iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** X **Month of publication:** October 2021

DOI: https://doi.org/10.22214/ijraset.2021.38675

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

Covid Classification Using Audio Data

Adwait Patil
Computer Engineering

Abstract: ***Coronavirus outbreak has affected the entire world adversely this project has been developed in order to help common masses diagnose their chances of been covid positive just by using coughing sound and basic patient data. Audio classification is one of the most interesting applications of deep learning. Similar to image data audio data is also stored in form of bits and to understand and analyze this audio data we have used Mel frequency cepstral coefficients (MFCCs) which makes it possible to feed the audio to our neural network. In this project we have used Coughvid a crowdsource dataset consisting of 27000 audio files and metadata of same amount of patients. In this project we have used a 1D Convolutional Neural Network (CNN) to process the audio and metadata. Future scope for this project will be a model that rates how likely it is that a person is infected instead of binary classification.***

Keywords: Audio classification, Mel frequency cepstral coefficients, Convolutional neural network, deep learning, Coughvid

I. INTRODUCTION

Coronavirus has been spreading widely throughout the world and one of the major causes for this is inaccessible testing centers or large crowds at testing centers which made the masses apathetic of covid tests. This project aims to give an early assessment for covid for a patient provided their coughing audio and relevant historic health data. This would give them an idea about their situation and can help them decide on visiting a testing center for further confirmation. The dataset used is Coughvid a crowdsource dataset created to help researchers get an updated dataset to work on coronavirus related research. Our model uses a 1D CNN to process audio and metadata and gives prediction whether a person is healthy or infected. The audio files were extracted into a readable format and converted using MFCCs to extract the features of audio and then they were fed to the model in order to train it and get predictions.

II. PROPOSED METHODOLOGY

The audio files were present as tar files in the provided dataset. The first challenge was to extract the audio files in a readable format.

A. Extraction

The glob module was used to get the locations of tar files as a list and then further these files were extracted using the librosa module and its mcffs feature to convert them into readable format. The path generated by the glob module was passed to the librosa's load () method to load the audio file in memory and then extract its features using the mcffs feature. The most important part of this extraction was to correctly map the audio with the patient this was done by providing a unique identity number to each audio file.

```
path=glob.glob('./public_dataset/public_dataset/*.csv')

path_ogg = glob.glob('./public_dataset/public_dataset/*.ogg')

metadata=pd.read_csv(path[0])
```

Fig.1 Glob Module

As seen in above figure the glob module is used to extract the path of ogg files and store them in a path_ogg variable.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue X Oct 2021- Available at www.ijraset.com

B. Exploratory Data Analysis

Once the data was in a readable format the next step was to analyze the data. There were two types of data that I had to analyze numerical patient metadata and audio data. First let's discuss the audio data. Depending on the patient's condition (Infected/Healthy) audio data was broadly of two types. The visualization of coughing voice for an infected patient showed a spikier graph with the peaks lying very close to each other.

