ser

May 21, 2023

Authors:

Rana Mohamed Barakat (ID:6926), Mohab Mohamed Ali (ID:7199)

0.1 Problem Statement

Speech is the most natural way of expressing ourselves as humans. It is only natural then to extend this communication medium to computer applications. We define speech emotion recognition (SER) systems as a collection of methodologies that process and classify speech signals to detect the embedded emotions. SER is not a new field, it has been around for over two decades, and has regained attention thanks to the recent advancements. We have built different models with different features extracted from audios to try to detect the embedded emotions in each audio. We used the CREMA dataset in this model which can be found here.

0.2 Imports

[880]:
```
import numpy as  np
import pandas as pd
import matplotlib.pyplot as plt
import os
import seaborn as sns
import random
import IPython.display
import librosa
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
import tensorflow as tf
import tensorflow.keras.layers as L
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2
from tensorflow.keras.regularizers import l1_l2
from tensorflow.python.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.python.keras import backend as K
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv1D, MaxPooling1D, LSTM, Dense, Dropout
```

[878]:
```
config = tf.compat.v1.ConfigProto( device_count = {'GPU': 1 , 'CPU': 6} )
sess = tf.compat.v1.Session(config=config)
K.set_session(sess)
```

0.3 Download the Dataset and Understand the Format (10 points)

We start by loading each audio as a normalized numpy array using librosa's load function. We set the duration to 2.5 seconds, and the offset to 0.3. So, 2.5 seconds will be retained from each audio, starting at 0.3 seconds. This is important to ensure that all audios have the same feature space dimensions. However, we can still skip this and retain the whole audio, then pad the feature spaces later on (We have implemented this as well).

[881]:
```
audios = []
labels = []
audio_files = os.listdir('Crema')
for file in audio_files:
    labels.append(file.split('_')[2])
    # fs,data = wavfile.read(os.path.join('Crema',file))
    data, fs = librosa.load(os.path.join('Crema',file),sr=None,duration=2.
 ↪5,offset=0.3)
    audios.append([data,fs])
```

[882]:
```
aud = []
for audio,fs in audios:
    audio = librosa.util.fix_length(audio, size=35000)
    aud.append([audio,fs])
audios=aud
```

[883]:
```
print(len(labels))
print(len(audios))
```

```
7442
7442
```

[884]:
```
audios[1000][0].shape
```

[884]: (35000,)

[885]:
```
set(labels)
```

[885]: {'ANG', 'DIS', 'FEA', 'HAP', 'NEU', 'SAD'}

[886]:
```
fs_list = [x for _,x in audios]
set(fs_list) # all files have a sample rate of 16kHz
```

