# Cover Sheet Information

|  |  |  |
| --- | --- | --- |
| *Project Title: Smart Meeting Minutes* | | |
| *Team Name: Silverback Tech.* | | |
| *ID Number* | | *Name* |
| 1 | 101228706 | Mohammad Mobin |
| 2 | 101214608 | Masrur Rahman Zahin |
| 3 |  | Noor Ul Ain Khurshid |

# Introduction

This project is based on Audio Transcription. We are to build 2 different deep learning models and compare them based on their evaluations. This project falls under the category of machine learning. As any machine learning problem, we have a goal, dataset, model and evaluations. In our case, our goal is to build a model which recognises words from audio input. A basic supervised learning model requires the dataset and its labels or supervisory signal. The dataset must be split to training and testing set and must be fed to our neural network. Just like how a regular model requires lots of training data to learn a specific class, we have training data for each word. Each word is a class. There are multiple speakers speaking the same word, which have their own waveform, although different speakers, the waveform appears similar. We feed the neural network different waveforms of the same word, so that it can be able to recognise that word from a different speaker. Our dataset is in audio form with a folder of different words, which contains multiple speakers speaking out the word. Our program converts the audio to its respective spectrogram which becomes our data to be trained and tested, i.e. the Xtrain and Xtest variables. Then we attach them with their respective labels, which are the words, i.e. the ytrain and ytest variables.

# Overall System Architecture

We are building two architectures: CNN and CNN based on transfer learning

For CNN, the name speaks for itself, a few layers of convolution and pooling layers are kept for feature learning and reduction respectively, reduce the complexity of the image, learn only the important features, and then predict using a fully connected layer. We are using 3 layers of convolution layer and pooling layer with a final dense layer.

For the second model, CNN architecture was used in conjunction with a pre-trained model using the ImageNet database. The model consisted of one Base layer followed by MaxPooling and Dropout layers, and finally, four Dense layers to learn all the features from the image. Softmax was used on the final activation layer. Adam was used as the optimiser, and categorical\_crossentropy was used for the loss function. Early stopping was also implemented to prevent overfitting with the patience of 10 epoch.

# Data collections and annotations

Dataset collections were the most challenging part of our task since our coding and everything related to our work depends on the quality and quantity of the dataset, its cleanliness and usefulness for our implementation methods.

We have looked over three different datasets:

Our first dataset was a very heavy file of around 75 gigabytes. It was good, except the dataset was in sentences; we needed single words. Initially, the team thought of splitting it using a code, but the folder was too large for our computers to handle and work with and thus causing our computers to crash several times while working with it. However, by that time, we had found our second dataset. Due to all this, we have not looked into the data labelling method for our dataset.

Our second dataset looked very promising. It was clean,well organised, and was very useful in terms of our implementation stategy. The annotations were done by the csv files provided, where one column referred to the link of the audio file, and the other referred to the word( the label). The dataset had no sentences, and we proceeded with it till last week. It also had separate CSV files for train and test, meaning the splitting has already been done for us. However, due to computer lagging issues of two team members, we thought to write codes out of theory and debug it all later since this way we could focus on debugging together. During final week, everything was debugged except for the audio to image conversion, we faced several issues:

First, Mubin took a portion of the dataset, three classes to be specific to test out the code for audio to image conversion, the audio was in a .opus format and it worked well printing out the spectrogram in google colab. Later when Masrur ran the codes, it gave an unknown format error, so we wrote code to convert it all to .wav format, it still did not work and gave errors. Masrur tried a random wav file from the internet and it works, we assumed there was something wrong with the method of conversion, and we could not find any other method of conversion from opus file. We decided to discard the dataset since the only option left is to use colab, and we are in no time to upload heavy files into the colab.

The third dataset was found on the same day we encountered problems with our second dataset. This dataset was also clean and organised well. Instead of csv files, there were text files with each line, being the link to an audio. We wrote code in order to split the directory to store the label names. We quickly fine tuned our code to match our new dataset, and began audio conversion to one class. To our surprise, the conversion worked, and we saved up multiple spectrograms, and checked to see if they are different and the conversion works fine. Later we converted all the train and test audios to images and were finally ready for training the model since our training parameters were ready.

# Implemented Machine Learning Techniques For Automated Speech Recognition

CNN and rnn(lstm), I guess

Transfer learning technique was used for second model

# Scenarios/Examples To Demonstrate How The System Works

# Critical Analysis Of The Implementation

As we know, CNN is deep learning neural network model preferably used for image classification and is very good at extracting features from images and learning it.

In general image data has a lot of pixels, the traditional fully connected layers would be too much for the model to handle, here is where CNN comes in, basically it takes in the image, uses convolution layer to extract some features using a kernel or filter and then reduces them using pooling layer, it then keeps repeating the previous two steps depending on how complex the features are, then provides the important features to the fully connected layers for processing.

In other words, CNNs take an image, shortens it to just important features and proceeds with fully connected layers.

