Smart Glove Control System

Team 2

- Ata Gün Öğün
- Charles Donven
- Yucong Ma

Sprint 4 Recap

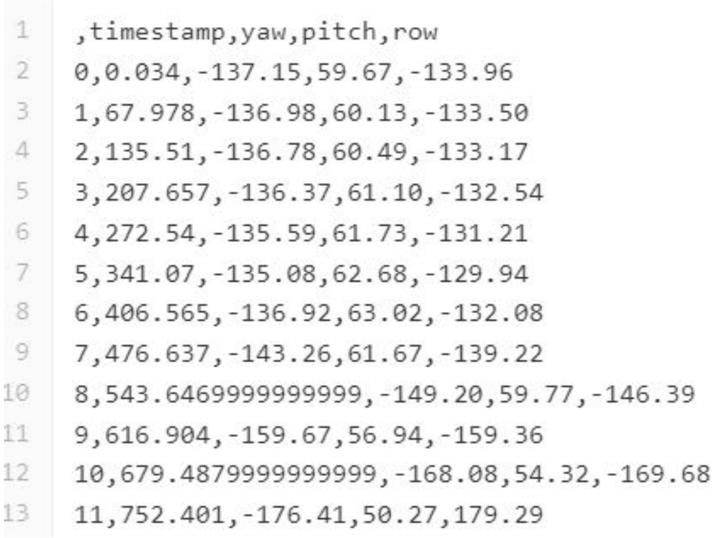
What has been done:

- Implement a web application which simulates an IoT device. We can use it for the demonstration.
- Refactored code structure.
- Recorded new gestures which are more intuitive (swipe left / right).
- Updated program workflow. Machine learning model is integrated into original program and recognizes specific user gesture on the fly.

Machine Learning: Data Model

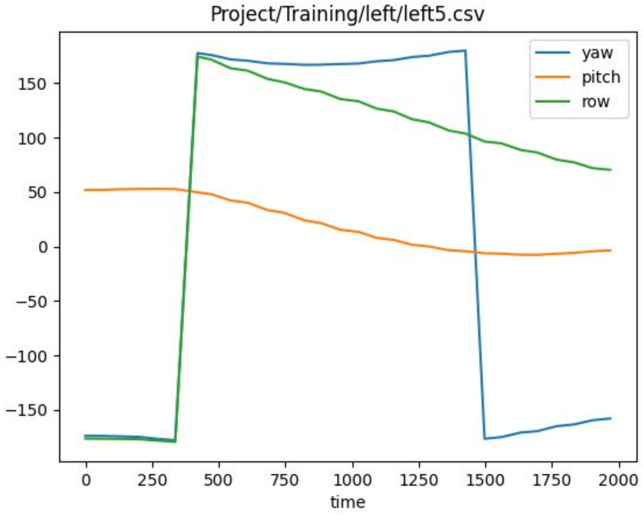
- Flex sensor data are used to specify controlling aspect (device or color)

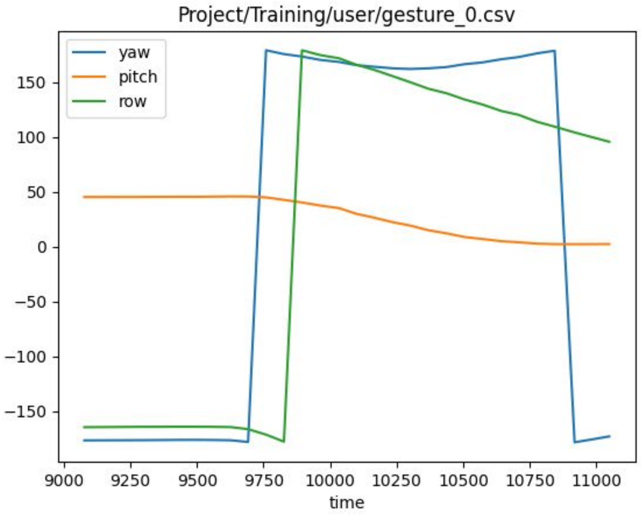
- IMU sensor data (yaw, pitch, row) are used to capture gesture movement



