EE3080 | Project ID E047

← Older posts

Week 11 Updates

Edit

Posted on 30 Oct 2021, Sat by PHILIP LEE HANN YUNG

Note: I didn't have time to update the group blogs for the past few weeks. But we have been updating our supervisor through the weekly updates slides throughout this time.

Prediction Results

- The real-time evaluation only performs well if the gesture is at a particular range and time synchronization.
- We discover that there was train data leakage leading to the high accuracy previously.
- We tried varying the occurrence time of the gesture within the frame sample but the prediction results were not good and unusable.
- Hence, two methods were experimented upon to improve our results and reliability.
    - Adding a wait time before the gesture is recorded to be sent to the ML pipeline
    - A voting technique was used to predict k frames before and after the actual frame (total 2k+1 votes)

- The Raspberry 3B+ was not usable as it was too slow and caused too much latency in the gesture prediction. The Raspberry 4B and computer had some differences in accuracy, we are still investigating the reasons.

Project Report

- We subdivided our portion for the project report and began to work on it.

User Interface

- We also start working on a user interface to showcase our project result and use case.

Posted in Phase 2, Phase 3 | Leave a comment

Week 10 Updates

Edit

Posted on 30 Oct 2021, Sat by PHILIP LEE HANN YUNG

Data Collection

- After analysis of our data last week, we decided to collect ‘cleaner’ data samples. We changed the range back to 0.2 – 0.5 m as there were spurious peaks in the 0.5 to 0.6 m range.
- We also changed the gestures again so that we can see their difference distinctly during data analysis. This will help the ML classifier make a sense of the difference of the gestures.

Raspberry Pi

- Managed to run the real-time python code on Raspberry Pi. Debugged and fixed some compatibility issues of our script with a Linux system.
- We find that our script is not computationally expensive and works well running on the 3B+ model which has lower specs.
- We also try to install PyQt5 on the Raspberry Pi so that we may use this for our GUI. The usual pip3 method does not work on the Raspberry Pi as it is a 32-bit ARM architecture.

User Interface

- Pitched up some ideas on the User Interface for DIP presentation, which include: Whiteboard, Menu, Games
- We will commence discussing further details from Week 11 after our ML model is finalized and performing well.

Prediction Results

- We conclude that our dataset is now good as we obtained a 98% accuracy on the test set for the deep learning model using STFT features. Other models also perform around the same accuracy.

Posted in Phase 2 | Leave a comment

Week 9 Updates

Edit

Posted on 30 Oct 2021, Sat by PHILIP LEE HANN YUNG

Data Collection

- Collected more gesture samples from all members due to insufficient data for the machine learning to detect many other different forms of abnormality/ non-ideal case (e.g. direction of swiping, hand size etc.).
- Gathered background data as well for the purpose of better supervision of the learning and not detect any form of gesture when it is not performed.
- After many observations from the GUI, the depth of rolling and pushing gestures is very much similar to one another, hence rolling will be removed and pushing will still be one of our primarily gesture.

Raspberry Pi

- Managed to connect the RPi to Acconeer radar sensor and display it via VNC Viewer.
- RPi3B+ operates using a Static IP address while RPi4 operates via a router

- Installed python package dependencies (e.g., tensorflow, opencv, numpy) using pip3.

Issues faced on Raspberry Pi

- Librosa/numpy cannot be installed in the raspberry pi model 3B+ via pip3.

Solution

- We decided to use Conda which caused a massive destruction and changed the environment path variables.
- We needed to temporarily add the correct system path to obtain sudo command privileges, and go into the bashrc file and modify the path variables to uninstall conda from the Raspberry Pi