In our case, our data was audio, we had to convert the audio to image, its spectrogram, basically every audio has a waveform image. The idea was to use the waveform of multiple speakers of the same word, which would be similar, and the model would learn the similar features so that during prediction, the model would predict that specific word.

We provided 3 layers of convolution and pooling layers with the activation function of relu. And the final layer with the number of nodes same as that of the number of classes with the activation function of softmax since we were working on multi class classification.

For rnn, (masrur please fill in)

Later we wrote code for evaluation using confusion matrix and plotting graps of two models and compared them

# Practical Application Of The Description

(insert spectrogram pictures, screenshots of the working model, the gui, explain how the model works with proof)

# Summary

In summary, all the sleeps we lost, the (non-existent) sex we missed, the appetites we lost, the sanity we have left, have kinda not been worth it?

The model making was fairly easy and was no big deal, our only challenges were audio conversion, the datasets, and the gui, other than that everything was alright.

Speaking about the model, in conclusion, we would like to say that the model could have been implemented better if it werent for the opus file format of the second dataset, due to less time, we switched to a simpler dataset in order to show demonstration of the model and its working and our knowledge and work to be assessed in this situation. Although less data, the working and the coding would have been the same if we had bigger data.

Project Presentation Link: Hold on homie.

# End of Semester Final Deliverables Check List

The following documents must be included in the final submission by the due date.

|  |  |
| --- | --- |
| Deliverables Check List | Submitted Yes/No/NA |
| Main Document |  |
| * Cover Sheet | Yes |
| * Check List | Yes |
| * Project Report (Video link attached) | Yes |
| Appendices |  |
| * Source code and executable | Yes |
| * Who did what? | Yes |
| * Presentation slides | Yes |

**Who did what?**

As you are to be individually assessed it is necessary to ensure your marker understands your individual contribution. This document is to demonstrate who was responsible for each piece or contribution to each piece of work in your project. The following is a template to present and **must be signed by all team members**.

|  |  |
| --- | --- |
| Project Title | |
| Mohammad Mobin | Activity (these are suggestions of the ways you may have contributed. There may be others and some may not be applicable).  You may also add comments to further explain your contribution or partial contribution to an activity.  I collected the first dataset. I have done the coding for the second dataset, and took upon the liberty to convert audio to image, found a good code source, and implemented it within our dataset. At first, I wrote down all codes from my theoritical knowledge and forwarded it to masrur and we both collectively debugged the code after meeting each other physically, and I finalised the conversion of audio to image. I started the layout of the report and wrote major things in a vague fashion. |
| Masrur Rahman Zahin | Built and trained the second model based on transfer learning and helped debug the first CNN model. Helped clean up and process the dataset for training and prediction, the training was done on my computer. Finalised and proof read the report. |
| Noor Ul Ain Khurshid | I have done dataset collection for the second dataset, researched the gui implementation and presented my gui idea, its interface, functioning system to my team and finished it, assisted masrur in the coding and discussion in lstm network. I assisted them different sites and helped input during our discussions. I helped in additional theory knowledge for assisting masrur for report writing. It was my idea to suggest a third dataset. Also I did mini progress checks in the beginning since I wanted to learn better from my teammates. |

# Group reflection

|  |  |
| --- | --- |
| Project Title | |
| Mohammad Mobin | Activity (these are suggestions of the ways you may have contributed. There may be others and some may not be applicable).  You may also add comments to further explain your contribution or partial contribution to an activity.  I learned good team management and the importance of setting our own deadlines to meet requirements. At times, I wanted to procastrinate but since I made up our deadlines, I could not come up to disrespect the management, and followed accordingly to get the work done. I also realised that my pc was way too slow to deal with ai codes so I wrote down codes and debug d together in masrur’s laptop since that way we both could work our heads on solving problems together and faster |
| Masrur Rahman Zahin | I had an epic setup, learned to value my setup more lol, |
| Noor Khurshid | Dk fam, I legit have lunch, and the homies have already finished discussing their next step towards the whole goddamn project and I jus be like dafak jus hapenned. |

I declare this is an accurate description of team contributions of the team members

|  |  |  |
| --- | --- | --- |
| Team Member Name | Signature | Date |
| Mohammad Mobin | my sign | 24.05.2022 |
| Masrur Rahman Zahin | not final xD |  |
| Noor Ul Ain Khurshid |  |  |

# References

Theory: Cbse artificial intelligence for class 10 book

Converting audio to image:

<https://towardsdatascience.com/seeing-is-believing-converting-audio-data-into-images-5ed1a2ca6647>

Understanding melspectrogram:

<https://librosa.org/doc/main/generated/librosa.feature.melspectrogram.html>

Display the spectrogram:

<https://stackoverflow.com/questions/52432731/store-the-spectrogram-as-image-in-python>

Saving the spectrogram:

<https://localcoder.org/store-the-spectrogram-as-image-in-python>

Different codes we tried which did not work: