Music Genres Classification with Neural Networks

Nitiwit Kuldiloke (2K19/CO/263)
Nitiwitkuldiloke_2k19co263@dtu.ac.in
Pooja Jaiswal (2K19/CO/275)
Poojajaiswal_2K19co275@dtu.ac.in

Abstract

The sound documents are playing the big role as information in Artificial Intelligence in term of teaching a Machine Learning model to classify or recognize the speech. The voices are the way we interact to agent such as Google in such a word "Hey Google" then the search engine will be there. In a few previous years, in Deep Learning field of study, the succeed many fields of Music suggestion, Speech Recognition. In this report , we explore 2 approach 2 architecture for music genres classification. The architectures are used the based ideas of Neural Network, 1st is Neural Network Layers and 2nd is Conventional Neural Network. Which both of architectures convert music file into mel spectrogram formed as images and one collected the data as CSV for Neural Network layer and the other collected as image to recognize the pattern of spectrogram. And the aim of these project is to understand the architecture of successful speech recognition algorithm in both Alexa agent of Amazon and Song Recognition of Google and Shazam.

1 INTRODUCTION

Classification of the Music Genre is a well-known problem in field of study of machine learning with the various of application. It could be used to classify any music in a massive music library by genre and subgenre, which can then be used to find comparable songs. Other application is used in music streaming company in algorithm of music recommendation based on deep learning. We can cluster similar songs and those that aren't by using functionalities out of a layer in model to classify Users' preferences and share the result with user for more customer consumption rate. Originally, the MFCC which is the music file extraction features is the beginning checkpoint of the project as the data set collection come from it, both CSV of spectrogram data collection for ANNs model and image of wave for pattern recognition of CNNs and. Even though, good method to approach in this project is to let neural network select feature which dataset which might be useful on its own. In this project, all of the music files which we used as the dataset(GTZAN Music Classification Dataset) are converted in to mel spectrogram which will represent the amplitude of each musical files in dataset which containing 9 different genres as 'blues, classical, country, disco, pop, hiphop, metal, reggae, rock'. Which the spectrogram of each music's genres is remarkable to classify. Then we could feed the imagine and data of amplitude as the dataset for both Neural Networks we approach in this project; amplitude of file for NNs layer approach and imagine is for CNNs. This report is divided into five sections: part 1 characterizes the problem, part 2 explain the dataset as well as the generating of Mel spectrograms, part 3 we will discuss about our implementation of our project part 4 we will talk about the result and performance of the model we created, and section part 5 we go for conclusion of our project and future scope of project and problem. Code of this project have been attached in the Zip file.

1.1 Related Work

There has been a lot of research done on using convolutional, recurrent, combination of both, and a lot more in these field of studies which we are approaching. Thus, Keunwoo et al's[1] ones were the paper we related to the most. It present CNN model for recognition of genre, facial expression, musical instruments and era out of the Million Song Data by using a 2D convolution model woth recurrent layer and connected both of them together for classification of things. The model of our work will be described later in this report. And for Neural Network layer we got referred by the work of Vaibhavi, Macharla & Radha Krishna, Pisipati. (2021). Music Genre Classification using Neural Networks with Data Augmentation[2] which make inspired the approach which will be describe in further section.

1.2 Objective of the project

To develop the model which will be able to recognize the music genre and classify it out which including the understand of data manipulation and data visualization and model training and data augmentation. And to have understanding of neural network and be able to approach the idea of neural network for the further project. To have a better understanding of the various layers of image classification that assist in image recognition. And to be able to develop music classification to the speech recognition and challenge in this field of study.

1.3 Problem Statement

We are lack of data from the selected dataset. Thus, we decided to manipulate the dataset by our own with data augmentation. The validation of the model is needed to be perfect fit model in order to classify the given test audio file precisely.

2 Approach

In this section, we will discuss about the dataset we used and how we manipulated the dataset to get the better result of project.

2.1 Selection of Dataset

As far as we trying to look up for the suitable dataset in this project, we came out of 3 different datasets that will suit this project. We decide to select GTZAN dataset – Music Genre Classification [3] which provided 1000 songs of 10 difference genres of 30 seconds each. Thus, in total 100 songs each genre. Which seem to be not enough after we have been trying to build the model. With the lacks of dataset, the 1st model that we have been trained using NN layers is Underfitting problem. Thus, we decide to increase the number of audio dataset by concatenated the dataset from 30 sec of 100 songs each genre into 3sec of 1000 songs each genre. Thus, instead of 1000 files, we got 10,000 files to be manipulated and trained in our model. Both dataset of original and manipulated by us one will be provided in the zip file. And with some problem, when we are developing the CNNs model which will be our 3rd model in this project. It come out that MFCC could not classify amplitude of the genres by MFCC feature thus we decide to cut it out of the list of classification model with CNN approach.

2.1 Mel Spectrogram Generation

Every sound file got turned into a spectrogram, which is a graphical depiction of the frequency spectrum across time. A typical spectrogram is indeed the square of the magnitude of the sound signal's short term Fourier transform (STFT). This standard spectrogram is compressed using the Mel scale to turn the sound waves to something an individual could recognize. We used the MFCC feature from librosa python library to perform transformations of music file into spectrogram The most vital parameters employed in the transformation are – periodic length that indicates the window of your time to perform Fourier rework on and hop length which is that the variety of samples between consecutive frames.

As each music genre have their own musical amplitude thus we would be able to recognize it and classify it out. But as the 30 seconds file are not enough to make a good model thus we do use the 3 seconds to classify the Figure NO.1,2 will show you the different in some music genres in comparison and Figure No.3,4 will compare you the 30 sec spectrogram amplitude with 3 sec spectrogram amplitude.

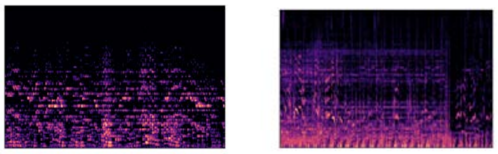


Figure No.1: left is classical genre and left is pop genres in comparison of difference of genre spectrogram amplitude

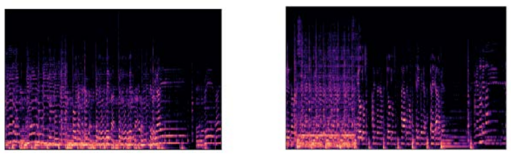


Figure No.2: left is metal genre and left is rock genres in comparison of difference of genre spectrogram amplitude

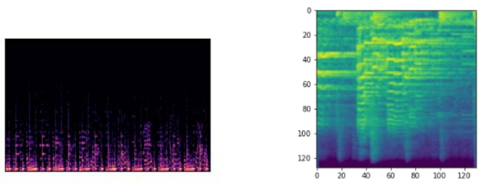


Figure No.3: left is blues genre 30sec and left is blues genres 3sec in comparison of difference of genre spectrogram amplitude in smaller time frame.

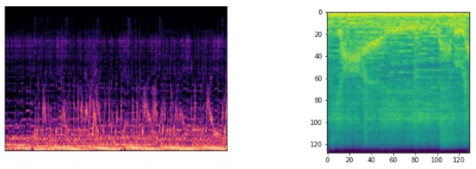


Figure No.4: left is rock genre 30sec and left is rock genres 3sec in comparison of difference of genre spectrogram amplitude in smaller time frame.

3 Implementation

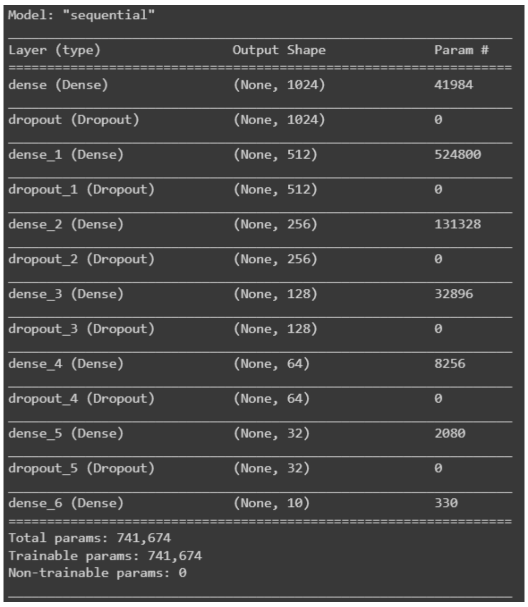
3.1Requirement:

