EE / CprE / SE 492 - sddec22-13

Simultaneous Call Transmission

Bi-Weekly Report 4

10/12/2022 - 10/25/2022
Client: Collins Aerospace
Faculty Advisor: Dr. Andrew Bolstad

Team Members:

Sullivan Jahnke - Project Manager and Machine Learning Co-Lead
Json Rangel - Reports, Webmaster, and Communication Systems Co-Lead
Tyler Mork - Reports and Communication Systems Co-Lead
Austin Rognes - Research and Machine Learning Co-Lead
Hani El-Zein - Digital Signal Processing Lead and Research

Past Week Accomplishments:

- Sully: Batch Generator, Refactoring
    - Implemented a generator function that yields batches of data to the Keras fit() function.
        - After reading documentation for Keras and considering our mistake with implementing batch training, I found that the fit() function can take as input a “generator”.
        - This generator is a function that uses the ‘yield’ keyword to yield a tuple containing a batch of data with their corresponding labels.
        - Keras requires it to be “infinitely iterable”, which means that it is an infinite loop.
            - Specifying the “steps per epoch” parameter in the training is what stops the loop.
    - Refactoring
        - We were able to settle on a “final design” for the way our algorithm preprocesses data (i.e. shaping and labeling).
        - We had a branch being used for our “experimental” development, and this branch also contained the stuff we actually wanted to implement.
            - To make merging easier, I created a separate branch off master named ‘simplify’. Using this branch, I brought in changes (one at a

time) from the unorganized branch, and merged from the 'simplify' branch. Personally I thought this was much better than going through the tedious process of resolving merge conflicts in the unorganized branch.

- Json: Simulated Data Generation, Labeling & Simulation Tweaks
    - Continued to run simulations to generate data.
        - Generated five more data sets containing an interfering signal.
            - Interfering signal varied at entrance time.
            - One data set contained an interfering signal that would interfere intermittently.
        - Exported complex data to .csv file and pushed to the personal branch of the Git Repository.
    - Modified simulation Simulink to create an intermittent interferer.
    - Explored new audio data to input to simulation.
- Tyler:
    - Continued archiving ATC live scripts for inputs to data simulation
        - Downloaded MP3 files are chunked into 1 to 10 second intervals and saved as separate .wav files
            - 1 to 10 seconds was chosen as standard considering most Air Traffic transmissions do not exceed 10 seconds in length from one transmitter
        - .wav files are further upsampled to 48000 Hz to be utilized in Simulink.
    - Implemented matlab code of Doppler Shift to be imported into Simulink simulation
        - Shift is considered to be a frequency shift at DC baseband
        - Variables include commercial aircraft velocities, VHF [Very High Frequency] band frequencies, and consideration of aircraft moving towards or away from receiver
        - If Required by Collins: Consideration will be needed for military applications and for supersonic vehicles
- Austin:
    - Created Diagram showing exactly how our data is imported and transformed for use.
    - Ripped out data importer and reimplemented data importing according to new diagram.
    - Created a mat to csv/bin converter script to automatically generate data from either .mat or .wav files.
- Hani:
    - Research on doppler and implementation
    - Revised simulation and attempt documentation

Pending Issues:

- Discuss with the team and Dr. Bolstad on time signal manipulations for algorithm to increase overall performance
- Determine whether frequency shift caused by Doppler is negligible at the Commercial Airliner level for radio transmissions specifically.
- Begin EE documentation for simulation.
- The program should not continue if the ReLU output network is specified and the correlating WindowLabeler is not.
- Need to start training multiple times a week and evaluating parameters that change the performance (evaluated using the new evaluate and predict functions).
- Documentation could always be worked on.

Individual Contributions:

Team Member	Contribution	Hours Spent	Total Hours
Sullivan Jahnke	Batch Generator, Refactoring	7	65.5
Json Rangel	Data extraction, generation, and labeling. Simulation tweaks.	10	70
Tyler Mork	Doppler Effect, Archiving Live ATC Segments	8	59.5
Austin Rognes	Diagram of current Data importer	8	50
Hani El-Zein	Research and Doppler Effect	9	47

Plans for Coming Week:

- Sully
  - Get the master branch finalized.
    - Still have a few things to merge / test.
  - Train, Train, Train, Train, Train
    - Once the last few things get implemented, it is time to really dive into training.

- Tyler
  - Evaluate whether Doppler Shift varies a considerable amount over a 10 second interval [Dependent on velocity of aircraft and distance from receiver?]
    - Would require Doppler Shift to be a time varying variable within Simulink Simulations
  - Continue archiving piece segments of Live ATC audio transmissions
- Json
  - Continue to extract and analyze simulated data.
    - Vary the audio being tested.
    - Label data once extracted.
    - Export to suitable file format for Software team (.csv).
    - Verify signal data sent is complex.
  - Push labeled data to Git for Software to test.
  - Generate a test data set for the software team.
  - Upsample new LAX ATC audio.
  - Begin simulation documentation.
- Austin
  - Train and tweak config files. Find best form of data to use in training.
- Hani
  - Continue work with doppler and improve model.
  - Add documentation.