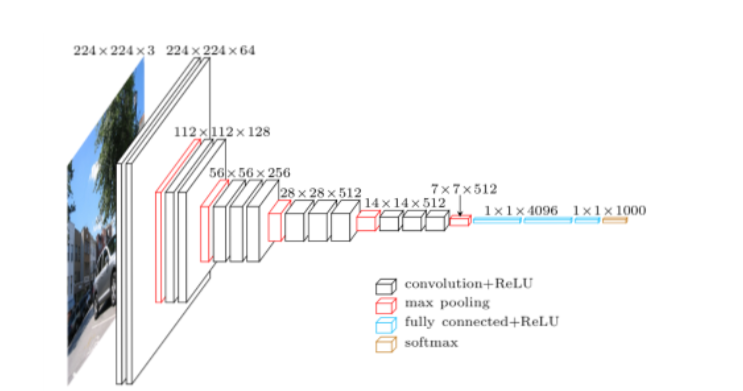
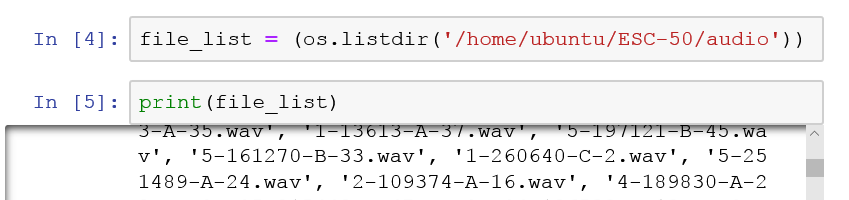
**+Your Proposed Solution [ Include Images and Tables, Do not copy content from internet otherwise your submission would not be evaluated. Please include each and every detail. This report should be in 4-6 pages]**

The solution that I preferred was inspired from vgg(Virtual Geometry Group) neural network, the advantage of this neural network is that It can classify a broad range of categories with great accuracy.It takes an input of 224x224x3 image, has 16 to 19 layers.

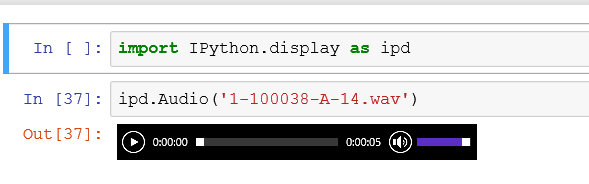


**Data Pre-processing (Explain in detail)**

Libraries used for pre-processing: os,IPython,pydub(AudioSegment) and librosa for visualization. First all the audio files(.wav files / data) have been listed into the file\_list variable using *os.listdir*



Then appended all the paths in a list called *file\_path* and for editing the wav file we used the pydub library and duplicated it to form an audio of 10 sec from an audio of 5 sec.

**

*newAudio = pydub.AudioSegment.from\_wav(file\_path[i])*

*newAudio = newAudio\*2*

*newAudio.export(file\_path[i], format="wav")*

Export saves a new file to the local drive



Cls variable contains the name of the classes of each audio file and variable labels contains the labels or the category to which each file belongs which was found out by splitting the file name as.

*for i in range(len(file\_list)):*

*splited = file\_list[i].split('-')*

*cls\_spl = splited[3].split('.')*

*clss = int(cls\_spl[0])*

*labels.append(clss)*

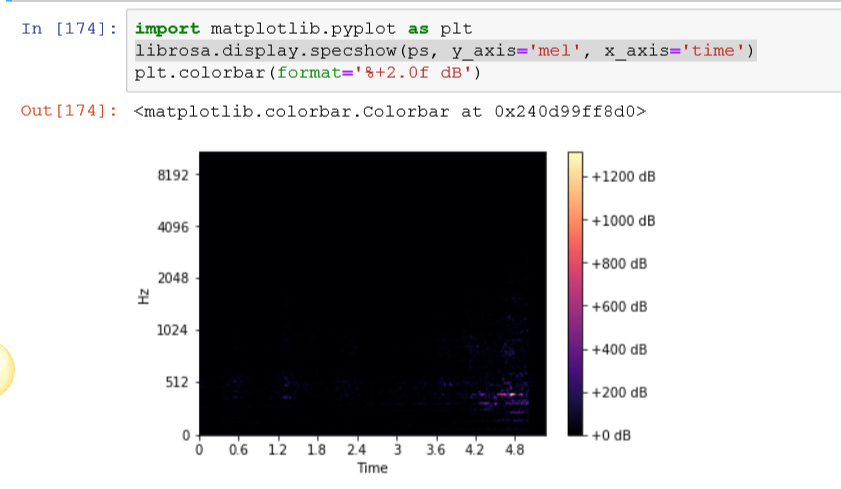
**Spectrum Generation (Explain in detail):**