- Languages:
  Python
- Operating System:
  Window10
- Library Packages:
  Tensorflow : Machine Learning Model training
  tensorflow.keras : Machine Learning Model training
  Pandas : Data Manipulation and Calculation
  Scipy : Scientist Related formula
  Pydub : To concatenate audio file
  Pydot :
  IPython.Display : Display result
  Matplotlib : Plotting of Graph
  Numpy : Data Manipulation
  Os : Operating perform
  Librosa: MFCC feature
- Software used:
  Anaconda Prom
  Jupyter Notebook
  Kaggle
  Google Colab
  Google Drive
  Google API

3.1 ANNs Model with 30 sec music datasets

With the basic knowledge of neural network, we use neural network layer approach as the baseline model approach we feed the model with the input of 30sec amplitude data from GTZAN dataset for the music classification which contain "filename,length,chroma_stft_mean, chroma_stft_var
rms_mean,rms_var,spectral_centroid_mean, spectral_centroid_var, spectral_bandwidth_mean, spectral_bandwidth_var, rolloff_mean, rolloff_var, zero_crossing_rate_mean, zero_crossing_rate_var, harmony_mean, harmony_var, perceptr_mean, perceptr_var, tempo, mfcc1_meanm, mfcc1_var, mfcc2_mean, mfcc2_varmfcc3_mean,mfcc3_var, mfcc4_mean, mfcc4_var, mfcc5_mean, mfcc5_var mfcc6_mean mfcc6_var
mfcc7_mean mfcc7_var
mfcc8_mean mfcc8_var
mfcc9_mean mfcc9_var
mfcc10_mean mfcc10_var
mfcc11_mean mfcc11_var
mfcc12_mean mfcc12_var
mfcc13_mean mfcc13_var
mfcc14_mean mfcc14_var
mfcc15_mean mfcc15_var
mfcc16_mean mfcc16_var
mfcc17_mean mfcc17_var
mfcc18_mean mfcc18_var
mfcc19_mean mfcc19_var
mfcc20_mean mfcc20_var label"

To the input layer of 1024 batch nums and 5 hidden layers of 512,256,128,64,32 each layer and drop out (0.3) of data each layer and output data with softmax activation as the final layer. We trained the model 150 times with num_batch_size of 32. With data input of X train and test 40 and Y are 10. The figure 5 will show summary of model.



```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 1024)              41984
_________________________________________________________________
dropout (Dropout)            (None, 1024)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               524800
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_2 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 128)               32896
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 64)                8256
_________________________________________________________________
dropout_4 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_5 (Dense)              (None, 32)                2080
_________________________________________________________________
dropout_5 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_6 (Dense)              (None, 10)                330
=================================================================
Total params: 741,674
Trainable params: 741,674
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: summary of 1st Model training

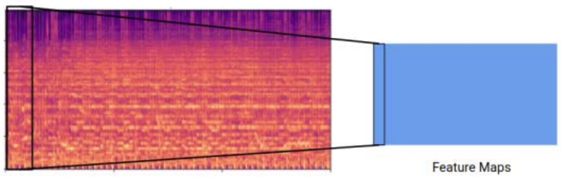
3.2 ANNs Model with 3 sec music datasets

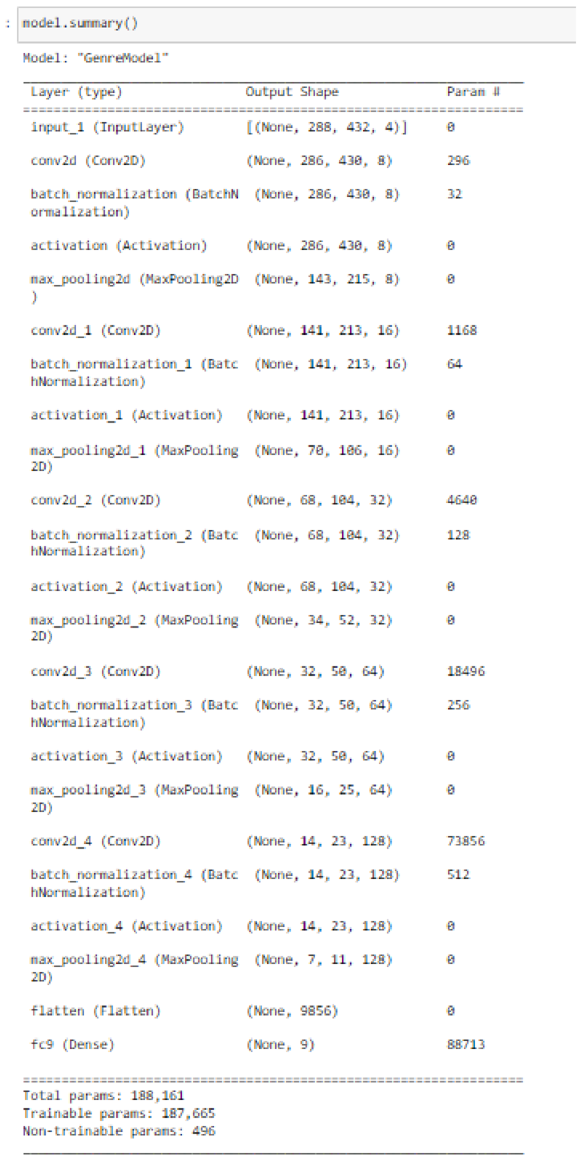
In this model , we use the same approach as previous model with difference in datasets which we used dataset of 3sec instead as the result we got in previous model is not good and will be described in result. As we using the dataset of 3 seconds, we increase the amount of data. Thus, we can get more data to train. That might get us the more accuracy. But, the other set of setting are exactly the same. And Figure 6 will show the summary of this model.

Figure 6: summary of 2nd Model training

3.3 Conventional Neural Networks Model with 3 sec music datasets

In previous model Baseline Model with 3 sec music datasets, the result also come out as not as we expect. Thus, we decided to develop another model based on CNN model which use Mel Spectrogram that we used MFCC feature from librosa to exact 3 second file of audio provided in dataset which it will provide image file and use it with CNNs as images file to recognize the pattern out of the Mel spectrogram with instead of 10,000 files we used 9,000 files excluding Jazz out of the list. We used Conv2 to implement CNNs layer model with activation of ('relu') with Figure 7 will show the summary of model.