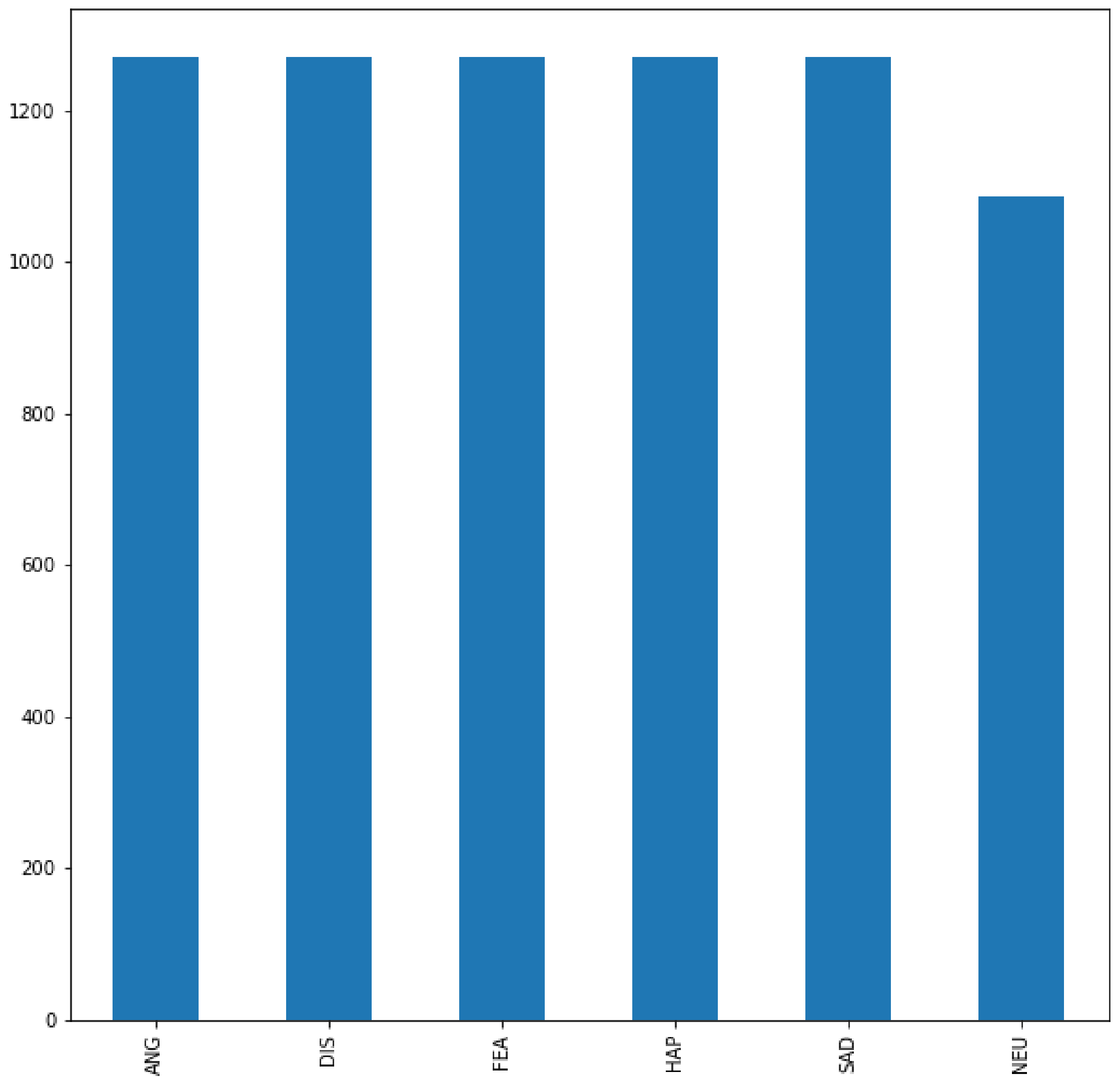
[886]: {16000}

[887]:
```python
df = pd.concat([pd.Series(audios,name='data'),pd.
 ↪Series(labels,name='label')],axis=1)
df['label'].value_counts().plot(kind='bar',figsize=(10,10))
df
```

[887]:

	data	label
0	[[0.0006713867, 0.0006713867, 0.00048828125, 0…	ANG
1	[[0.008575439, 0.008636475, 0.009094238, 0.009…	DIS
2	[[3.0517578e-05, -0.000579834, -0.0010070801, …	FEA
3	[[0.004699707, 0.0038452148, 0.0040283203, 0.0…	HAP
4	[[0.0045776367, 0.0049438477, 0.0043029785, 0…	NEU
…	…	…
7437	[[0.0020751953, 0.0020446777, 0.002166748, 0.0…	DIS
7438	[[-0.004119873, -0.0042419434, -0.004272461, -…	FEA
7439	[[-0.0035705566, -0.0033569336, -0.0028381348,…	HAP
7440	[[-0.00894165, -0.008422852, -0.008148193, -0…	NEU
7441	[[-0.0051879883, -0.0052490234, -0.0053100586,…	SAD

[7442 rows x 2 columns]



0.4 Exploring the Dataset

We implement a `load_audio()` function that plots and outputs a random audio of the desired emotion. We also plot the spectrogram of that audio using `create_spectrogram()`.

[888]:
```
emotions = {'ANG':'Angry','SAD':'Sad','NEU':'Neutral','HAP':'Happy','FEA':
 ↪'Fear','DIS':'Disgust'}
colors = {'ANG':'red','SAD':'navy','NEU':'black','HAP':'yellow','FEA':
 ↪'purple','DIS':'green'}
audioss=[]
for file in audio_files:
    data, fs = librosa.load(os.path.join('Crema',file),sr=None)
    audioss.append({'audio':data,'sr':fs,'label':file.split('_')[2]})
```

