# Final-Project
# Latex Report

Fuzayil Bin Afzal Mir

8/02/2021

# 1 Speech Command Recognition

## • Theory

Speech is one of the most effective way for human and machine to interact. This project aims to build Speech Command Recognition System that is capable of predicting the predefined speech commands.

Speech recognition, also known as automatic speech recognition (ASR), computer speech recognition, or speech-to-text, is a capability which enables a program to process human speech into a written format. While it's commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user's voice.

## • Data Set

A data set is a collection of data.Every data set can be created in a different way.Some of the data sets can be machine generated data sets and some may be data that's recorded from human observations.

The data set of the given speech recognition system consists of the following pre recorded commands:

1)Forward

2)Back

3)Left

4)Right

5)Stop

The sampling rate of the recorded voice commands is 16 KHz and 80 utterances of each command have been generated.All the samples have been trimmed to 1 second.

**About the code:**

The code has been written using the python programming language and the packages and libraries used are : numpy, librosa, random, tensorflow and soundfile.

**Instructions:** 1)The Code is written using Google Colab.

2)Open ColabNotebook.ipynb and write the code in different consecutive cells.

3)Upload the Data Set to Colab.

4)Run the cells in the same order.

5)Locate the folder where you save your model.h5 file.

6)Start speaking when you see mike in the bottom right pane of the task bar or see red blinking dot in the title bar.

7)The repeated errors cab be minimised by setting the mike recorder time in the previous record function up to 3 sec.

- ## Description

1)Using Convolutional layers ahead of LSTM is shown to improve performance in several research papers.
2)BatchNormalization layers are added to improve convergence rate.
3)Using Bidirectional LSTM is optimal when complete input is available. But this increases the runtime two-fold.
4)Final output sequence of LSTM layer is used to calculate importance of units in LSTM using a FC layer.
5)Then take the dot product of unit importance and output sequences of LSTM to get Attention scores of each time step.
6)Take the dot product of Attention scores and the output sequences of LSTM to get attention vector.
7)Add an additional FC Layer and then to output Layer with SoftMax Activation.
8)The model is successfully built and has achieved the highest accuracy of 98.626%.

*The End*