**Project proposal**

**--Speech Commands Identification Based on Deep Learning**

1. **Members in group:**

Jingyu Deng(N11750965); Zhiwei Chai(N18705962); Ao Deng(N15059076)

1. **Problem description:**

What we are going to do is a little App which can identify several short verbal commands such as ‘on’, ‘off’ or numbers. This small project is offered by the TensorFlow and AIY teams of Google AI to help people to get started on deep learning for speech and other audio recognition problems. Precisely, this App will be able to take in a single spoken command by microphone, identify it and show up which command you have said (or operate exactly as your command suggest,like adjusting the volume of the computer,turning on/off switches,etc). All commands used are prepared in advance.

1. **Approaches:**

The TensorFlow and AIY teams prepared a tutorial:

<https://www.tensorflow.org/versions/master/tutorials/audio_recognition>.

It contains no detail codes but the method of processing the data and training the model. We will basically follow the method in the tutorial to preprocess the audio data and train the model. However, we will design our own model and apply it in the App we mentioned above.The major approach used in this speech recognition problem is utilizing deep neural network and some part of NLP technique to better train our model and improve the accuracy and robustness of prediction.

1. **Data set:**

The TensorFlow and AIY teams also offered the needed data set which contains the audio files of 30 words including number 0 to 9 and some common commands with each command having thousands of samples. The data set also has background noise such as white noise to help training. All files in the data set are WAV files with about 1 second duration time.

1. **difficulties:**
2. We don’t know where we could get more audio command samples. We can only

choose commands from our limited commands data sets that contain 30 different commands.If we have more data samples available,the effect of model training and the final prediction precision would be better. Could you give us some advices?

1. We have the idea of integrating the output of our model(oral commands) with the actual executable command in the computer(setting API between the predicted commands with the operation of computer) so as to implement a complete functionable software or system. But we don’t know whether it is feasible or how to do it. Do you have any suggestions about it?