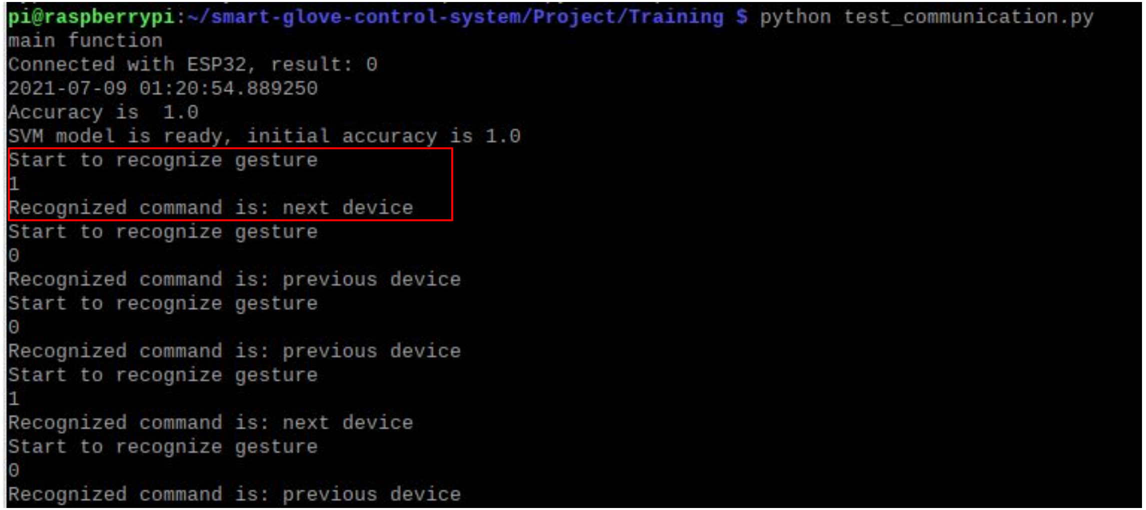
```
,timestamp,yaw,pitch,row
0,0.034,-137.15,59.67,-133.96
1,67.978,-136.98,60.13,-133.50
2,135.51,-136.78,60.49,-133.17
3,207.657,-136.37,61.10,-132.54
4,272.54,-135.59,61.73,-131.21
5,341.07,-135.08,62.68,-129.94
6,406.565,-136.92,63.02,-132.08
7,476.637,-143.26,61.67,-139.22
8,543.6469999999999,-149.20,59.77,-146.39
9,616.904,-159.67,56.94,-159.36
10,679.4879999999999,-168.08,54.32,-169.68
11,752.401,-176.41,50.27,179.29
```