```
: model.summary()

Model: "GenreModel"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 288, 432, 4)]     0

 conv2d (Conv2D)             (None, 286, 430, 8)       296

 batch_normalization (BatchN  (None, 286, 430, 8)      32
 ormalization)

 activation (Activation)     (None, 286, 430, 8)       0

 max_pooling2d (MaxPooling2D  (None, 143, 215, 8)      0
 )

 conv2d_1 (Conv2D)           (None, 141, 213, 16)      1168

 batch_normalization_1 (Batc  (None, 141, 213, 16)     64
 hNormalization)

 activation_1 (Activation)   (None, 141, 213, 16)      0

 max_pooling2d_1 (MaxPooling  (None, 70, 106, 16)      0
 2D)

 conv2d_2 (Conv2D)           (None, 68, 104, 32)       4640

 batch_normalization_2 (Batc  (None, 68, 104, 32)      128
 hNormalization)

 activation_2 (Activation)   (None, 68, 104, 32)       0

 max_pooling2d_2 (MaxPooling  (None, 34, 52, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 32, 50, 64)        18496

 batch_normalization_3 (Batc  (None, 32, 50, 64)       256
 hNormalization)

 activation_3 (Activation)   (None, 32, 50, 64)        0

 max_pooling2d_3 (MaxPooling  (None, 16, 25, 64)       0
 2D)

 conv2d_4 (Conv2D)           (None, 14, 23, 128)       73856

 batch_normalization_4 (Batc  (None, 14, 23, 128)      512
 hNormalization)

 activation_4 (Activation)   (None, 14, 23, 128)       0

 max_pooling2d_4 (MaxPooling  (None, 7, 11, 128)       0
 2D)