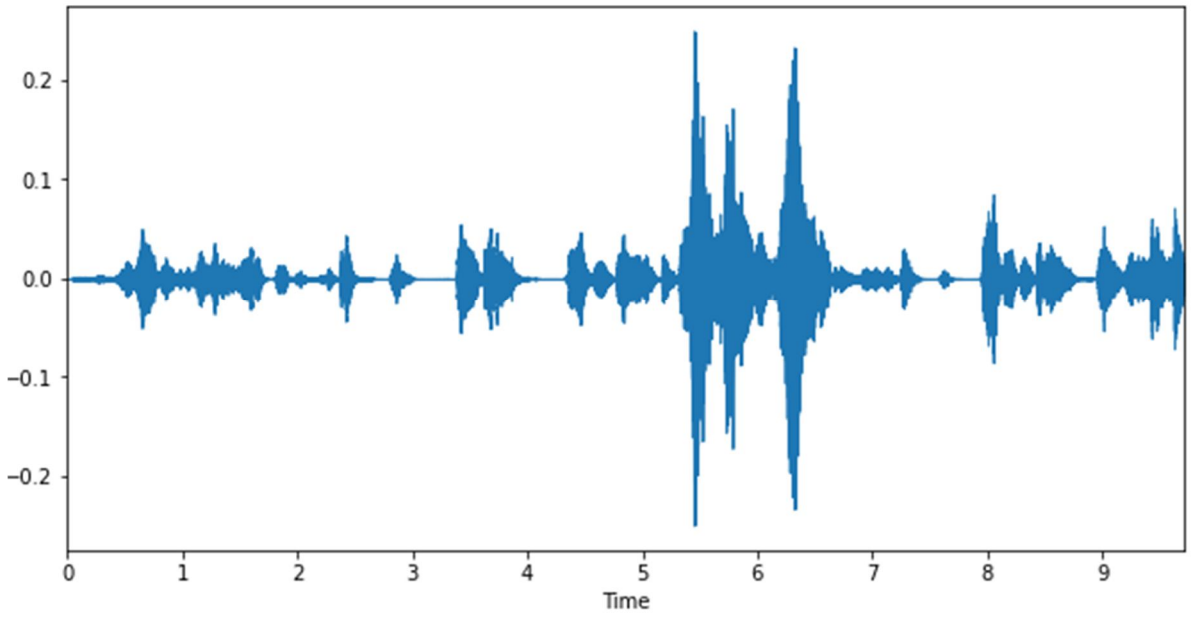


Fig.2 wave plot for Covid infected patient

Whereas the visualization for a healthy person's coughing voice showed a more clustered wave plot with peaks merging into one another.

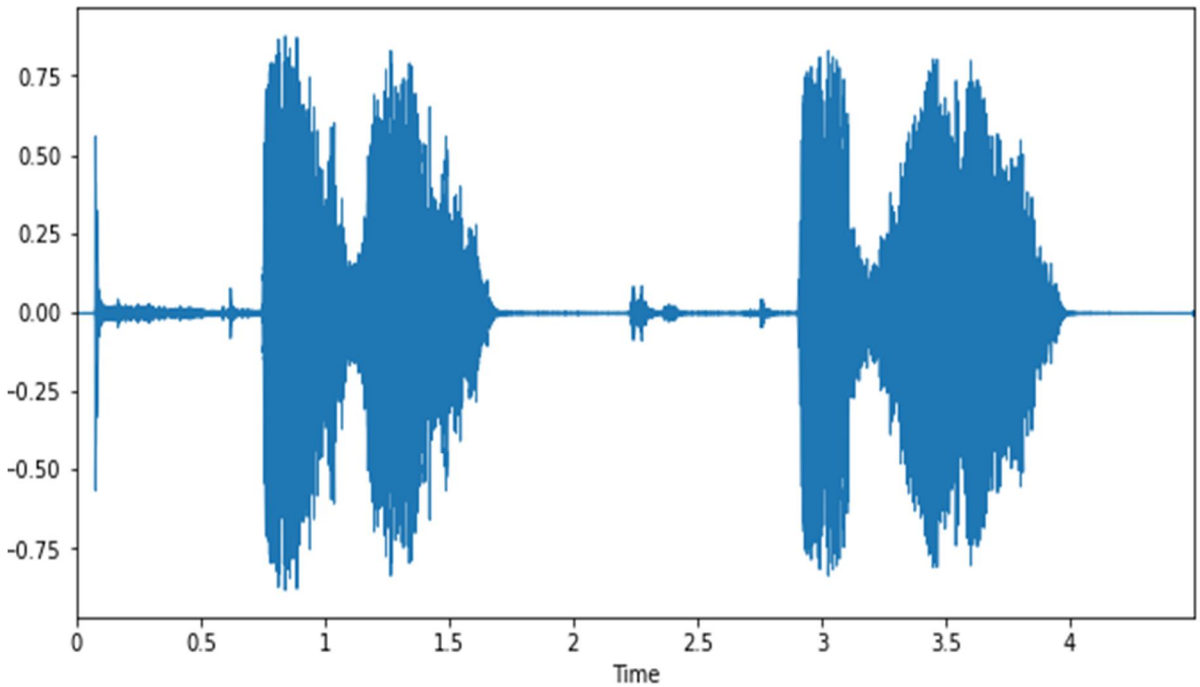


Fig.3 Wave plot for a healthy person

©IJRASET: All Rights are Reserved

Now for the metadata there were a number of columns so first I divided those into categorical and numerical columns. Then using a simple python command identified the columns with high number of null values and dropped the one's which couldn't be imputed. Furthermore to get idea about which numerical attributes were highly correlated to the target I created pair plot and heatmap.