Final Use Case

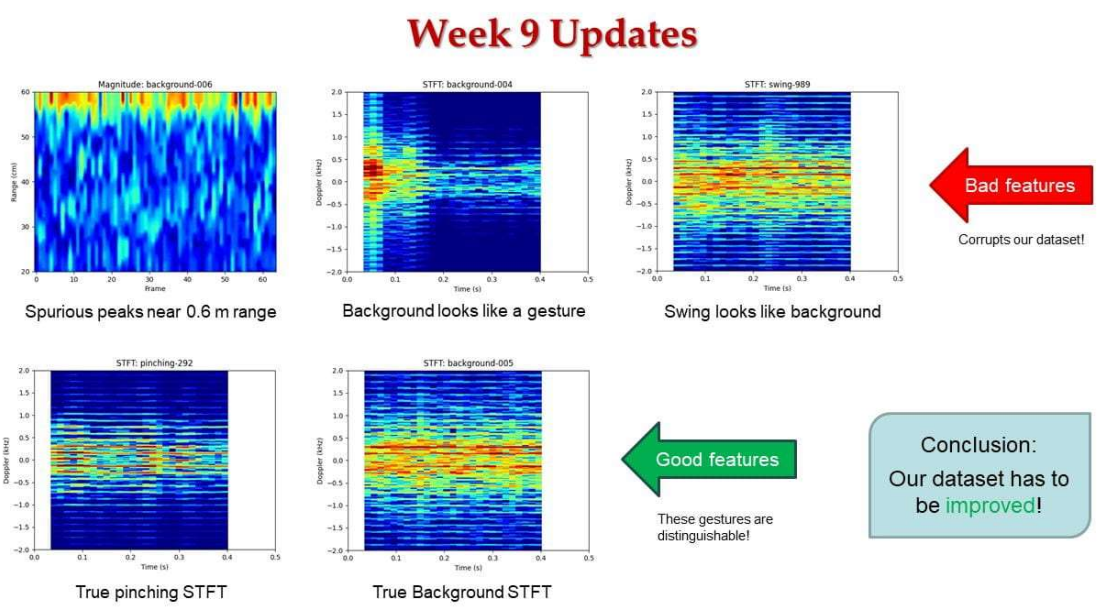
- We also started ideation and briefly discussed on ways to present our project in a usable case scenario. Each member was to research possible use cases.

Prediction

- Experiment with STFT and MFCC parameters.
- Based on the new dataset, we managed to obtain a test accuracy of 80%. Based on this, we tested it real-time too but found disappointing results.
- For each gesture sample, an image file was generated for all of the features (i) magnitude, (ii) STFT, (iii) MFCC.

```
 File "c:\Users\      \Documents\GitHub\smart-touchless-control-radar\final-pipeline\master\preprocess_visualize.py", line 92,
in multiproc_loop
   get_single_stft_plot(index=i, class_labels=class_labels, source_dir=source_dir, X=X, Y=Y)
100.00% :::::::::::::::::::::::::::::::::::::::: |      100 /      100 |:  loop 500-599
100.00% :::::::::::::::::::::::::::::::::::::::: |      100 /      100 |:  loop 600-699
100.00% :::::::::::::::::::::::::::::::::::::::: |      100 /      100 |:  loop 700-799
100.00% :::::::::::::::::::::::::::::::::::::::: |      100 /      100 |:  loop 800-899
100.00% :::::::::::::::::::::::::::::::::::::::: |      100 /      100 |:  loop 900-999
```

- We compared within each gesture class and found that there are many non-distinguishable samples. Hence, we set out a plan to further improve our data collection process



Posted in Phase 2 | Leave a comment

Week 8 Updates

Edit

Posted on 30 Oct 2021, Sat by PHILIP LEE HANN YUNG

Data Collection (1st Round)

- During the process of data collection, We had finalised the range interval to be 0.2m – 0.5m. The range was chosen as such is because it can detect proper peaks
- During the process, we had tried various different parameters namely adjusting the following. (i) Smoothing factor (0.9) (ii) Down-sampling factor (iii) HW acceleration average samples
- One key observation made during the data collection, When the gesture tend to be rapid, the sensor was not able to pick up any peaks, hence we have to slow down our hand gesture in order for the radar to peak up sensible data.
- The GUI was used to extract the gestures (~200 samples total).

Real-Time Predictions

- The real-time script was also finalised and we compared our ML and deep learning models performance.

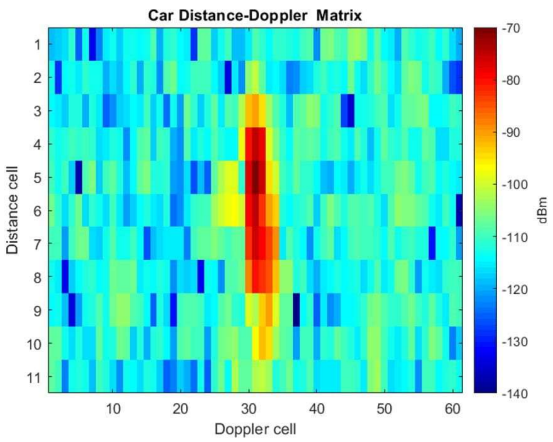
- Able to connect to the Acconeer sensor directly and make predictions of the gestures. The predictions were printed out every 64 frames, based on our training samples.
- The predictions were not reliable so we proceeded to improve based on a data-centric method the following week.