Machine Learning: Swipe Left Gesture



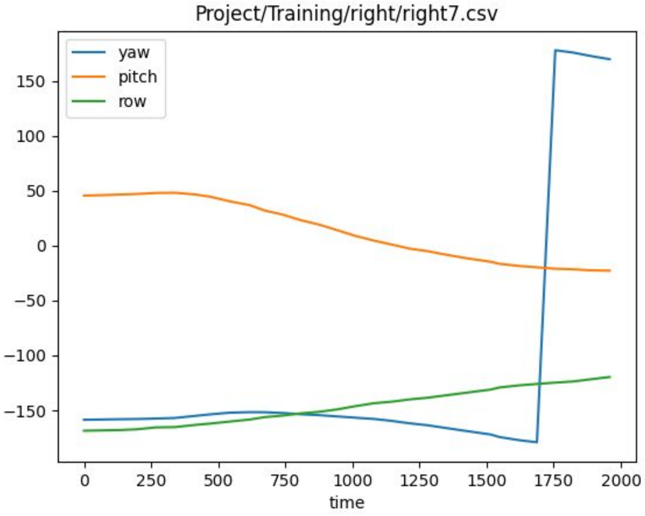


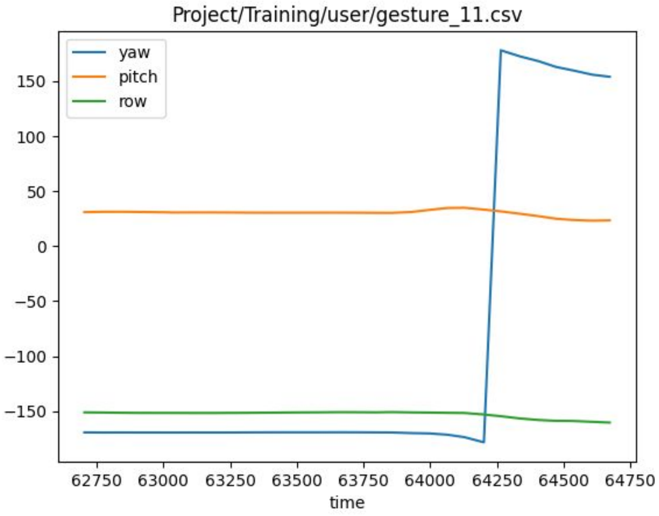
Machine Learning: Swipe Left Gesture



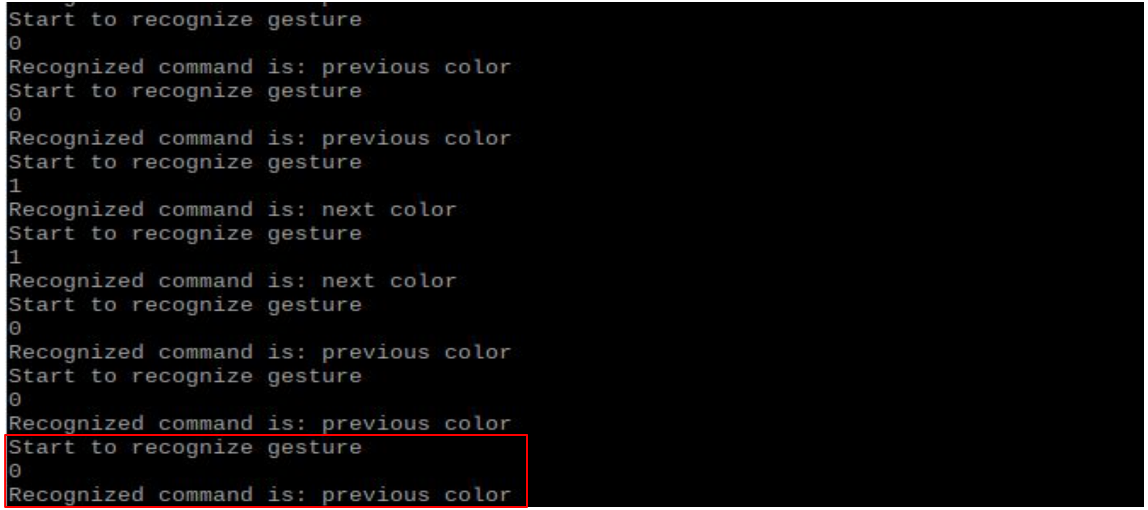
```
pi@raspberrypi:~/smart-glove-control-system/Project/Training $ python test_communication.py
main function
Connected with ESP32, result: 0
2021-07-09 01:20:54.889250
Accuracy is  1.0
SVM model is ready, initial accuracy is 1.0
Start to recognize gesture
1
Recognized command is: next device
Start to recognize gesture
0
Recognized command is: previous device
Start to recognize gesture
0
Recognized command is: previous device
Start to recognize gesture
1
Recognized command is: next device
Start to recognize gesture
0
Recognized command is: previous device
```