Spectrum of each audio file is generated by using librosa

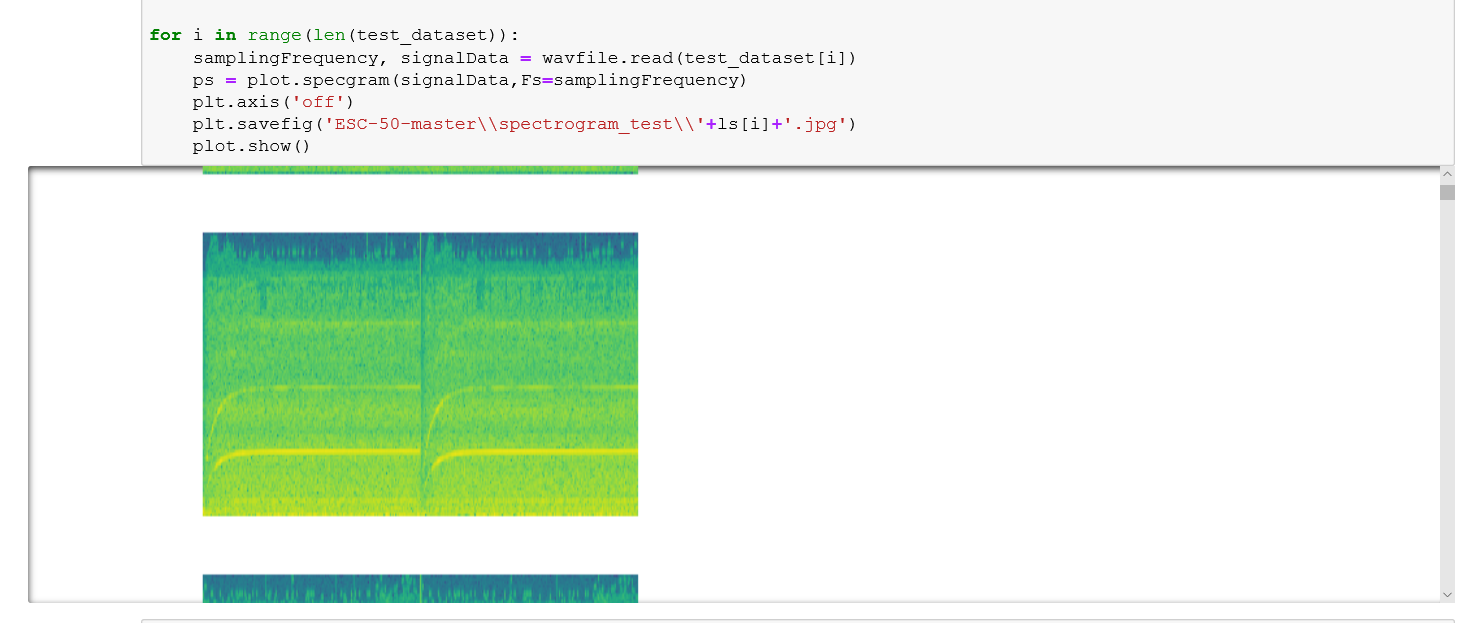
*y, sr = librosa.load(file\_path[i], duration=2.97)*

*ps = librosa.feature.melspectrogram(y=y, sr=sr)*

Here we used duration 2.97 to make the dimention 128x128. we get a spectrograph similar to this.