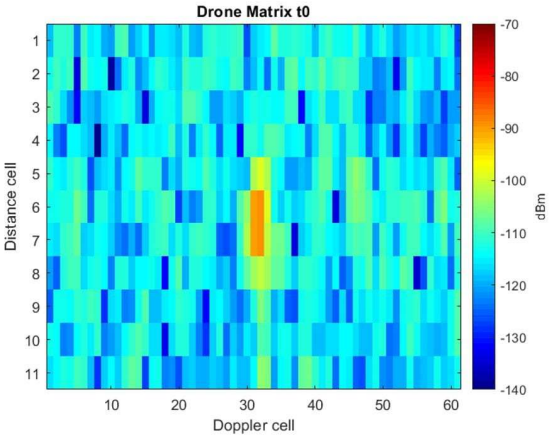
Posted in Phase 2 | Leave a comment

Recess Week Updates

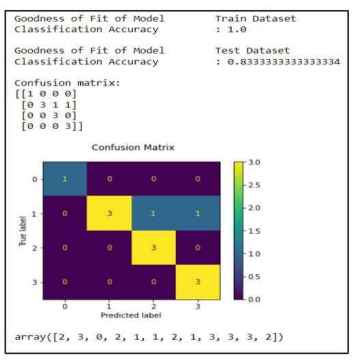
Edit

Posted on 6 Oct 2021, Wed by PHILIP LEE HANN YUNG

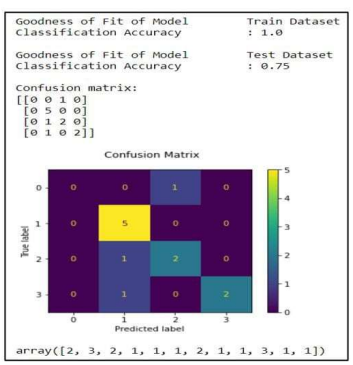




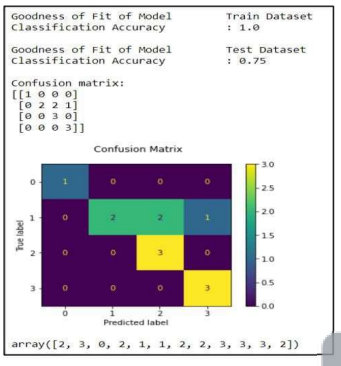
- We used a mock radar dataset of cars/drones/people (https://www.kaggle.com/iroldan/real-doppler-raddar-database) to test our machine learning pipeline when we didn't have enough data.
- The diagrams below show examples of Doppler matrices of the mock dataset to be fed into the models.



(i) LinearSVM model



(ii) Decision Tree model



(iii) Logistic Regression model

- We then implemented the pipeline using our own dataset based on the collected radar data, with a train-test-split ratio of 0.7.
- The dataset size consists of 40 magnitude plots of four gestures (10 plots each) (i) pushing, (ii) rolling forward, (iii) pinching, and (iv) swipe left to run some machine learning models such as KNN, linearSVM, logistic regression etc to find out the best model performance.
- We found that linear SVM, decision tree and logistic regression model can show good classification results on our radar dataset. Improvement will be done in the subsequent meeting.

