ZX-1 — Mobile Computing for Dementia Care

CS 4850 - Section 04 – Spring 2025

Final Report

Evan Bailey, Eric Bryant, and John Preisinger

Professor Perry

April 20, 2025

Website: in progress

Project is working under an NDA
(No source code or URL for Github)

Estimated hours: 432
Actual Hours: 300

Status: 80% complete

Table of Contents

Introduction

In recent years, the applications for machine learning (ML) have increased exponentially. While there are justifiable luddite esk complaints with the expansion of ML in industry, there is an appropriate use case when it comes to healthcare. With the use of ML, or just Artificial Intelligence as everyone calls it, diagnosing illnesses has become more accurate, speedy, and detailed. In this project working with Professor Xie, the group aimed to fine tune and test a software that helped in the early detection of dementia. To detect such dementia, the audio of an individual speaking is analyzed using a model trained on an assortment of audio clips. These audio clips consist of two categories, those with dementia and those without. The end goal of Xie's project being the development of a downloadable app where an individual can record themselves speaking into the app and get some sort of result from it. This group in particular worked on a segment of the process that involved the augmentation of the audio clips, conversion into vectors, and the training of the model with said vectors. The audio clips must be augmented to acquire the important information once converted to a vector. The last goal of the group was to record and demonstrate fixes for any error that might have occurred in the process, which there were plenty. Due to this project being under an NDA, no code can be shown but hopefully this explanation can suffice.

Requirements

Technology

Our first requirement involves understanding the softwares included in this research. These softwares include a few cloud platforms such as Github, HuggingFace, and Google Colab. The source code modified for this research is stored in a Github provided by Professor Xie. Github is a free service that provides version control for codebases, HuggingFace is where the trainer for this project is stored, and Google Colab is where the code will be modified and run. It is also important for the team to understand other software such as Audiomentations and wav2vec2 which are used in the conversion of audio clips to become readable by the trainer.

Audiomentations

Our primary goal is to investigate how augmenting audio clips with Audiomentations affects the accuracy in the trainer. So we will need to test how accurate the trainer is with no augmentation as a control, then as many other augmentations as we can to see how they compare to that control. It is also possible that we can combine augmentations to create even more accuracy, so a lot of testing is needed.

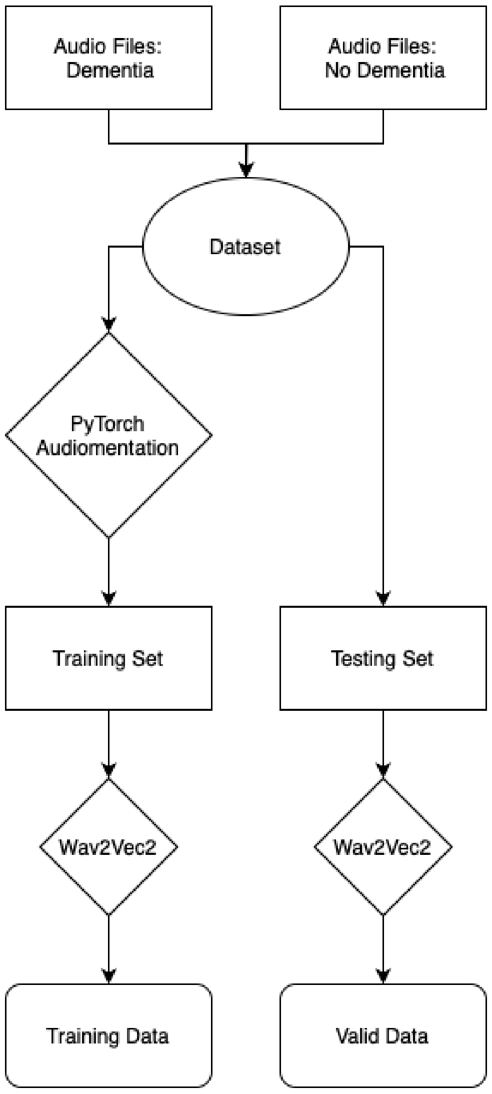
Wav2Vec2/Visualization

We also need to understand what is actually happening when an audio file gets sent through Wav2Vec2. This basically turns an audio file into a text representation. To do this we will run Wav2Vec2 on an audio file, print out the raw data generated, and we will go through that data to try and understand what is actually happening. There is something called the feature extractor which is something we want to focus on. We can then take this data and visualize it using different methods. Two recommended by Professor Xie were PCA and T-SNE. Of these two, T-SNE seems to be the easiest to understand and implement so we will stick with that one. T-SNE specializes in taking a multidimensional array/matrix and reducing it to a readable two dimensions. This will also be useful when comparing the different augmentations with each other.

Application

The last step of this process is to investigate how an Android application would look and function. So we need to find a way to implement a lightweight version of the dementia testing model into an android application. To do this we need to watch tutorials, practice, create a prototype, and just see if we can make it work. An idea for it would be to just have it be a voice memo type app. This app would record your voice or accept audio from your phone. This audio would then be sent to a server that houses the model, test the audio, then send it back to the phone with results so that the phone does not have to do any work. This would involve setting up a server and then investigating APIs that could be created for that server.