[900]:
```
def load_audio(audioss,emotion):

    # audios = list(df[df['label'] == emotion]['data'])
    audios = []
    for a in audioss:
        if a['label'] == emotion:
            audios.append(a)

    idx = random.randint(0,len(audios)-1) # choosing a random file of that␣
↪emotion
    # audio = audios[idx][0]
    audio = audios[idx]['audio']

    # fs = audios[idx][1]
    fs = audios[idx]['sr']

    duration = len(audio)/fs

    time = np.arange(0,duration,1/fs) # time vector

    plt.figure(figsize=(16,8))
    plt.plot(time,audio,color=colors[emotion])
    plt.xlabel('Time (s)')
    plt.ylabel('Amplitude')
    plt.title(emotions[emotion] + ' Waveform')
    plt.grid(color = 'green', linestyle = '--', linewidth = 0.5)
    plt.show()
    return audio, fs
```

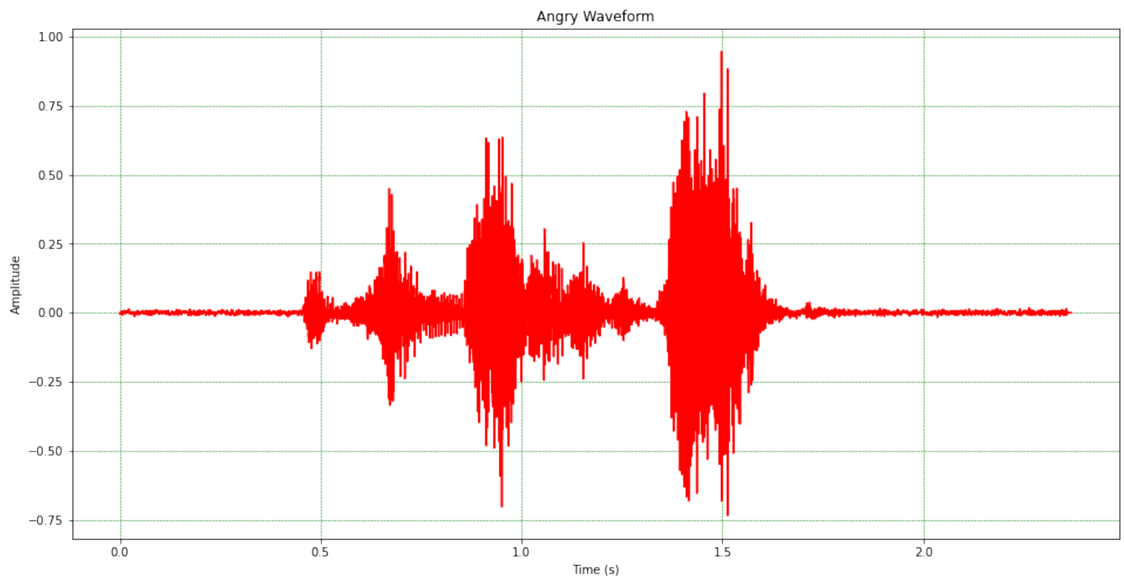
[896]:
```
audioss[0]['label']
```

[896]: 'ANG'

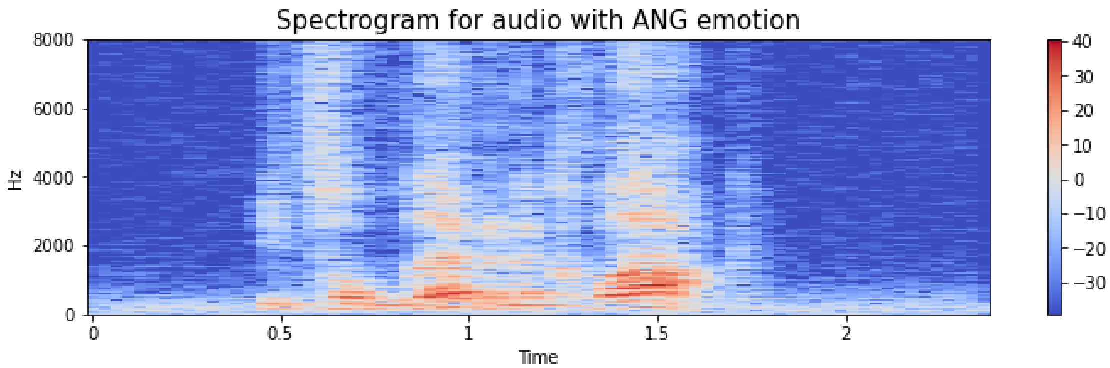
[901]:
```
def create_spectrogram(data, fs, emotion):
    # stft function converts the data into short term fourier transform
    X = librosa.stft(data)
    Xdb = librosa.amplitude_to_db(abs(X))
    plt.figure(figsize=(12, 3))
    plt.title('Spectrogram for audio with {} emotion'.format(emotion), size=15)
    librosa.display.specshow(Xdb, sr=fs, x_axis='time', y_axis='hz')
    plt.colorbar()
```

