time. The input is landmark information generated using Mediapipe, fed by video from a webcam. The landmarks will be passed into an autoencoder model that will generate the target joint angles for the virtual hand. The model will be responsible for smoothing jerky motions generated by Mediapipe in real time.

Accomplishments:

Significant accomplishments: complete XACRO model of the robotic hand, methods to generate, label, and display examples, real-time movement capture, and framework is ready to integrate real-time capture and the Rviz simulation once the model is complete.

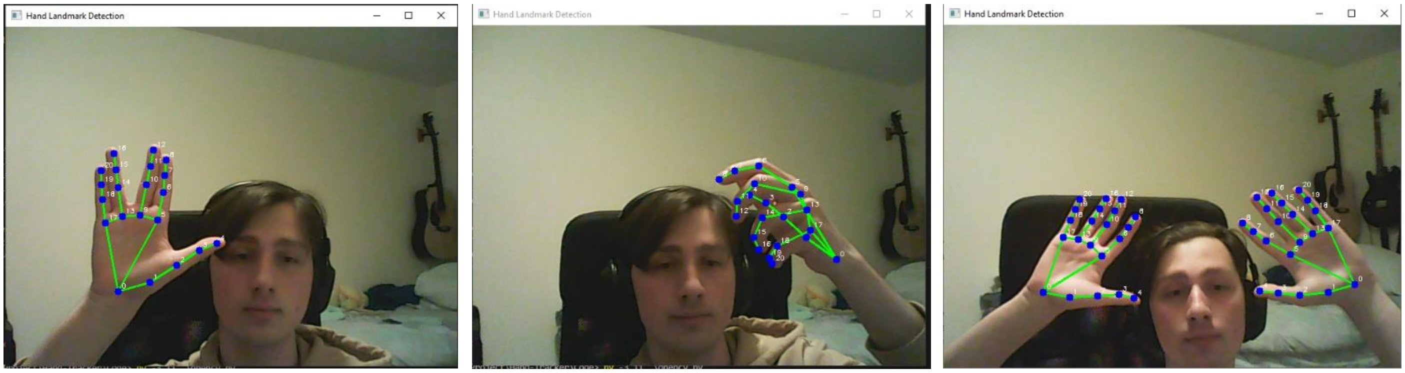
Model of the hand:



The left image represents the hand model generated in XACRO based on the prototype to the right. The joint angles on the virtual model are limited to 180 degrees, and each link, other than the lower thumb, is comprised of the same segments.

The original plan included simulating the hand in Gazebo, but for the demonstrations RViz is an easier alternative. One aspect of the hands that are currently incomplete is the joint state publishers responsible for manipulating the hands in RViz.

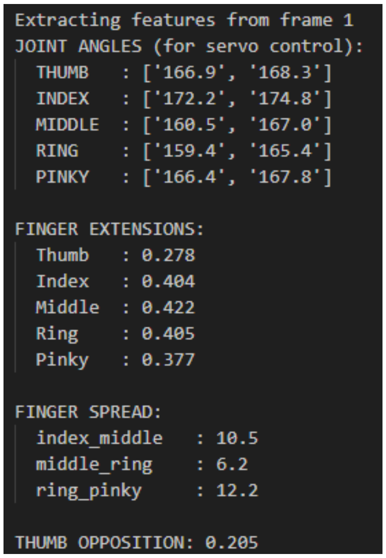
Real Time Hand Detection:



The video is captured by a webcam and processed with OpenCV and Mediapipe HandLandmarker. Once a frame is read in by opencv it is flipped then converted from Blue Green Red to Red Green Blue. Afterwards, the landmark data, generated by hand_landmarker.task from Mediapipe, is drawn to the frame.

Labelled examples are generated by capturing the frames and calling the method *generate_labelled_data(frame_list)* with a list consisting of 5 frames.