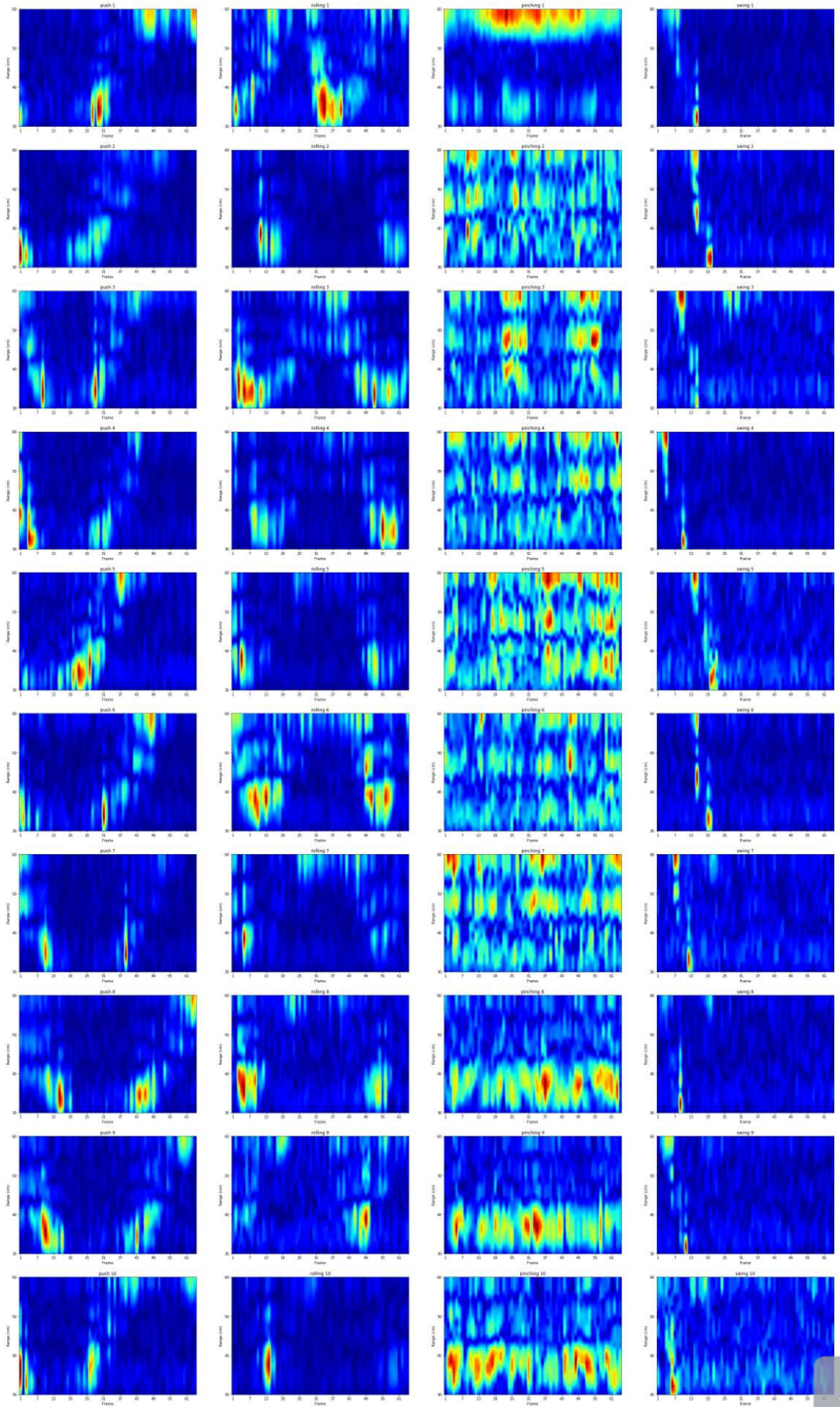
Posted in Phase 2 | Leave a comment

Week 7 Updates

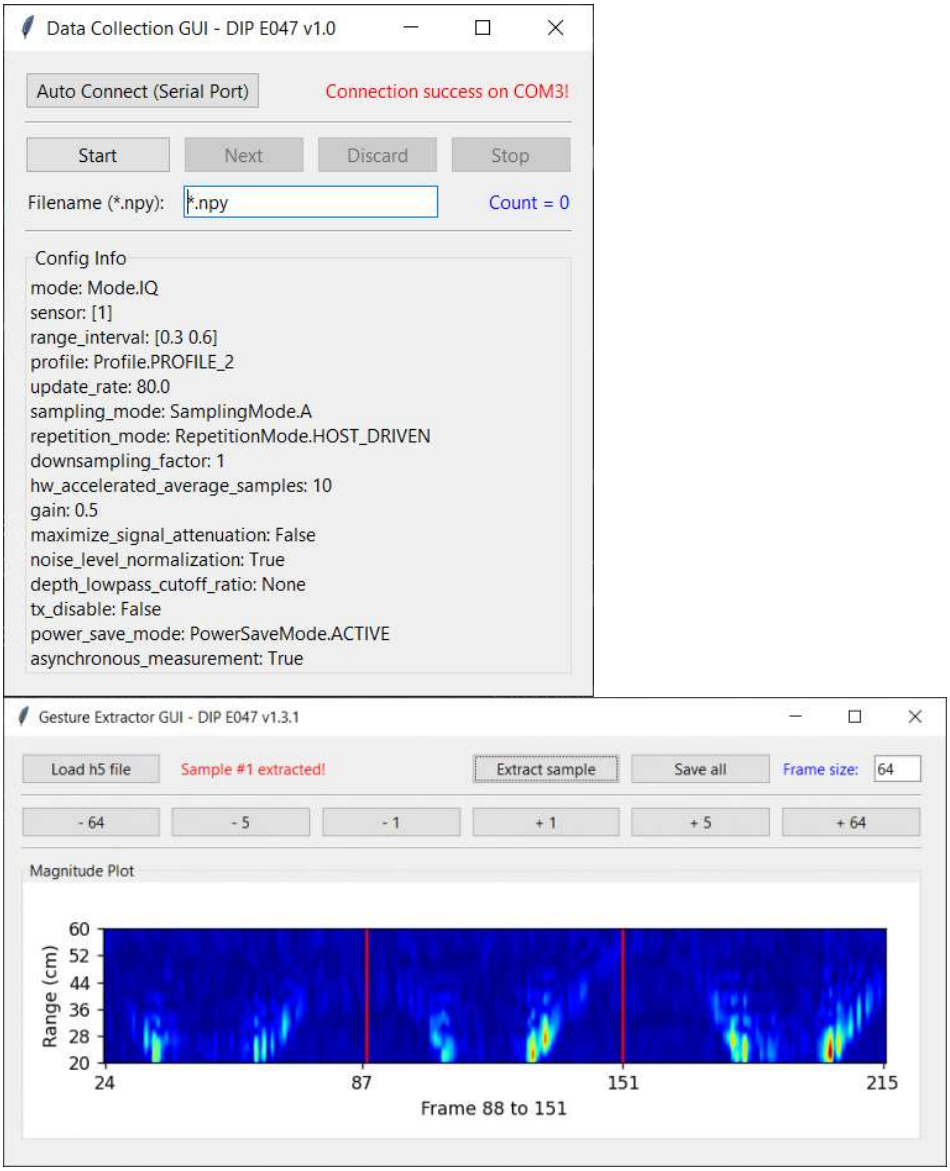
Edit

Posted on 6 Oct 2021, Wed by PHILIP LEE HANN YUNG

Section 1 – We have finalised the gestures that we will collect after discussing with Prof Lu. A preview of the magnitude plots is below, which gives a good indication of whether our data collected is good. We discovered there were some alignment issues so we made a data sampler tool to help this process (see section 2).


push 1
rolling 1
pinching 1
swing 1
push 2
rolling 2
pinching 2
swing 2
push 3
rolling 3
pinching 3
swing 3
push 4
rolling 4
pinching 4
swing 4
push 5
rolling 5
pinching 5
swing 5
push 6
rolling 6
pinching 6
swing 6
push 7
rolling 7
pinching 7
swing 7
push 8
rolling 8
pinching 8
swing 8
push 9
rolling 9
pinching 9
swing 9
push 10
rolling 10
pinching 10
swing 10

Section 2 – We made a customized Graphical User Interface (GUI) for radar data collection (left) and for gesture extractor (right).



- **gesture_extractor_gui.py** is a Python script developed to automate the process of extracting gesture samples from a continuous recording obtained from the GUI tool provided by Acconeer. This fixes the alignment issue faced when recording one sample at a time.
- The tool saves extracted samples in .npz format with the intention of a small file size. To read data from these new files, please use *np.load('filename-000.npz')['sample']*, where filename is a placeholder.

Posted in Phase 2 | Leave a comment

Week 6 Updates

Edit

Posted on 20 Sep 2021, Mon by PHILIP LEE HANN YUNG

- We had gathered some images based on the gestures such as pushing, swiping left, swiping right, swiping left and right, swipe up, swipe down and switching. Based on those data gathered alongside with the help of machine learning we observed that swiping left and right could not be differentiated.