0.4.1 Angry Emotion

[902]:
```
# ANGRY
audio,fs = load_audio(audioss,'ANG')
create_spectrogram(audio,fs,'ANG')
IPython.display.Audio(audio,rate=fs)
```



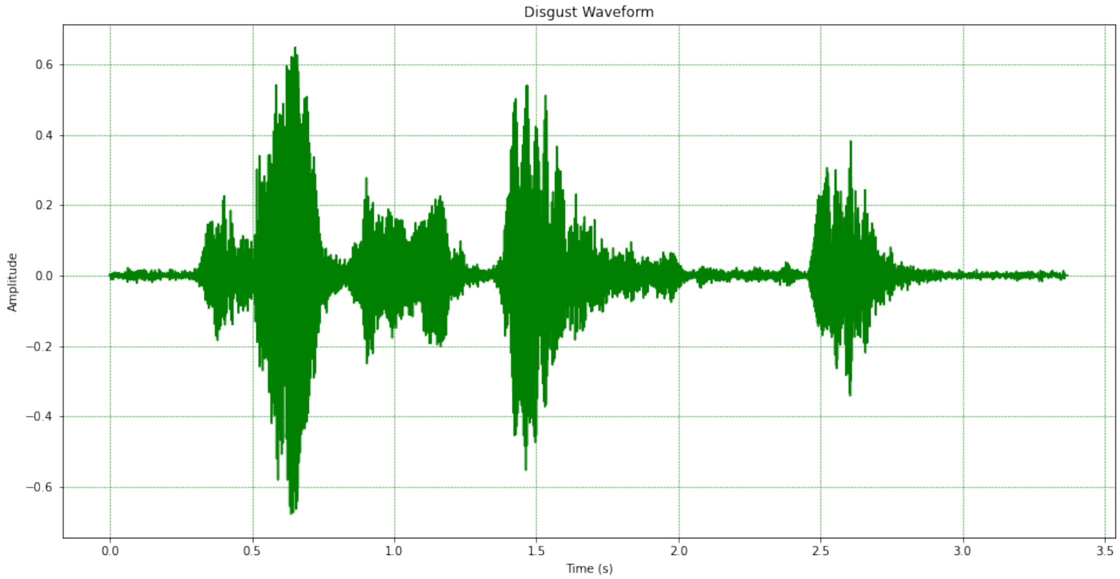
[902]: <IPython.lib.display.Audio object>



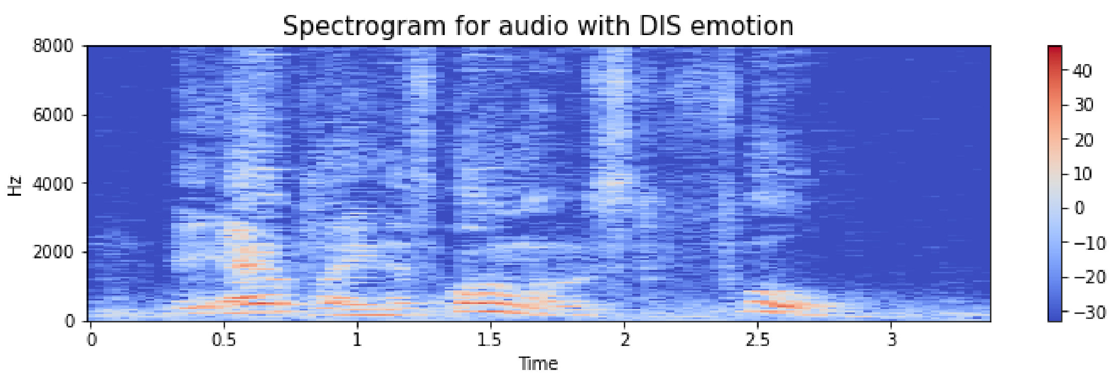
0.4.2 Disgusted Emotion

[903]:
```
# DISGUSTED
audio,fs = load_audio(audioss,'DIS')
create_spectrogram(audio,fs,'DIS')
```

```
IPython.display.Audio(audio,rate=fs)
```