Generating Examples:

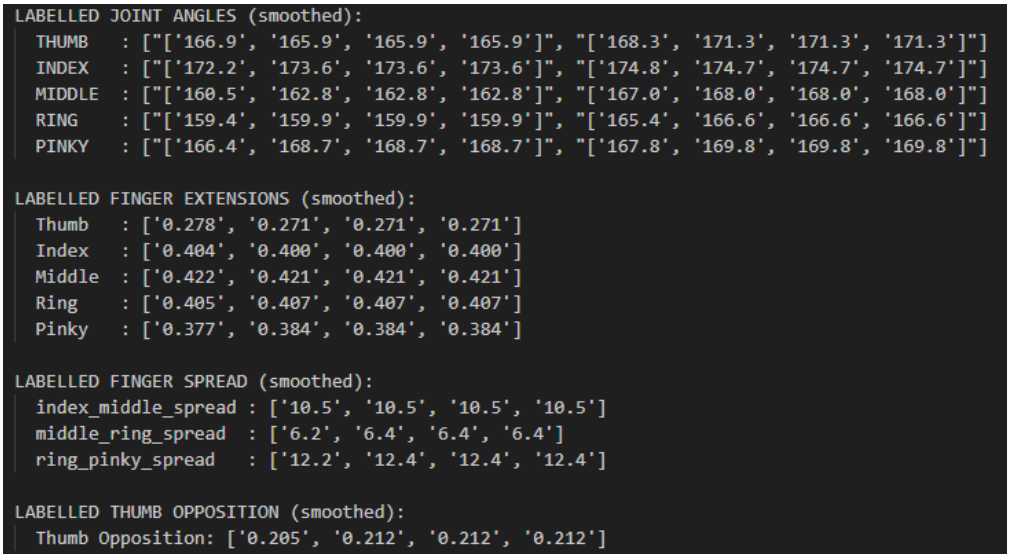


```
Extracting features from frame 1
JOINT ANGLES (for servo control):
  THUMB   : ['166.9', '168.3']
  INDEX   : ['172.2', '174.8']
  MIDDLE  : ['160.5', '167.0']
  RING    : ['159.4', '165.4']
  PINKY   : ['166.4', '167.8']

FINGER EXTENSIONS:
  Thumb   : 0.278
  Index   : 0.404
  Middle  : 0.422
  Ring    : 0.405
  Pinky   : 0.377

FINGER SPREAD:
  index_middle   : 10.5
  middle_ring    : 6.2
  ring_pinky     : 12.2

THUMB OPPOSITION: 0.205
```



```
LABELLED JOINT ANGLES (smoothed):
  THUMB   : ["['166.9', '165.9', '165.9', '165.9']", "['168.3', '171.3', '171.3', '171.3']"]
  INDEX   : ["['172.2', '173.6', '173.6', '173.6']", "['174.8', '174.7', '174.7', '174.7']"]
  MIDDLE  : ["['160.5', '162.8', '162.8', '162.8']", "['167.0', '168.0', '168.0', '168.0']"]
  RING    : ["['159.4', '159.9', '159.9', '159.9']", "['165.4', '166.6', '166.6', '166.6']"]
  PINKY   : ["['166.4', '168.7', '168.7', '168.7']", "['167.8', '169.8', '169.8', '169.8']"]

LABELLED FINGER EXTENSIONS (smoothed):
  Thumb   : ['0.278', '0.271', '0.271', '0.271']
  Index   : ['0.404', '0.400', '0.400', '0.400']
  Middle  : ['0.422', '0.421', '0.421', '0.421']
  Ring    : ['0.405', '0.407', '0.407', '0.407']
  Pinky   : ['0.377', '0.384', '0.384', '0.384']

LABELLED FINGER SPREAD (smoothed):
  index_middle_spread : ['10.5', '10.5', '10.5', '10.5']
  middle_ring_spread  : ['6.2', '6.4', '6.4', '6.4']
  ring_pinky_spread   : ['12.2', '12.4', '12.4', '12.4']

LABELLED THUMB OPPOSITION (smoothed):
  Thumb Opposition: ['0.205', '0.212', '0.212', '0.212']
```

The mediapipe landmarks are provided as normalized coordinates, representing the position of the hand within the video frame. The current plan is to process these normalized coordinated into categories shown to the left, and use these to train the autoencoder.

The examples are labelled with the values of scipy's *savgol_filter()* method, as show in the right image.

Challenges and Roadblocks:

As a result of my first time using many libraries like OpenCV, Mediapipe, Ros2, I have found the hardest challenge is creating a system that makes joining the different aspects of the project together. Once a working model is established, my goal will be to modularize the project as much as possible using what I have learned. This will let me making modifications as necessary and allow me to experiment to get a better idea of what is impacting performance.