Machine Learning: Swipe Right Gesture



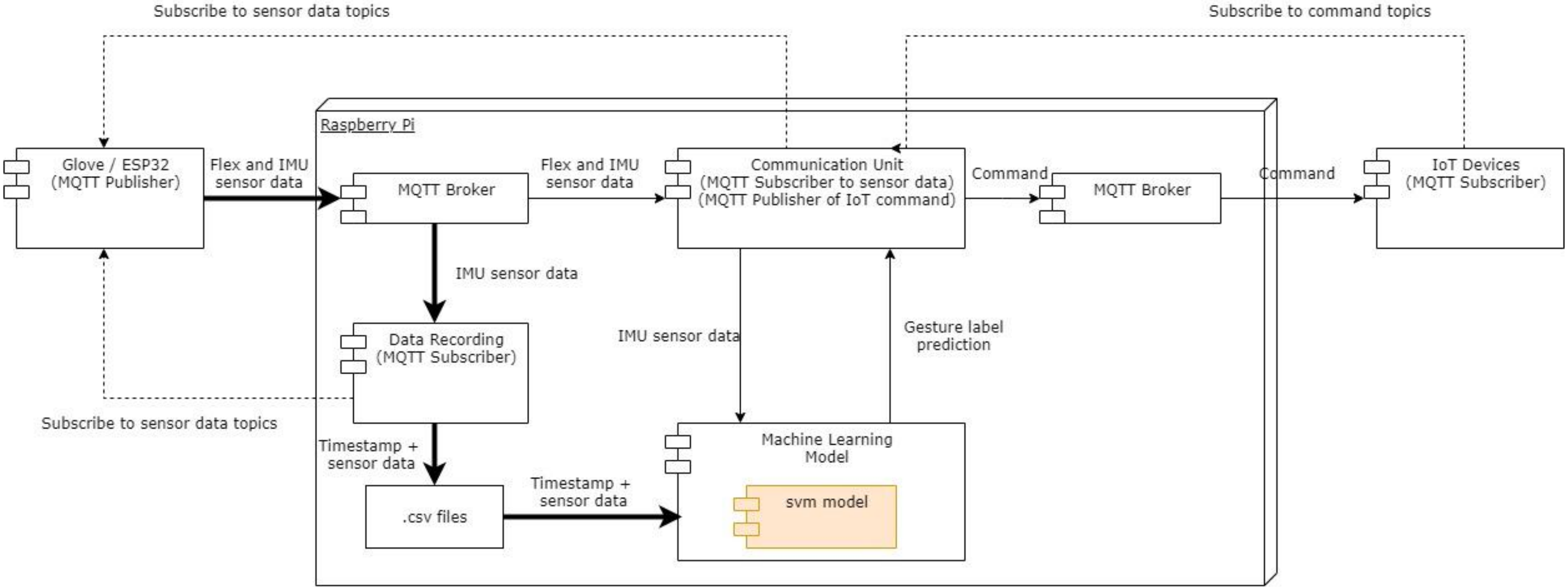


Machine Learning: Swipe Right Gesture

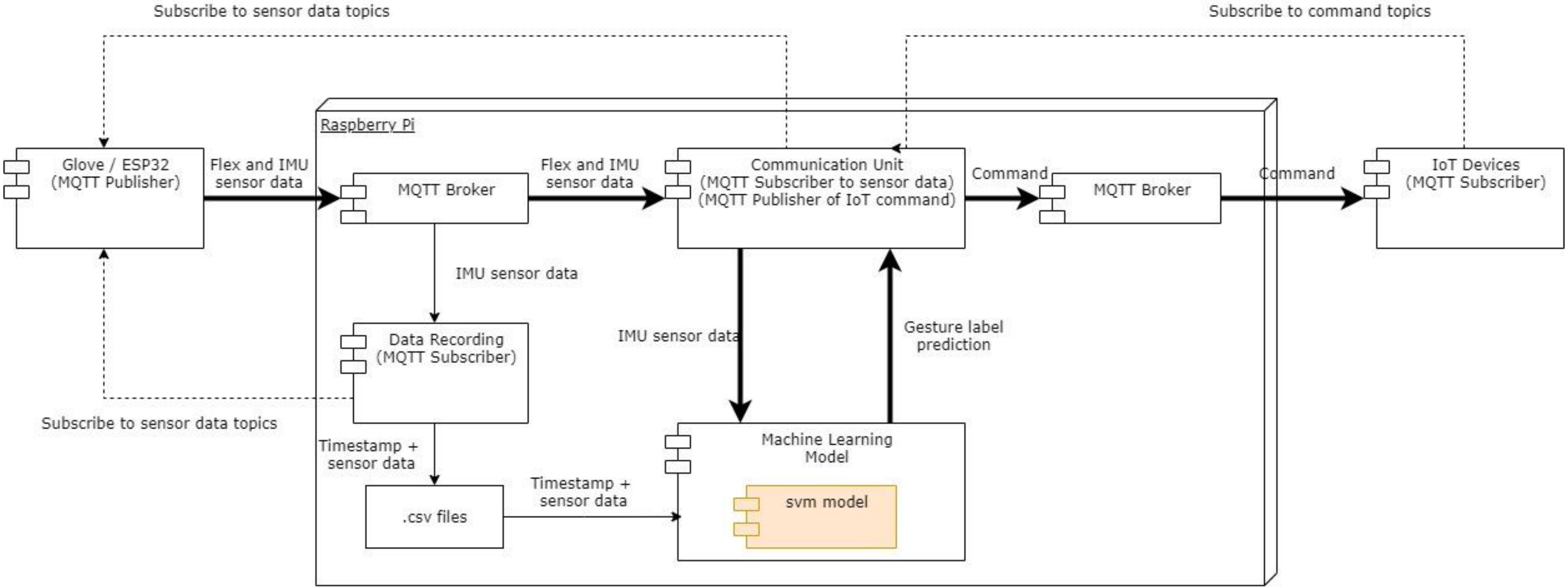


```
Start to recognize gesture
0
Recognized command is: previous color
Start to recognize gesture
0
Recognized command is: previous color
Start to recognize gesture
1
Recognized command is: next color
Start to recognize gesture
1
Recognized command is: next color
Start to recognize gesture
0
Recognized command is: previous color
Start to recognize gesture
0
Recognized command is: previous color
Start to recognize gesture
0
Recognized command is: previous color
```