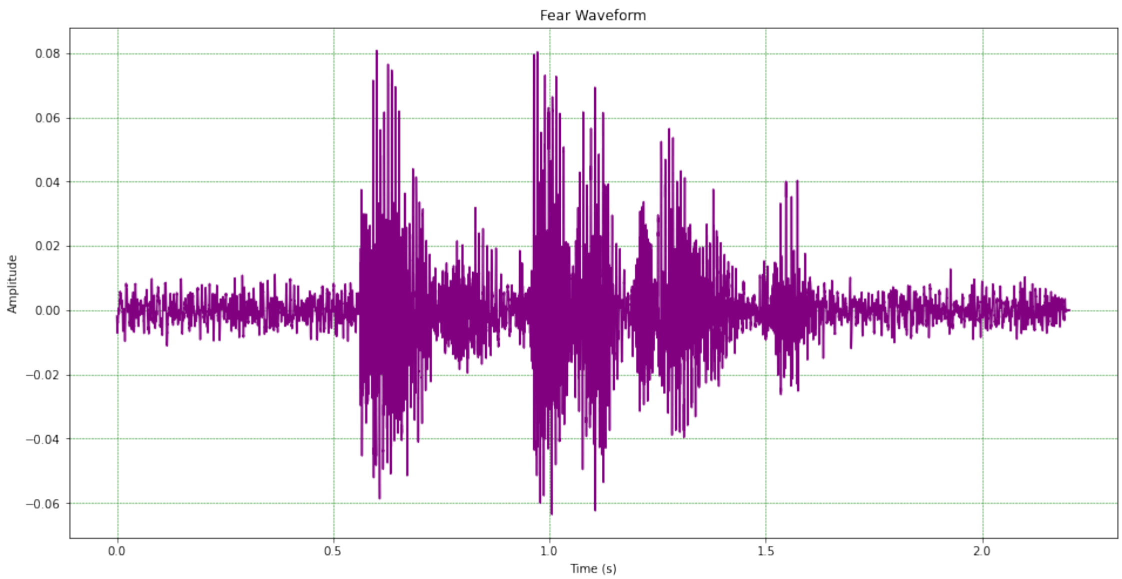


[903]: <IPython.lib.display.Audio object>

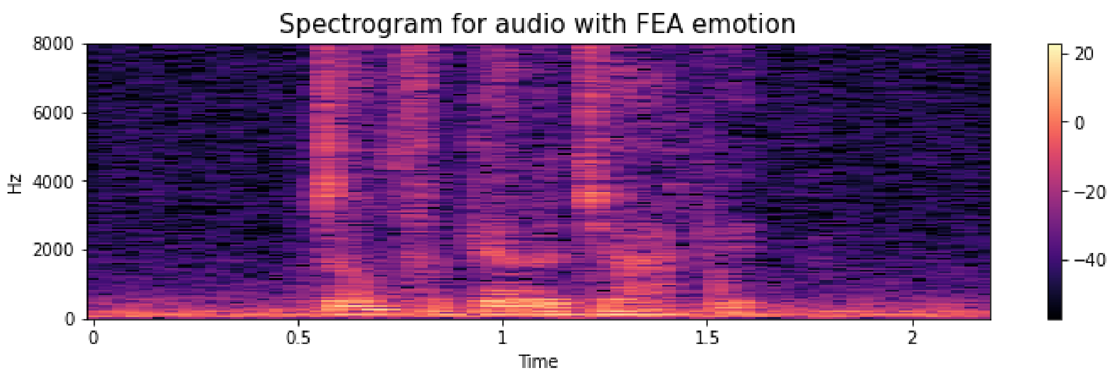


0.4.3 Fear Emotion

[904]:
```
# FEAR
audio,fs = load_audio(audioss,'FEA')
create_spectrogram(audio,fs,'FEA')
IPython.display.Audio(audio,rate=fs)
```