 flatten (Flatten)           (None, 9856)              0

 fc9 (Dense)                 (None, 9)                 88713

=================================================================
Total params: 188,161
Trainable params: 187,665
Non-trainable params: 496
_________________________________________________________________
```

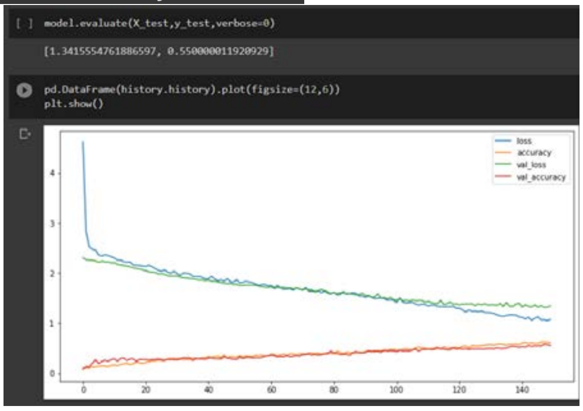
Figure 7: 3rd Model Summary.

4 Result and Outcome of the project

4.1 ANNs Model with 30 sec music datasets result:

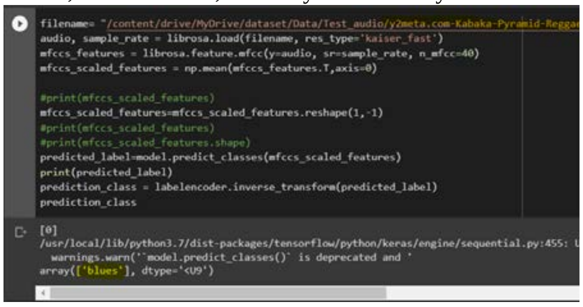
With the model , we trained as the dataset of 30sec music dataset which contains 1000 dataset of 10 difference genres. But unfortunately, the dataset was not enough to train the model as the Figure 8 will show you <u>that model is underfitting</u> as we are lacking of dataset which can teach machine to learn to classification the song as the Figure 9. Which we use sample test file as Reggae song but model classify it as the Blues genres song.

```
Result: loss: 1.0507 - accuracy:
0.6183 - val_loss: 1.3152 -
val_accuracy: 0.5700
```



```
[ ] model.evaluate(X_test,y_test,verbose=0)

    [1.3415554761886597, 0.550000011920929]

pd.DataFrame(history.history).plot(figsize=(12,6))
plt.show()
```

Figure 8: Model Evaluation of 1st model and Graph of Value loss, Validation loss, Accuracy and Accuracy Validation.



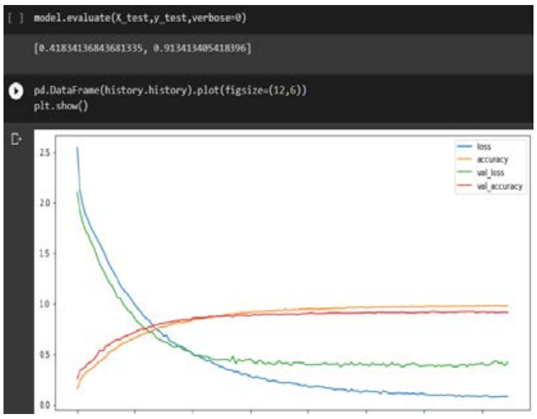
```
filename= "/content/drive/MyDrive/dataset/Data/Test_audio/y2meta.com-Kabaka-Pyramid-Reggae
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)

#print(mfccs_scaled_features)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
#print(mfccs_scaled_features)
#print(mfccs_scaled_features.shape)
predicted_label=model.predict_classes(mfccs_scaled_features)
print(predicted_label)
prediction_class = labelencoder.inverse_transform(predicted_label)
prediction_class

[0]
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:455: U
  warnings.warn('`model.predict_classes()` is deprecated and '
array(['blues'], dtype='<U9')
```

Figure 9: We give the test sample audio file as reggae but result of prediction come out to be “Blues”

4.2 ANNs Model with 3 sec music datasets result:

As the problem state in 1st model, we are lack of dataset. Thus, we decide to manipulate the dataset by ourselves by concatenate dataset of 30sec of 900 files into 3sec of 9,000 files(excluding Jazz out). And we trained the model by inputting the data of spectrogram amplitude and mean ETC,. Figure 10 and 11 will show you the result of 2nd model.

The result come out to be Overfitting as it too good to be real because the dataset is not that much of various. Thus, we decide the augment the dataset by ourselves to implement it in 3rd model.

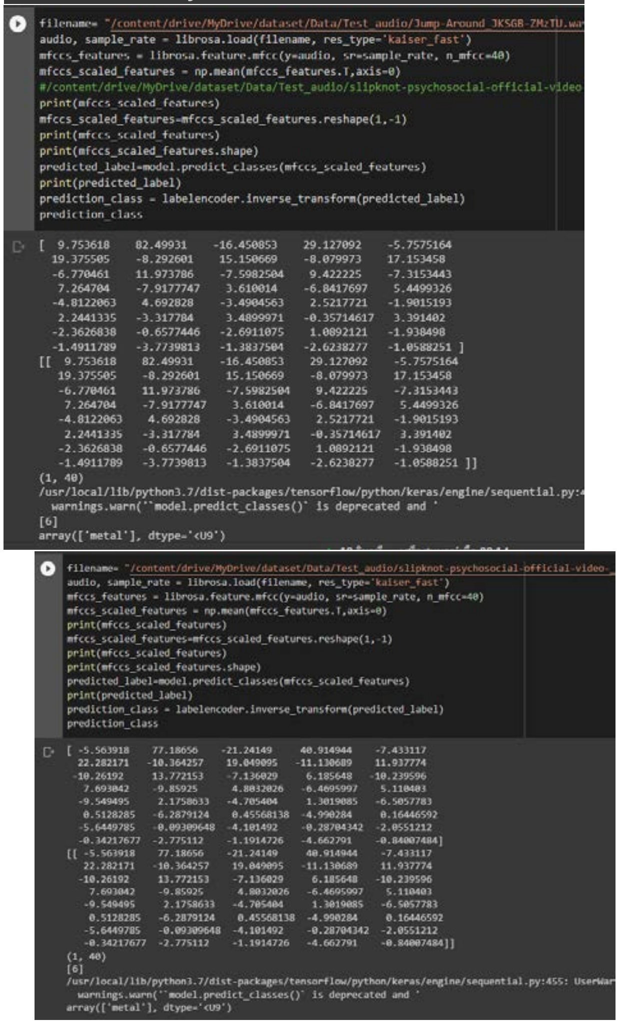