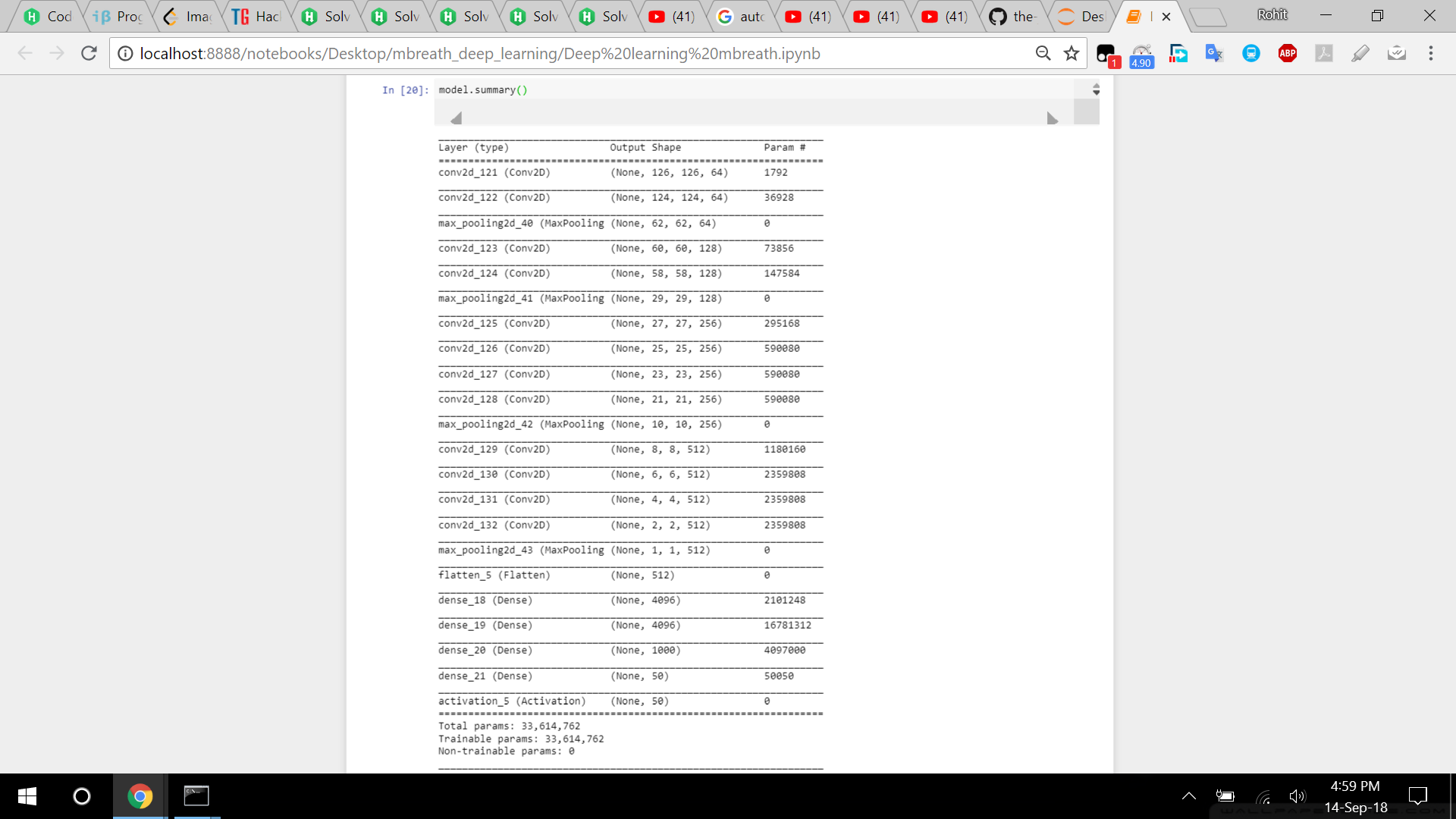


For spectrum in rgb we use wavfile from scipy.io and matplotllib. We use the specgram model from matplotlib to plot the spectrum, whereas wavfile.read gives the **rate** i.e. (int ) sample rate of the wav file and **data** i.e. (numpy array) data read from wav file



