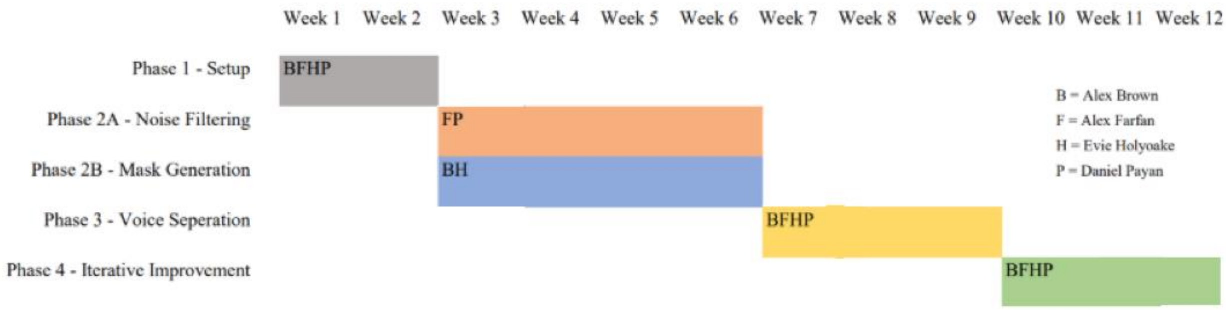
Alex Brown, Alex Farfan, Evie Holyoake, Daniel Payan

Team daea

Bi-Weekly Progress Meeting 4

4/5/2022

Gantt Chart:



<u>Phase 1 - Setup:</u>

- Set up workspace/environment
  - Version control using Git
  - Python3 development using Jupyter Notebook through in an Anaconda environment
  - Simulink and Matlab development using Matlab IDE
- Begin implementing initial filters
- Use Matlab to set up FIR/CIC filters to familiarize ourselves with noise reduction implementations
- Set up/Initialize Coral Board
  - Will need a similar development environment for testing of system implementation
  - Produce .wav files using on-board Digital PDM Microphones
- Research further into quantitative analysis system performance
  - As this is an investigative development, standards of performance may be subject to change
  - Performance may be relative across iterations, rather than compared to a known metric, as MFCC mask voice separation is a not well-explored domain

<u>Phase 2A - Noise Filtering:</u>

- Alex Farfan and Daniel Payan will lead this phase

- Attempt to implement separation of background noise from voices
    - Separate background noise from a single voice
    - Proceed to separate background noise from multiple voices
    - Proceed to separate background noise from stacked voices
- Implement MFCC feature vector production through Matlab
- Output MFCC feature vectors to file (for later use by Python3 code)
- Begin testing filter performance by comparing clean audio samples to artificial noise that was filtered by the program

<u>Phase 2B - Mask Generation:</u>

- Alex Brown and Evie Holyoake will lead this phase
- Develop infrastructure for interfacing with potential audio clip databases
- Implement MFCC feature vector production through Python3 (utilizing the Librosa library)
- Input MFCC feature vectors from file (previously output by Matlab code)
- Research into mask generation based on existing in generating masks in the Time-Frequency spectrogram domain
- Setup system to create an "enrollment profile" of audio clips for individuals from database sources
- Visualize MFCC time-domain data
- Implement a mask generation machine learning algorithm
    - Uses MFCC feature vectors produced from enrollment profiles
    - Produces a "mask" of multipliers/scalars that represent an individual's vocal presence in the MFCC-domain

<u>Phase 3 - Voice Separation:</u>

- Continue training mask generation model
- Use the MFCC feature masks to attempt to separate a known speaker from other speakers
    - Experimentally determine if MFCC feature mask multipliers are sufficient for attenuating non-target voices
    - Experimentally determine if a voice separation model is needed
- Reconstruct audio streams based on MFCC feature vectors with Python3 (utilizing the Librosa library)
- Implement the noise filtering algorithm on the Google Coral board, testing its performance

- Experimentally determine how to use both microphones concurrently whether this improves performance
- This must filter out noise recorded using the on-board microphone(s)
- Determine how to improve performance to meet latency requirements

Phase 4 - Iterative Improvement:

- Iteratively improve noise filtering implementation
- Iteratively improve mask generation model
- Iteratively improve voice separation implementation
- Iteratively improve testing mechanics

Git Repository:

1. Provide an update on your progress according to your Gantt chart in the proposal; provide over/under of your progress vs plan

We're in week 11 or Phase 4 of our Gantt chart, but we are now following our development plan which we are in line with. After successfully implementing the wrapper, we have moved on to training our own model following the ConvTasNet architecture with a customized audio data loader. We have been able to successfully train a model with one epoch using our custom data set using Google Colab.

After communicating our difficulties with the Google Coral Dev Board after all means of problem solving were exhausted, we have decided to use a Raspberry Pi as our embedded environment. Though this will no longer be what our sponsor asked of us, we will at least be able to have a demonstration and functioning version of our project.

2. Provide an update of teamwork: who is doing which tasks

- Alex Brown & Evie Holyoake
  - ✓ Created custom data loader
  - ✓ Trained custom model with ConvTasNet architecture
  - ✓ Interfaced with Google Colab to utilize their free GPU
  - ✓ Set-up Google Colab environment
  - o Test custom model
- Alex Farfan & Daniel Payan
  - ✓ Re-flashed Google Coral Dev Board
  - ✓ Met with Professor Bell about Dev Board issues
  - ✓ Begin environment set-up on Raspberry Pi
  - o Embedded environment for voice-splitting

3. If any new bottleneck/hurdle is discovered, please describe; also describe your solution to address the hurdle

- ☹ Google Colab Limitations
  - o We were able to successfully utilize Google Colab's free GPU, but due to the amount of resources needed to train our model we were unable to access another GPU after one epoch completed
  - o We haven't had a chance to test the time frame of limitations
  - o We may have to purchase a subscription to Google Colab to properly train our model

4. If you are behind schedule please provide a plan to address

With the development plan we are not behind schedule.

5. Describe your next step for the next 2 weeks

- Alex Brown & Evie Holyoake
  - o Determine a way to circumvent the GPU limitations in Google Colab
  - o Leverage the custom data loader for testing to efficiently test the model
  - o Implement early stopping to prevent over-training
  - o Iteratively improve the model

- Alex Farfan & Daniel Payan
    - Create a MATLAB CIC/FIR filter executable to run in front of the wrapper
    - Set up Raspberry Pi environment
    - Implement physical interface to indicate recording and number of speakers