Analysis/Design



This flowchart represents the navigation of the sample audio files and how they end up as training/valid data. There are samples with those with and without dementia speaking, that comprises the initial dataset. Randomly those files get split up, some go to Pytorch Audiomentation and some go to the testing set. Those who went to audiomentation get modified through added noise and/or speed control. Once completed, they comprise the training set. Both sets are then processed through Wav2Vec2 which turns the audio clips into vector representations of themselves. Once completed, they end up in Training Data or Valid Data. Training data is what is used to train the neural network while valid data is tested against that neural network for results.

Project set up steps

1. Get code from our source
2. Set up hugging face account and get a key
3. Get the audio clip data into google drive in folders named accordingly
4. Inside Google Colab, open the code
5. Insert the hugging face key and google folder link
6. Run the code

7. Interpret the result

Development

Cloud Platform

We were given specific cloud platforms to use. Github was where we took the code from to use. It was from an open source dementia repository. We also used Google Colab which is a Jupyter Notebooks but like a google doc. Inside of it, we utilized Hugging Face to store the ML libraries that we loaded into Google Colab.

Dataset

The dataset we have is a combination of audio clips consisting of those with or without dementia speaking. This is to train the dataset so it knows what it sounds like when someone with dementia speaks, and examples of what someone without it sounds like. From this it can maybe find the small differences that a human may not even notice.

Pytorch Audiomentation

This is a process where you modify the audio file. This can do things such as make the data better for ingestion. You can do things such as add noise, change the pitch, and so on. There is research about which methods are optimal for dataset use.

Wav2vec

This is a process that turns an audio clip into vectors. The audio files can not be interpreted by the ML model while still stored as waveform files. This is the step that is needed to add it to the ML dataset so the AI can reference it. There is a mathematical representation of that audio clip now.

Visualize Data

Once we have these arrays, we can visualize the data for comparisons. We Use two different techniques, T-CNA and PCA. These just allow us to understand how the audiomentation changes how the data is seen by the dataset.

Research Papers

We are currently reading more about audiomentation and how it helps with generating good datasets.

Test Plan

For the test plan, Evan and John will be testing the code as they are in charge of programming. They will be testing it in Google Colab using files from google docs and the tools from Hugging Face. The test will involve seeing if the Google Colab code will run through, do the needed audiomentations, give a visualization, then load into the trainer.

Methodologies

We were tasked by the professor at this point to find any errors in the process, so we have been testing this code and getting different errors. Most errors are related to incompatibilities between different python libraries being used, mostly numpy and scipy. These errors are then documented and written down for the professor. We also have been trying to graph the data after trying different audiomentations. This is mostly to see which ones are the most effective at helping the model.

Failures

Analyze how PyTorch audiomentations affect model accuracy - The big problem with this was how abstract the results were. While it was recognizable that certain audiomentations affected accuracy through graphing, it almost felt more like intuition rather than direct evidence.

Successfully run wav2vec on an audio file / Print and analyze raw wav2vec outputs - both of these issues arise from Google Colab downloading new versions of libraries without the permission of the user. They are classified as a success and failure as it can do it at any time without the knowledge of the group, a solution to this problem is discussed in version control.

Develop functional Android app prototype - This failure is not one of capability but one of time, the group remained on the audio clip augmentations resulting in a lack of time for the implementation of this app.

Features Tested

Test	Pass	Fail	Severity
Access and review Github / HuggingFace resources	X		
retrieve and run provided Colab code	X		
Analyze how PyTorch audiomentations affect model accuracy	X	X	
Successfully run wav2vec on an audio file	X	X	
Print and analyze raw wav2vec outputs	X	X	This breaks a lot
Visualize intermediate data using T-SNE	X		
Compare results between raw and augmented data	X		
Develop functional Android app prototype		X	

X in both is for ones that work inconsistently

Version Control

Due to the use of Google Colab for this project, as per our instructors requirements, there are not any types of version control. Actually this particular non-use of version control did cause problems. As mentioned before, one of the biggest issues facing the program was Google Colab automatically installing new versions of libraries even when this updating would completely make the program unusable. Rather than use Google Colab, it would have probably been in the group's best interest to use Jupyter Notebooks and linking it to a github so that the program could have been rolled back in cases such as this.

Summary

The goal of this project was to investigate different audio augmentations on audio clips and when run through a trainer, how it affects the accuracy of the trainer. The audio clips in question are of people with dementia and without dementia. The purpose of this trainer is to find distinctions between those types of audio clips so that when given a new piece of audio, it can categorize it accurately. The purpose of these augmentations is to give the program an easier time finding these distinctions. A few augmentations tested were Gaussian noise, pitch shift, and time masking. Through these tests, it was found that pitch shift was the most effective when paired with Gaussian noise. The biggest issue with this research was the use of Google Colab. When tested, it could take up to three hours to test one augmentation type when hundreds needed to be processed. Not only that, but half of the time the Colab would time out or break, needing a complete restart.

The app was not completed but the end goal was always to create a mobile way to test for dementia, so hopefully as this projects moves forward with Professor Xie and can come to fruition.