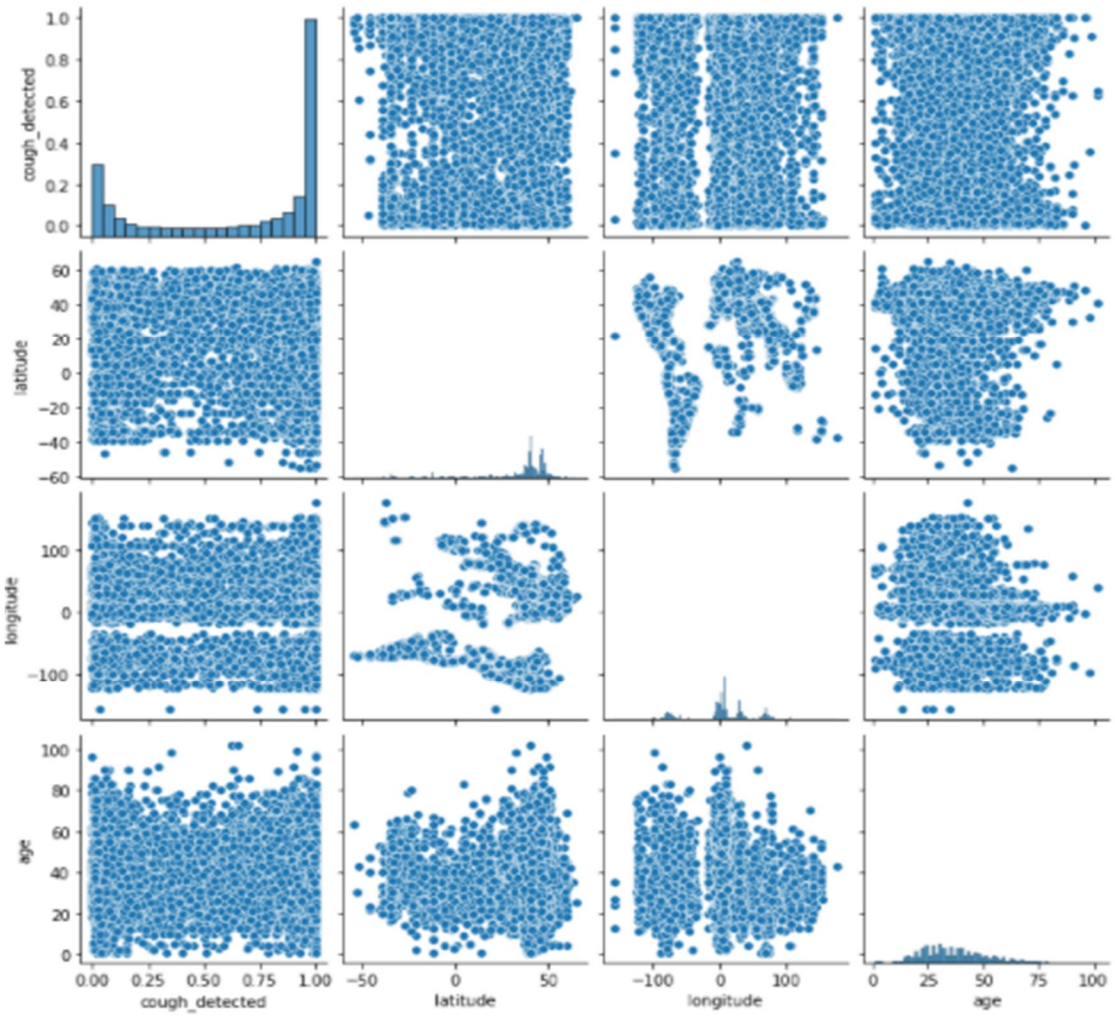


Fig.4 Pair plot

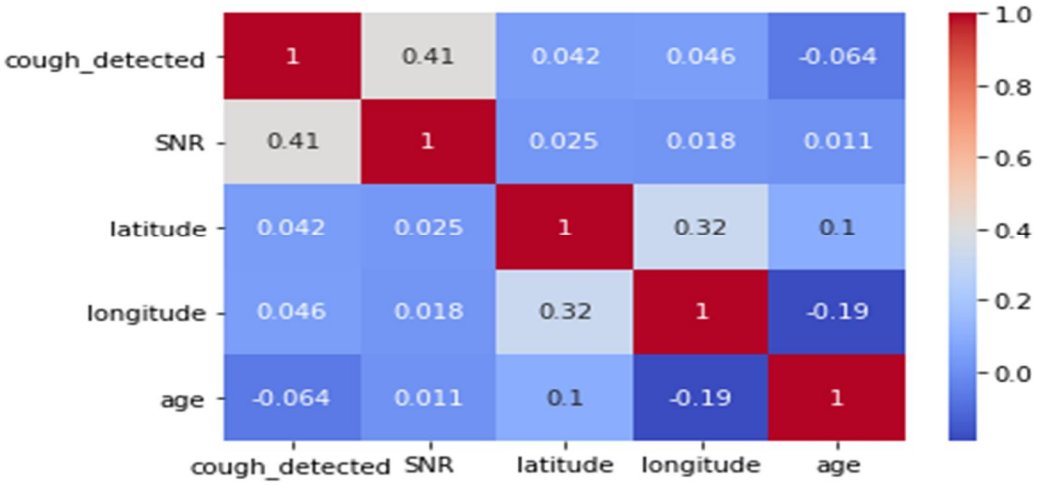


Fig.5 Heat map

©IJRASET: All Rights are Reserved

C. Feature Engineering

After exploratory data analysis I did feature engineering to impute features with missing data and the scaled those features so that they can be then fed to the model. The features that were imputed are latitude, longitude , age, gender, respiratory condition , fever-muscle pain and status, this features were highly correlated to the target feature and hence could not be dropped.

III. MEL FREQUENCY CEPSTRAL COEFFICIENTS

The feature engineering for audio data was quite different than the metadata, for that I used the mfcc method provided by the librosa library to convert the audio data into features that can be fed to the model. Let's first learn about mfcc and how it creates features from audio data.

Mel-frequency cepstral coefficients (MFCCs) are coefficients that collectively make up an MFC.[1] They are derived from a type of cepstral representation of the audio clip (a nonlinear "spectrum-of-a-spectrum"). The difference between the cepstrum and the mel-frequency cepstrum is that in the MFC, the frequency bands are equally spaced on the Mel scale, which approximates the human auditory system's response more closely than the linearly-spaced frequency bands used in the normal spectrum. This frequency warping can allow for better representation of sound, for example, in audio compression. [1]

Using this we divided the audio data into 30 features.

We will use mfccs for getting information from audio files

```
mfccs = librosa.feature.mfcc(data,sample_rate,n_mfcc=30)
```

```
mfccs.shape
```

```
(30, 419)
```

```
#### Applying this to entire dataset of audio files
def extract_features(file):
    data,sample_rate=librosa.load(file)
    mfccs = librosa.feature.mfcc(data,sample_rate,n_mfcc=30)
    mfccs_scaled = np.mean(mfccs.T,axis=0)
    return mfccs_scaled
```