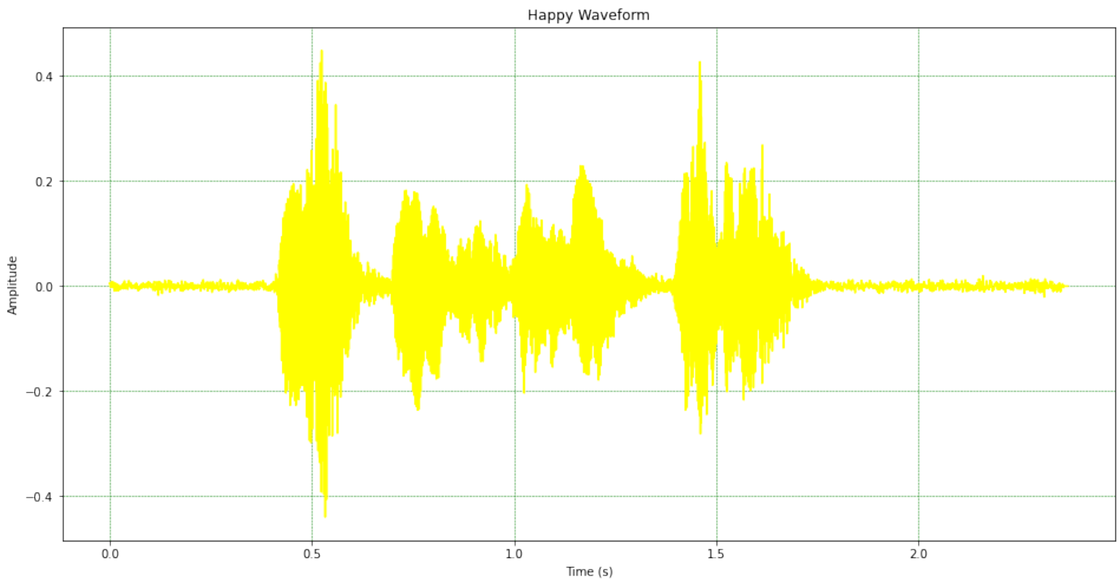


[904]: <IPython.lib.display.Audio object>

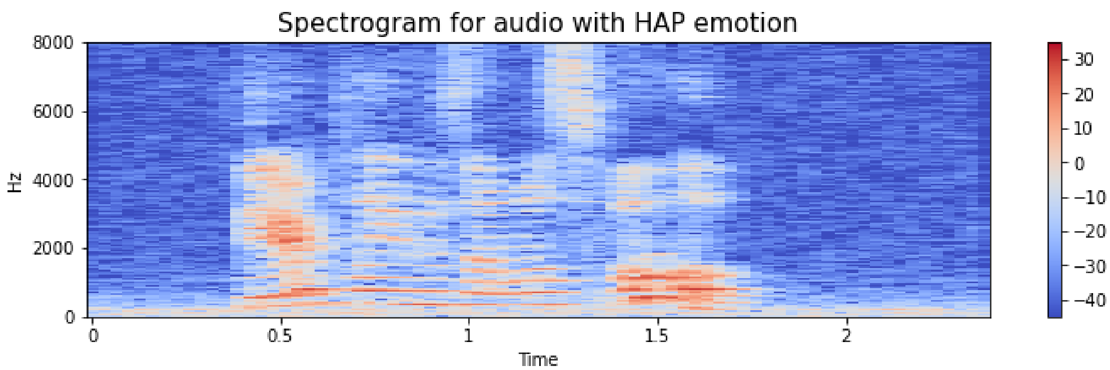


0.4.4 Happy Emotion

[906]:
```
# HAPPY
audio,fs = load_audio(audioss,'HAP')
create_spectrogram(audio,fs,'HAP')
IPython.display.Audio(audio,rate=fs)
```