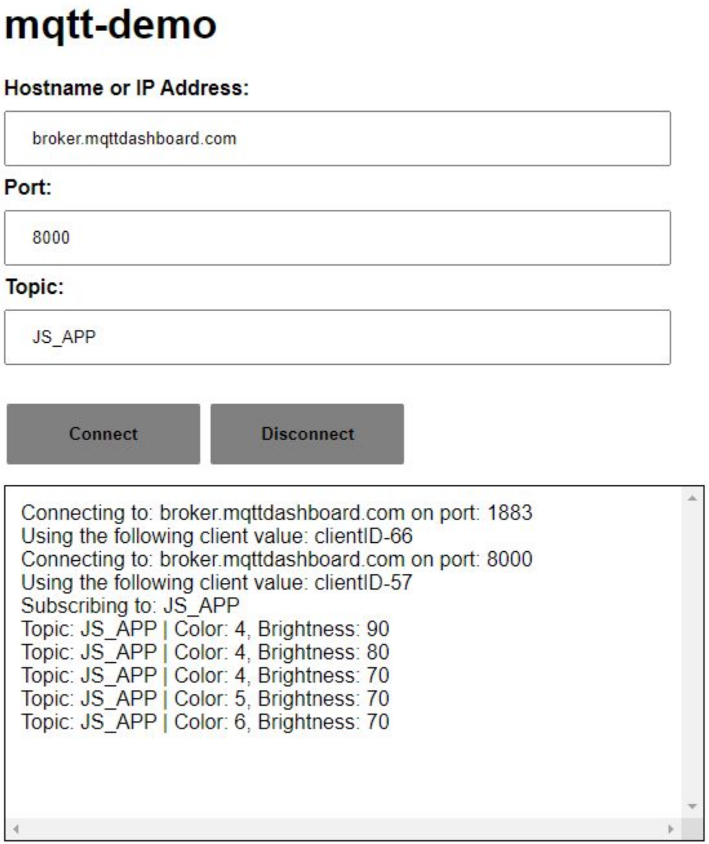
Machine Learning: Model Training


Subscribe to sensor data topics
Subscribe to command topics
Raspberry Pi
Glove / ESP32 (MQTT Publisher)
Flex and IMU sensor data
MQTT Broker
Flex and IMU sensor data
Communication Unit (MQTT Subscriber to sensor data) (MQTT Publisher of IoT command)
Command
MQTT Broker
Command
IoT Devices (MQTT Subscriber)
IMU sensor data
IMU sensor data
Gesture label prediction
Data Recording (MQTT Subscriber)
Subscribe to sensor data topics
Timestamp + sensor data
Timestamp + sensor data
.csv files
Machine Learning Model
svm model

Machine Learning: Model Application



WebApp



mqtt-demo

Hostname or IP Address: broker.mqttdashboard.com

Port: 8000

Topic: JS_APP

```
Connecting to: broker.mqttdashboard.com on port: 1883
Using the following client value: clientID-66
Connecting to: broker.mqttdashboard.com on port: 8000
Using the following client value: clientID-57
Subscribing to: JS_APP
Topic: JS_APP | Color: 4, Brightness: 90
Topic: JS_APP | Color: 4, Brightness: 80
Topic: JS_APP | Color: 4, Brightness: 70
Topic: JS_APP | Color: 5, Brightness: 70
Topic: JS_APP | Color: 6, Brightness: 70
```

Sprint 4 Result

What was missed out:

Given the time constraint, the machine learning data preprocessing (feature extraction) and a model switch from svm polynomial kernel to random forest were left out.

Performance tests were only conducted given correct gestures (left or right swiping). Tests of incorrect gestures were not included.

Next week Plan

- Integrate IoT simulating web app to the program and display visual demonstrations based on commands.

- Test out random forest regression model and compare it with original svm polynomial kernel model.

- Test model prediction given incorrect / random user gestures. Give feedback when random gestures are recognized.

- Prepare for the demo

Thank you!

Questions?