Fig.6 MFCCs code

IV. MODEL CREATION AND TRAINING

Once the feature engineering is completed the next step is selecting features and then creating model for accepting these features and generating output. As the data is multi modal we have used a 1 Dimensional (D) Convolutional Neural Network (CNN) to process the data. The model was created using the Sequential model an inbuilt model provided by Tensorflow and layers were added using the Dense and Dropout layers from Keras API.

```
#model

cnn_model = Sequential()

cnn_model.add(Conv1D(filters=32,kernel_size=(2),padding='same',input_shape=(37,1),activation='relu'))
cnn_model.add(MaxPool1D())

cnn_model.add(Conv1D(filters=32,kernel_size=(2),padding='same',input_shape=(37,1),activation='relu'))
cnn_model.add(MaxPool1D())

cnn_model.add(Conv1D(filters=64,kernel_size=(2),padding='same',input_shape=(37,1),activation='relu'))
cnn_model.add(MaxPool1D())

cnn_model.add(Flatten())

cnn_model.add(Dense(100,activation='relu'))
cnn_model.add(Dropout(0.5))
cnn_model.add(Dense(20,activation='relu'))

cnn_model.add(Dense(2,activation='sigmoid'))

#compile

cnn_model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
```

Fig. 7 Model Creation

©IJRASET: All Rights are Reserved

After this the model was trained using the training dataset and validated using the testing dataset both of which were derived from the Coughvid dataset using train_test_split from Scikit-Learn library. The kernel size was decided as 2 and the input shape (37, 1) represents the number of input features. The activation function used was 'ReLu' as it gave the best performance compared to other functions.

V. CONCLUSION

In conclusion the CNN based model gave a satisfactory precision and accuracy which is 72%. The validation loss was a bit higher than the training loss and the validation accuracy was bit lower than the training accuracy this was caused by lower number of patients with Covid symptoms in the dataset which affected the bias-variance ratio.

	Precision	Recall	F-1 Score	Support
0	0.69	0.72	0.70	4236
1	0.74	0.71	0.73	4856
Accuracy			0.71	9092
Macro avg	0.71	0.71	0.71	9092
Weighted Avg	0.72	0.71	0.71	9092