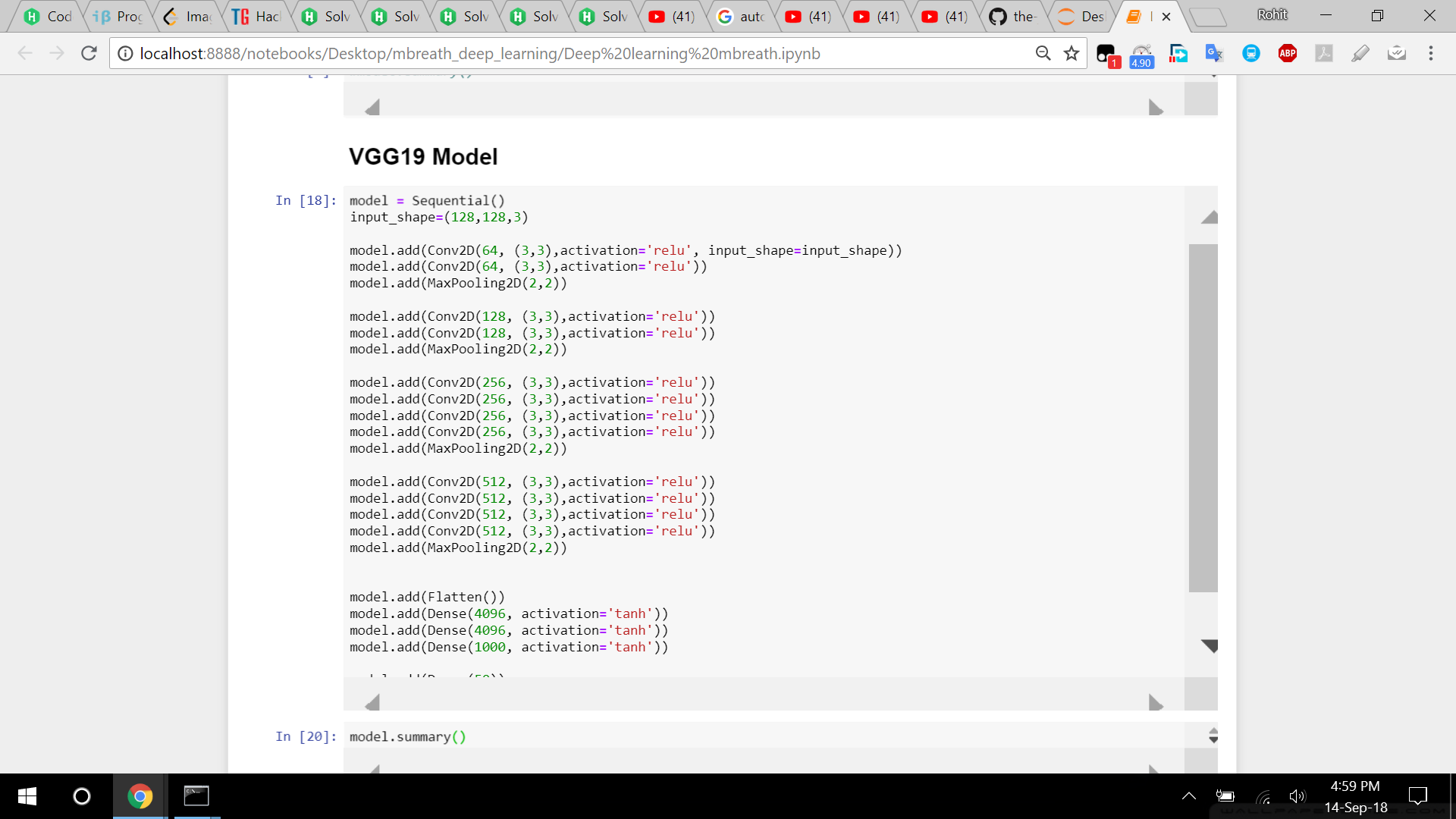
**Model Preparation (Explain it.)**

I have used a 16 layered neural network



The layers in deail:

|  |
| --- |
| Vgg19 |
| 2xcov  Size:3x3; ch:64; stride:1; Activation: relu |
| Maxpooling Size:2x2 |
| 2xcov  Size:3x3; ch:128; stride:1; Activation: relu |
| Maxpooling Size:2x2 |
| 4xcov  Size:3x3; ch:256; stride:1; Activation: relu |
| Maxpooling Size:2x2 |
| 4xcov  Size:3x3; ch:256; stride:1; Activation: relu |
| Maxpooling Size:2x2 |
| 4xcov  Size:3x3; ch:512; stride:1; Activation: relu |
| Maxpooling Size:2x2 |
| Flatten |
| Fully connected 4096 neurons  Activation: tanh |
| Fully connected 4096 neurons  Activation: tanh |
| Fully connected 1000 neurons  Activation: tanh |
| Output: SoftMax 50 object classes |



Later for real time audio classification. Wavio and Sounddevice libraries are used.

Sounddevice is used for recording an audio of 10 sec and wavio is used for saving it to the local device in .wav format. The model acquires an accurcy of 0.60 on test data.