**LANGUAGE RECOGNITION**

**PROBLEM STAEMENT**

In this projectwe will be doing language identification over six international lanuages i.e English, German, French, Espanol, Chinese and Russian. Here the user need to give recordings of the specific language as input and finnaly the input language will be predicted.

**DESCRIPTION OVERVIEW**

Language recognization is an important task in the field of natural language processing. Here the user will provide speech recordings of a specific language and then using deep learning approaches we will try to predict the spoken input language.

**TECHNOLOGY USE**

Here we will be using **Anaconda Python 3.6 , Keras 2.2.4 using TensorFlow GPU 1.14.0 backend CUDA 10 with CuDNN 10**

**INSTALLATION**

Installation of this project is pretty easy. Please do follow the following steps to create a virtual environment and then install the necessary packages in the following environment.

In Pycharm it’s easy

1. Create a new project.

2. Navigate to the directory of the project

3. Select the option to create a new new virtual environment using conda with python3.6

4. Finally create the project using used resources.

5. After the project has been created, install the necessary packages from requirements.txt file using the command pip install -r requirements.txt

In Conda also it’s eay

1. Create a new virtual environment using the command

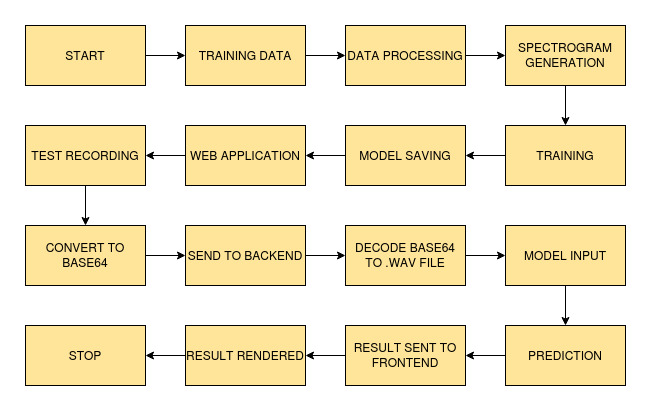
conda create -n your\_env\_name python=3.6

2. Navigate to the project directory.

3. Install the necessary packages from requirements.txt file using the command

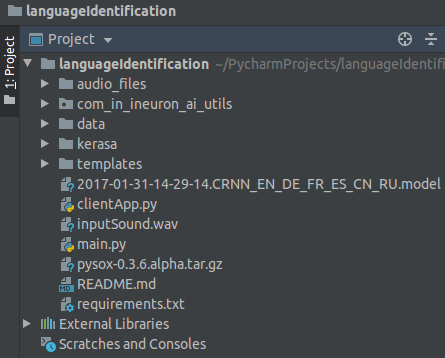
pip install -r requirements.txt

**WORKFLOW DIAGRAM**

****

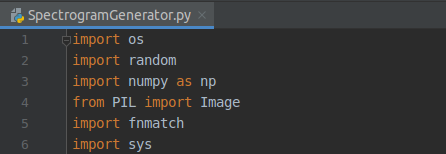
**IMPLEMENTATION**

**1. Project Directory**

****

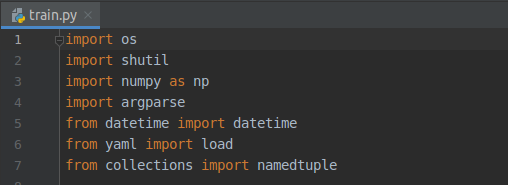
This above picture is of the project directory if we open the project folder using Pycharm.

**2. SpectrogramGenerator.py**

****

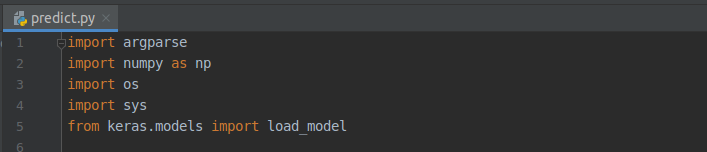
This file is present in the dataloaders folder. SpectrogramGenerator.py is used to convert our .wav speech recored files to spectrogram images which will be used for training.

**3. train.py**



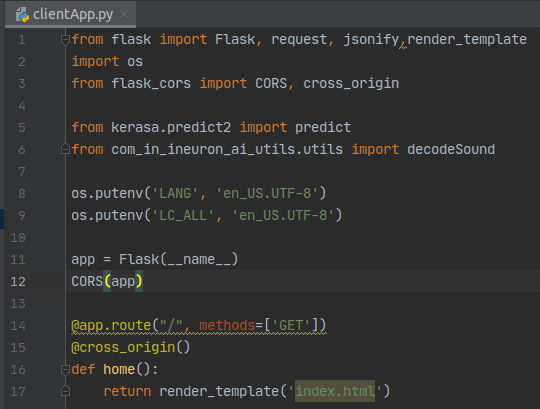
This file is used to do the training of the dataset and finally we will get the trained model which will be used for prediction.