```
[ ] model.evaluate(X_test,y_test,verbose=0)

    [0.41834136843681335, 0.913413405418396]

pd.DataFrame(history.history).plot(figsize=(12,6))
plt.show()
```

Figure 10: Model Evaluation of 2st model and Graph of Value loss, Validation loss, Accuracy and Accuracy Validation.

As the result, the value loss trend to be decreased into almost 0 value but validation lost is stable at 0.37425 which we could conclude that the model is overfitting.

```
Result: loss: 0.0792 - accuracy:
0.9779 - val_loss: 0.3725 -
val_accuracy: 0.9154”
```



```
filename= "/content/drive/MyDrive/dataset/Data/Test_audio/Jump-Around_JKSGB-ZMzTU.wa
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
#/content/drive/MyDrive/dataset/Data/Test_audio/slipknot-psychosocial-official-video
print(mfccs_scaled_features)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
print(mfccs_scaled_features)
print(mfccs_scaled_features.shape)
predicted_label=model.predict_classes(mfccs_scaled_features)
print(predicted_label)
prediction_class = labelencoder.inverse_transform(predicted_label)
prediction_class

[  9.753618    82.49931    -16.450853    29.127092    -5.7575164
  19.375505    -8.292601    15.150669    -8.079973    17.153458
  -6.770461    11.973786    -7.5982504    9.422225    -7.3153443
   7.264704    -7.9177747    3.610014    -6.8417697    5.4499326
  -4.8122063    4.692828    -3.4904563    2.5217721   -1.9015193
   2.2441335   -3.317784     3.4899971   -0.35714617   3.391402
  -2.3626838   -0.6577446   -2.6911075    1.0892121   -1.938498
  -1.4911789   -3.7739813   -1.3837504   -2.6238277   -1.0588251 ]
[[  9.753618    82.49931    -16.450853    29.127092    -5.7575164
   19.375505    -8.292601    15.150669    -8.079973    17.153458
   -6.770461    11.973786    -7.5982504    9.422225    -7.3153443
    7.264704    -7.9177747    3.610014    -6.8417697    5.4499326
   -4.8122063    4.692828    -3.4904563    2.5217721   -1.9015193
    2.2441335   -3.317784     3.4899971   -0.35714617   3.391402
   -2.3626838   -0.6577446   -2.6911075    1.0892121   -1.938498
   -1.4911789   -3.7739813   -1.3837504   -2.6238277   -1.0588251 ]]
(1, 40)
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:4
  warnings.warn('`model.predict_classes()` is deprecated and '
[6]
array(['metal'], dtype='<U9')
```

```
filename= "/content/drive/MyDrive/dataset/Data/Test_audio/slipknot-psychosocial-official-video-
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
print(mfccs_scaled_features)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
print(mfccs_scaled_features)
print(mfccs_scaled_features.shape)
predicted_label=model.predict_classes(mfccs_scaled_features)
print(predicted_label)
prediction_class = labelencoder.inverse_transform(predicted_label)
prediction_class