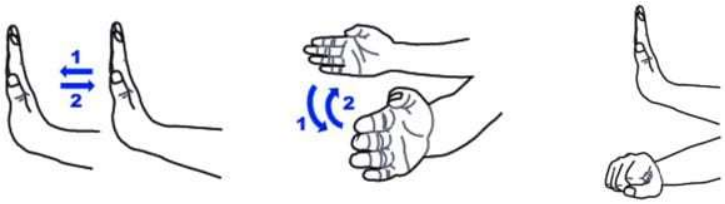
```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=7).fit(image_data)
kmeans.labels_
```

✓ 5.2s

```
... array([3, 0, 3, 1, 1, 1, 2, 6, 4, 2, 4, 4, 5, 2, 2, 2, 4])
```

left right | push | swipe down | swipe left | swipe right | swipe up | up and down finger

- Our team has done some testing on the gestures and had finalized 3 gestures, namely, (i) swiping, (ii) pushing and (iii) opening/closing palm



- We are also working on a Python script to implement the pipeline we mentioned. And for systematic and semi-automated data collection for gestures (to prepare for ML pipeline)
- We have asked the relevant department from Acconeer and they had responded the following. We would like to inquire based on the following data that was given on their end, the update rate could be manually tuned on the GUI, but specifically for the time window segment for the STFT where should we change the time window to 1 second.

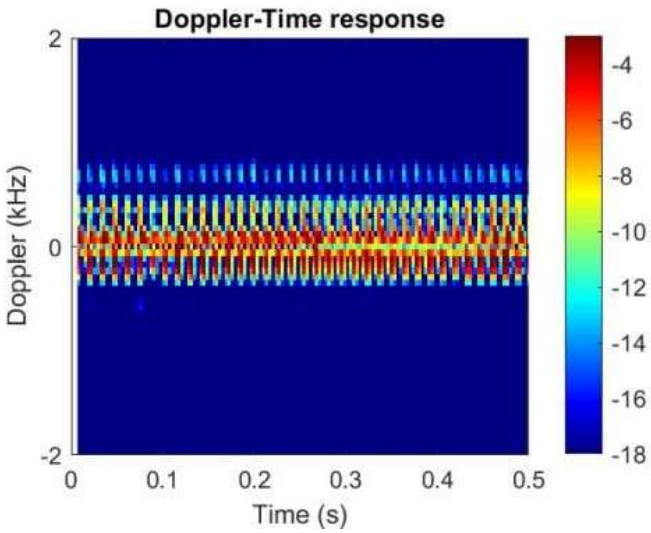
Posted in Phase 2 | Leave a comment

Week 5 Updates

Edit

Posted on 20 Sep 2021, Mon by PHILIP LEE HANN YUNG

- Examined the MATLAB script to see what features does it extract, and started the conversion to Python (except one function which has an error)
- Obtained data samples to see differences in STFT signatures
- Packaged the sensor hardware into a container for safekeeping.



Posted in Phase 2 | Leave a comment

Week 4 Updates

Edit

Posted on 2 Sep 2021, Thu by PHILIP LEE HANN YUNG

Things we learned today:
– IQ mode: Inphase and quadrature data has the most information so we should extract this as raw data to be processed in MATLAB/Python.
– Background data at fixed target distances (wall) of 0.5, 1.0, 1.5, 2.0 m.

Finalised pipeline:
– Collect radar sensor data (IQ mode) and export it as h5 file
– Parameters: frames 128/256, rate: 30Hz
– Process extracted h5 files via Python (code converted from MATLAB)
– Perform STFT on processed h5 file and export as png as a signature

– Intuition of RGB compressed into a grayscale image
– RGB equivalent to depth
– Window Size
– Image file (png file format) is a numpy array of shape 2
– Png file feed into ML Model to perform object classification via sklearn
– Implement into Raspberry Pi and integrate into workable product ie volume control, switching on/off

To do this week:
– Purchase container for sensors (Jun De)
– Make a script for h5 reading and STFT to generate signatures in Python (Philip)
– Gather sample hand gestures eg. 5 x 3 different gestures and perform STFT to compare results
– Read Chloe's FYP report so we are clear about the theory
– Watch https://www.youtube.com/watch?v=0xROkVQ1XWQ or https://www.autosofdallas.com/blog/learn-how-to-use-the-bmw-5-series-gesture-control-feature/ for inspiration for possible hand gestures