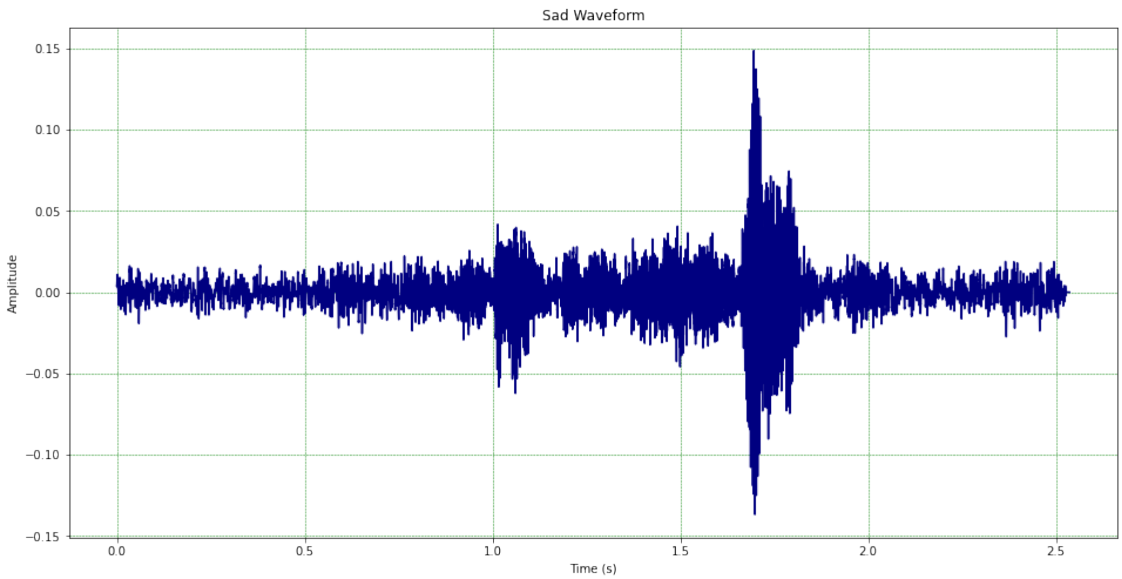


[906]: <IPython.lib.display.Audio object>

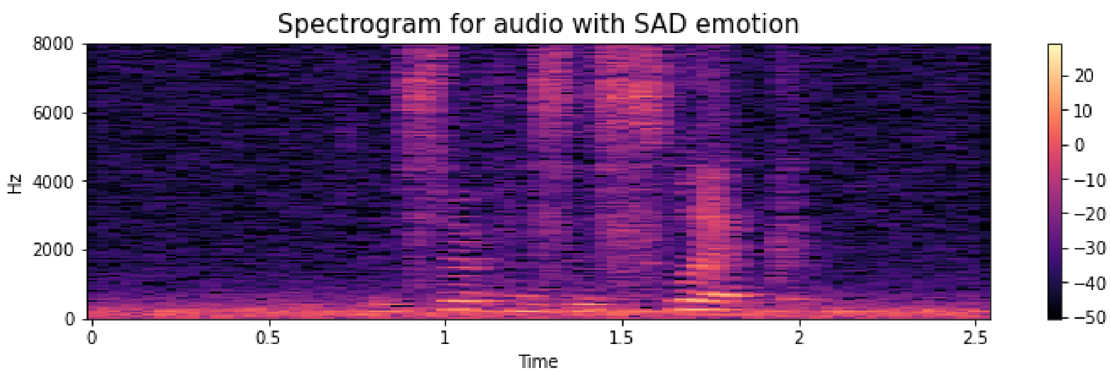


0.4.5 Sad Emotion

[907]:
```
# SAD
audio,fs = load_audio(audioss,'SAD')
create_spectrogram(audio,fs,'SAD')
IPython.display.Audio(audio,rate=fs)
```