[ -5.563918    77.18656    -21.24149     40.914944    -7.433117
  22.282171   -10.364257    19.049095   -11.130689    11.937774
 -10.26192     13.772153    -7.136029     6.185648   -10.239596
   7.693042    -9.85925      4.8832026   -6.4695997    5.110403
  -9.549495     2.1758633   -4.705404     1.3019085   -6.5057783
   0.5128285   -6.2879124    0.45568138  -4.990284     0.16446592
  -5.6449785   -0.09309648  -4.101492    -0.28704342  -2.0551212
  -0.34217677  -2.775112    -1.1914726   -4.662791    -0.84007484]
[[ -5.563918    77.18656    -21.24149     40.914944    -7.433117
   22.282171   -10.364257    19.049095   -11.130689    11.937774
  -10.26192     13.772153    -7.136029     6.185648   -10.239596
    7.693042    -9.85925      4.8832026   -6.4695997    5.110403
   -9.549495     2.1758633   -4.705404     1.3019085   -6.5057783
    0.5128285   -6.2879124    0.45568138  -4.990284     0.16446592
   -5.6449785   -0.09309648  -4.101492    -0.28704342  -2.0551212
   -0.34217677  -2.775112    -1.1914726   -4.662791    -0.84007484]]
(1, 40)
[6]
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential.py:455: UserWar
  warnings.warn('`model.predict_classes()` is deprecated and '
array(['metal'], dtype='<U9')
```

Figure 11: We give the test sample audio file as hip-hop and metal but result of prediction come out to be “Metal” only

4.3 Conventional Neural Networks Model with 3 sec music datasets:

As trained the 2nd model, the result of the model come out to be overfitting as the quality of data is not various and good enough to train the model. Thus, this idea inspired us to the last model of our project which we will augment the data by ourself by converting dataset of 3sec audio file into the spectrogram amplitude image and use CNNs model to recognize it as the picture and find the pattern of the spectrogram and figure 12,13 will show the model accuracy and loss and the result of prediction of 3rd model. And it come to the result of perfectly fit.

```
Result: loss: 0.0076 - accuracy: 1
.0000 - get_f1: 1.0000 - val_loss:
 0.0064 - val_accuracy: 1.0000
```

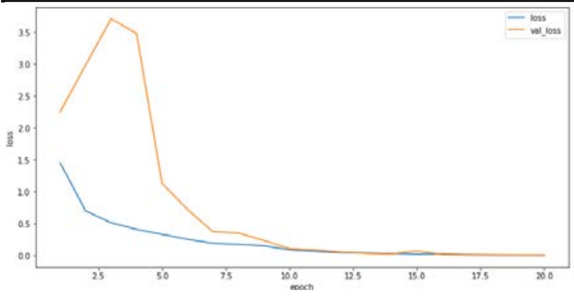


Figure 12: The value and validation loss both trend to be 0 show the fitness of the model

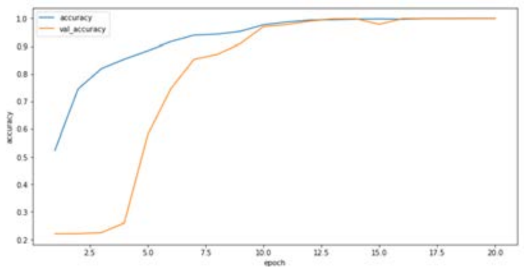
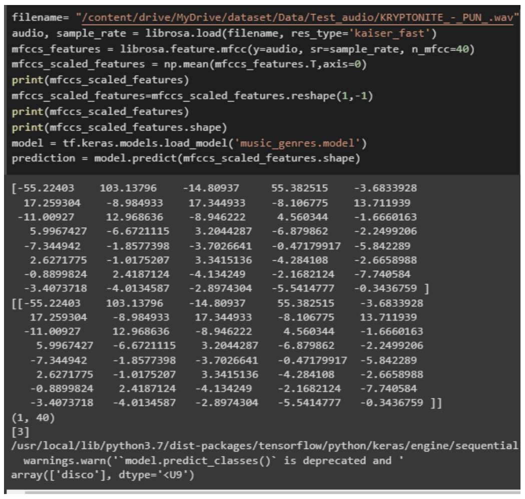
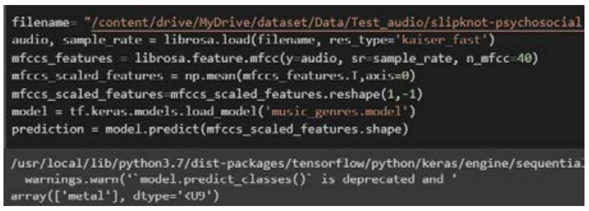


Figure 13: The model and validation accuracy both trends are 1.00 show the accuracy of the model