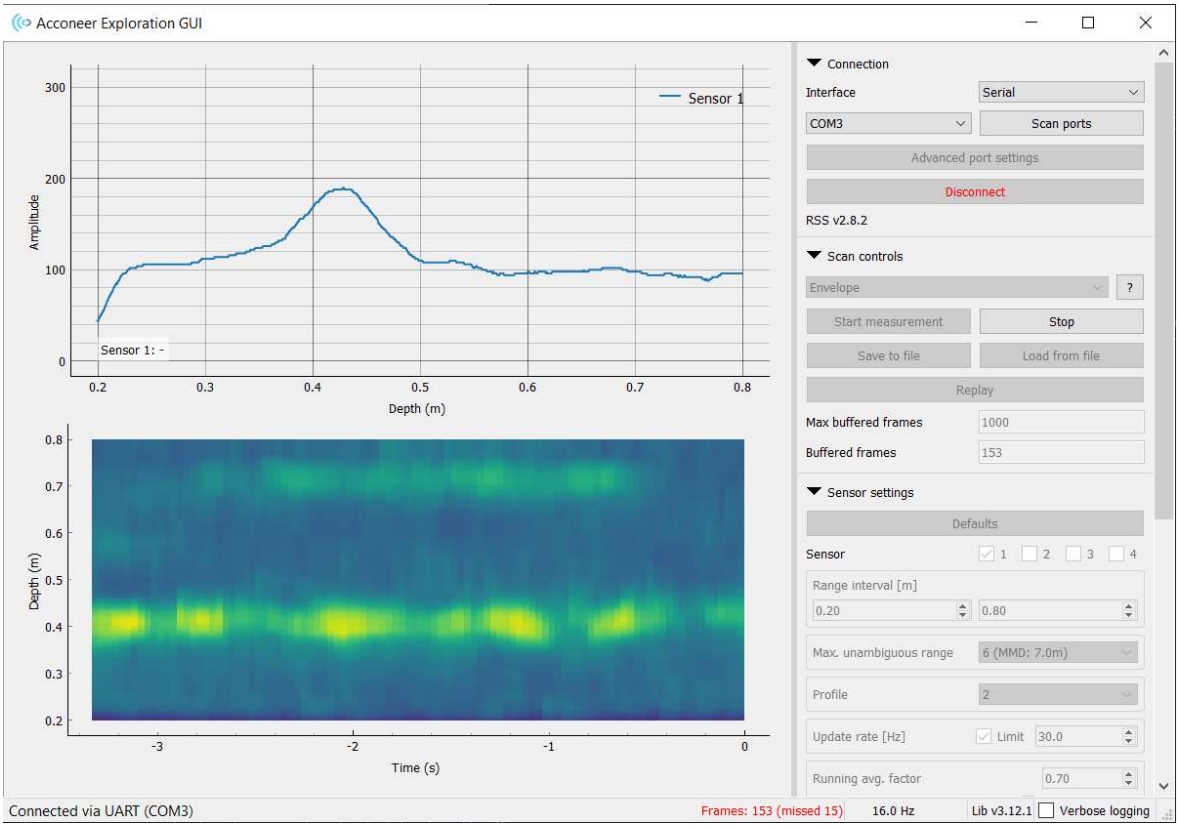
Posted in Phase 1, Phase 2 | Leave a comment

Week 3 Updates

Edit

Posted on 26 Aug 2021, Thu by PHILIP LEE HANN YUNG

The devices had arrived already, so for this week we were mainly focused on setting up the devices. We spent some time reading and downloading documentation for the sensors found at https://acconeer-python-exploration.readthedocs.io/en/latest/index.html and the Acconeer developer site. Before using the sensors, we need to flash a binary file onto the device. During the configuration, we found that only 1 of our member's laptop was able to interface with the devices and we spent most of the time debugging this. In the end, it was because we had to install a virtual COM driver so that the radar sensor will be recognised as a serial UART. We then spent our time experimenting with the demo application to see the potential use cases and limitations of the radar sensor. For the rest of the week, we plan to explore the data that can be collected by the sensor via the h5 file format.



Posted in Phase 1 | Leave a comment

← Older posts

blogs@NTU – a service by the Office of Information, Knowledge and Library Services

Best viewed in Chrome/Firefox/Safari