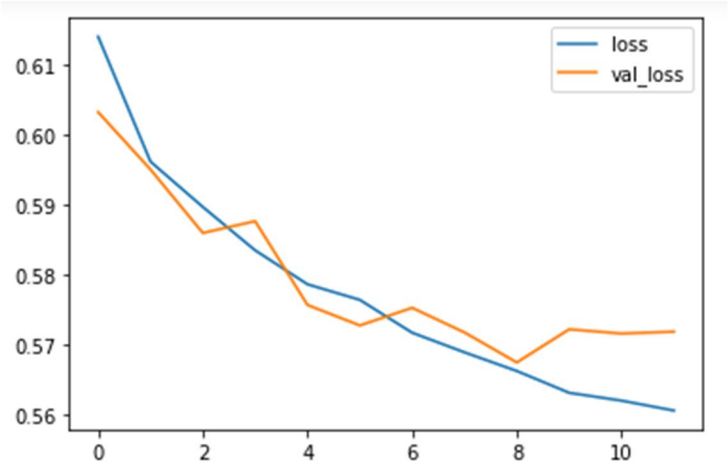


Fig.8 Loss

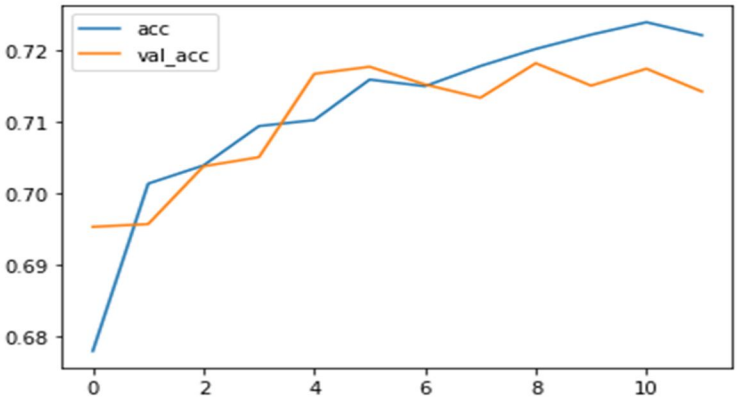
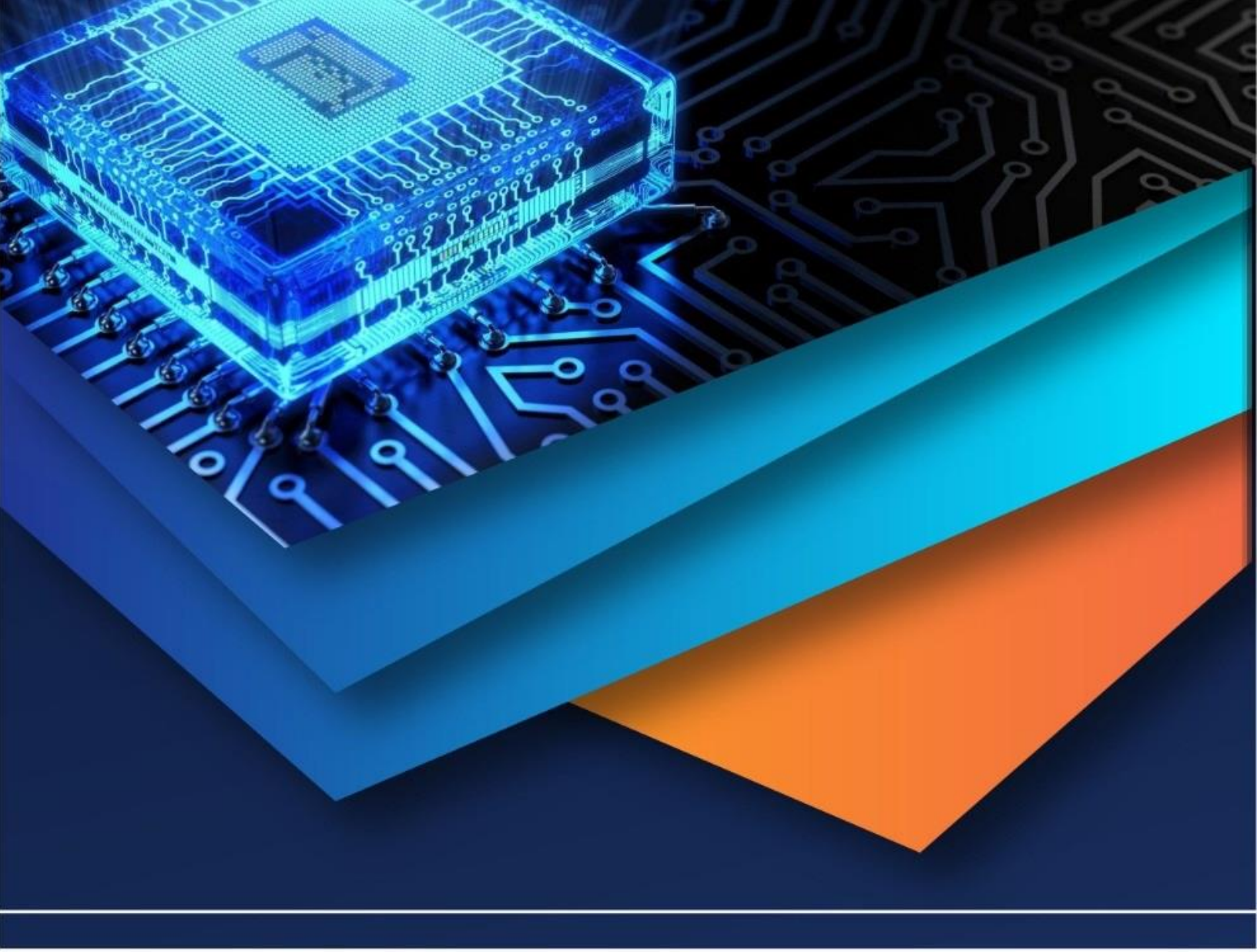


Fig. 9 Accuracy

REFERENCES

[1] En.wikipedia.org. 2021. Mel-frequency cepstrum - Wikipedia. [Online] Available at: <https://en.wikipedia.org/wiki/Mel-frequency_cepstrum>.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)