```
filename= "/content/drive/MyDrive/dataset/Data/Test_audio/KRYPTONITE_-_PUN_.wav"
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
print(mfccs_scaled_features)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
print(mfccs_scaled_features)
print(mfccs_scaled_features.shape)
model = tf.keras.models.load_model('music_genres.model')
prediction = model.predict(mfccs_scaled_features.shape)

[-55.22403    103.13796    -14.80937     55.382515    -3.6833928
  17.259304    -8.984933    17.344933    -8.106775    13.711939
 -11.00927     12.968636    -8.946222     4.560344    -1.6660163
   5.9967427   -6.6721115    3.2044287   -6.879862    -2.2499206
  -7.344942    -1.8577398   -3.7026641   -0.47179917  -5.842289
   2.6271775   -1.0175207    3.3415136   -4.284108    -2.6658988
  -0.8899824    2.4187124   -4.134249    -2.1682124   -7.740584
  -3.4073718   -4.0134587   -2.8974304   -5.5414777   -0.3436759 ]
[[-55.22403    103.13796    -14.80937     55.382515    -3.6833928
   17.259304    -8.984933    17.344933    -8.106775    13.711939
  -11.00927     12.968636    -8.946222     4.560344    -1.6660163
    5.9967427   -6.6721115    3.2044287   -6.879862    -2.2499206
   -7.344942    -1.8577398   -3.7026641   -0.47179917  -5.842289
    2.6271775   -1.0175207    3.3415136   -4.284108    -2.6658988
   -0.8899824    2.4187124   -4.134249    -2.1682124   -7.740584
   -3.4073718   -4.0134587   -2.8974304   -5.5414777   -0.3436759 ]]
(1, 40)
[3]
/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequential
  warnings.warn('`model.predict_classes()` is deprecated and '
array(['disco'], dtype='<U9')
```

Figure 14: Model predict the disco song correctly



```
filename= "/content/drive/MyDrive/dataset/Data/Test_audio/slipknot-psychosocial
audio, sample_rate = librosa.load(filename, res_type='kaiser_fast')
mfccs_features = librosa.feature.mfcc(y=audio, sr=sample_rate, n_mfcc=40)
mfccs_scaled_features = np.mean(mfccs_features.T,axis=0)
mfccs_scaled_features=mfccs_scaled_features.reshape(1,-1)
model = tf.keras.models.load_model('music_genres.model')
prediction = model.predict(mfccs_scaled_features.shape)

/usr/local/lib/python3.7/dist-packages/tensorflow/python/keras/engine/sequentia
  warnings.warn('`model.predict_classes()` is deprecated and '
array(['metal'], dtype='<U9')
```

Figure 15: Model predict Metal Song correctly

5 Conclusion

As far as the 3 models trained, the result come out that the CNNs model one is the most effective model which is perfect fit which is different from other 2 which are Underfitting and Overfitting. And we can conclude our project as successful as we could accomplish our objective which are training the model to be able to classification the genre of the song and understanding the machine learning model with neural network layer and understand its application in various way and know the variety of implementation it into many more real worlds to solve various of problem. And understand the algorithm behind of success of Voice Ai as Alexa of Amazon, Siri of Apple, and Hey google of google.

6 Future Scope

We could implement the project into speech recognition with NLP implementation to recognition of speech or sound or song in various way of sound industries and we could find the effective way to implement the recognition method to be easier and less value consumption in the future scope. And we could also implement it into the android home bot to answer or help us with only a sound command as the big company success

References

[1]..Keunwoo Choi, George Fazekas, Mark Sandler, Kyunghyun Cho. Convolutional Recurrent Neural Networks for Music Classification. ArXiv 1609.04243.

[2]. Vaibhavi, Macharla & Radha Krishna, Pisipati. (2021). Music Genre Classification using Neural Networks with Data Augmentation A Make in India Creation. 1. 21-37.

[3]. GTZAN Dataset – Music Genre Classification

[4]. Priya Dwivedi, Using CNNs and RNNs for Music Genre Recognition, Dec 14, 2018

[5]. Bob L. Sturm. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. arXiv:1306.1461v2

[6]. Keras Visualization: https://github.com/raghakot/keras-vis