**4. predict.py**

****

This file is used to do the prediction of the given user input with two other argument parameters.

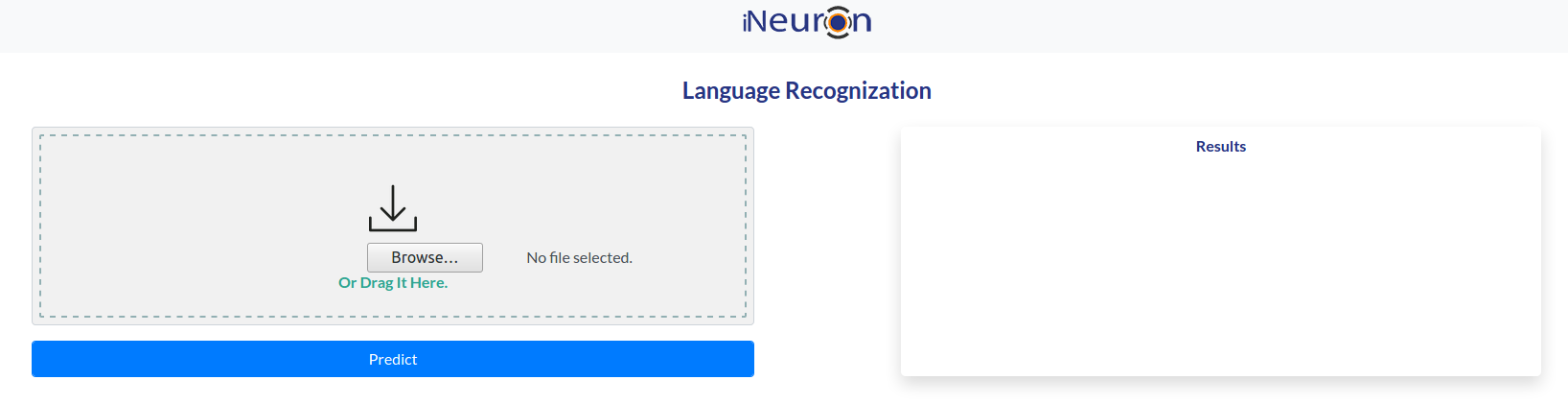
**5. clientApp.py**

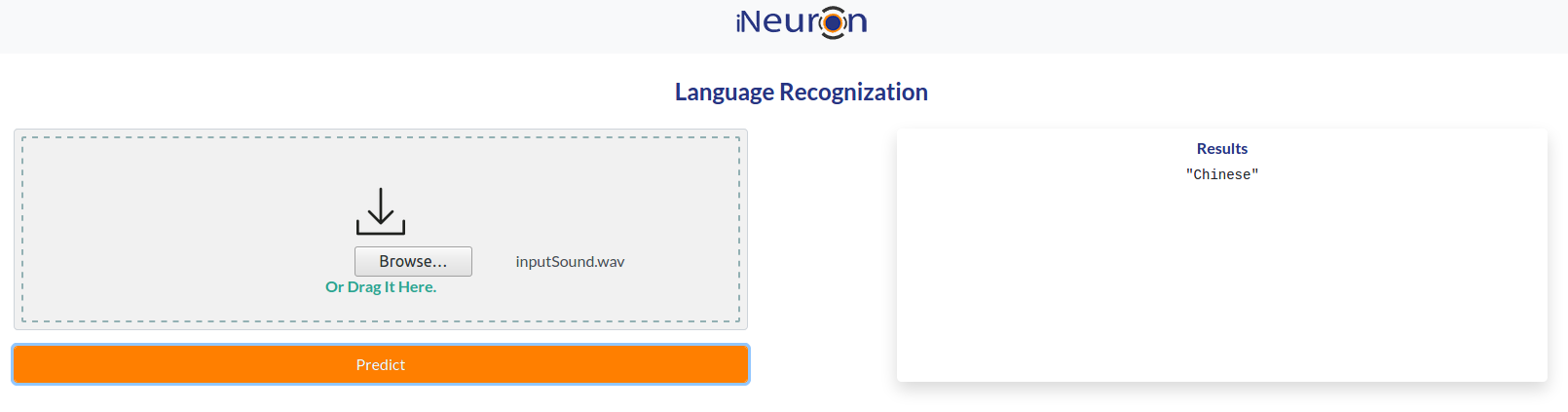


This file is the flask server file and entry point of application.

**TESTING IN LOCAL/API**

To do the test testing we need to run the clientApp.pyand after that web server will start at <http://0.0.0.0:5000/>

Enter the .wav file and click on predict button





Finally results are shown in the reults box.

**CONCLUSION**

In this project we have successfully built a language identification which can classify and identify six internatonal languages.

**COMPARISION**

Here we can do a lot of improvements. We can go with pre trained models like BERT , GPT2 etc to increase the accuracy. We can also increase the size of the training data.