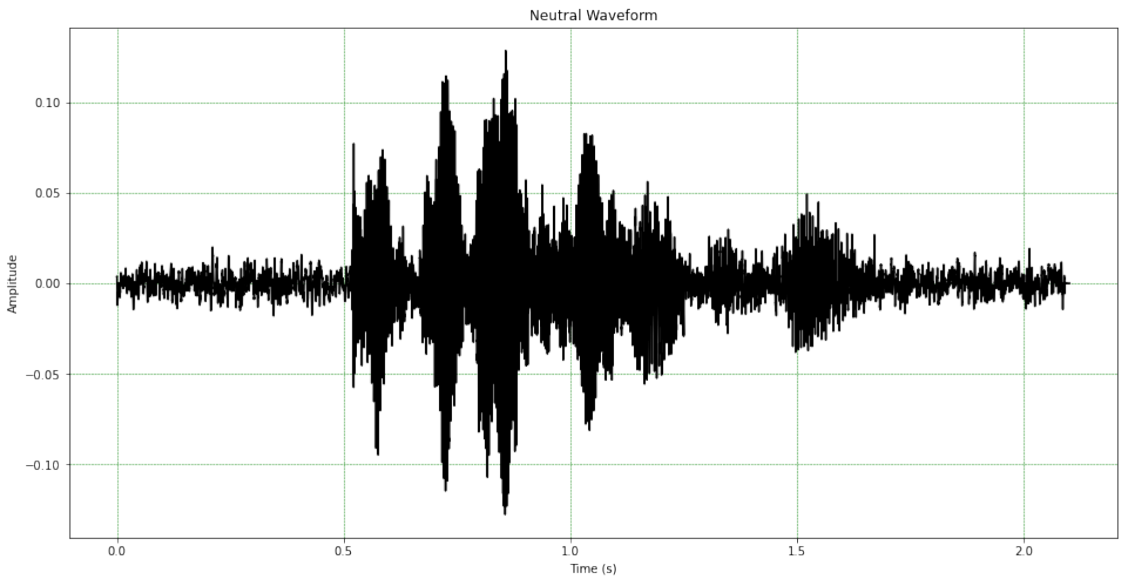


[907]: <IPython.lib.display.Audio object>

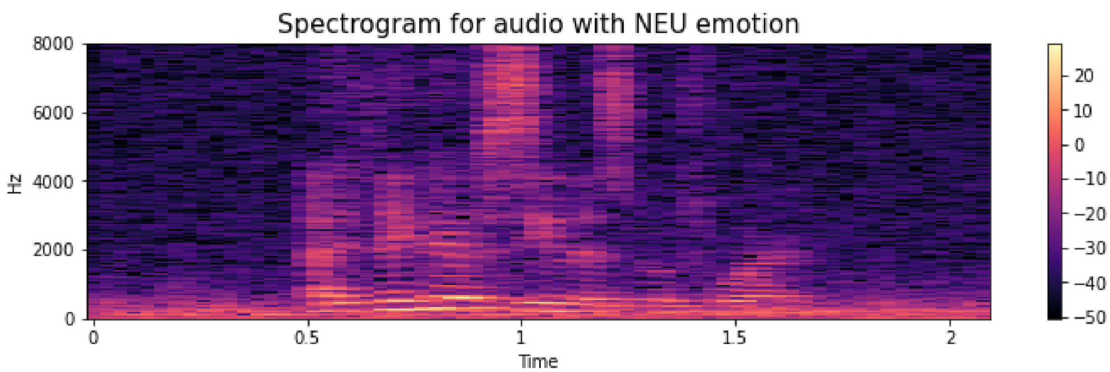


0.4.6 Neutral Emotion

[908]:
```
# NEUTRAL
audio,fs = load_audio(audioss,'NEU')
create_spectrogram(audio,fs,'NEU')
IPython.display.Audio(audio,rate=fs)
```



[908]: <IPython.lib.display.Audio object>



[ ]:
```
# THE FOLLOWING FUNCTION WAS USED TO UNIFY THE SIZE OF FEATURE VECTORS/MATRICES
↪BEFORE USING LIBROSA TO UNIFY THE LENGTH OF THE AUDIOS AT THE BEGINNING. WE
↪DON'T USE THAT NOW. LEFT FOR REFERENCE.
def pad(list,mel_spectrogram=False,max_len=None):
    padded_list = []
    if not mel_spectrogram:
        if max_len == None:
            max_len =  max([i.size for i in list])
        for l in list:
            if not l.size >= max_len:
                padded_list.append(np.pad(l, (0, max_len-l.size), 'constant'))
            else:
```

```
            padded_list.append(l[:max_len])
    else:
        if max_len == None:
            max_len =  max([i.shape[0] for i in list])
        for l in list:
            if not l.shape[0] >= max_len:
                padded_list.append(np.pad(l, [(0, max_len-l.shape[0]),(0,0)],␣
↪mode='constant'))
            else:
                padded_list.append(l[:max_len,:])
    return padded_list,max_len
```

0.5 Splitting the Dataset into Train, Test and Validation

We split the data into 70% train and validation, and 30% test. Then, we further split the train and validation set into 95% train and 5% vaidation. We proceed by encoding the labels using sklearn's `LabelEncoder()`.

[909]:
```
X_trainval, X_test, y_trainval, y_test = train_test_split(df['data'].
 ↪to_numpy(), np.array(labels), test_size=0.3, random_state=42,stratify=labels)
X_train, X_val, y_train, y_val = train_test_split(X_trainval, y_trainval,␣
 ↪test_size=0.05, random_state=42,stratify=y_trainval)
```

[910]:
```
le = LabelEncoder()
y_train = le.fit_transform(y_train)
y_test = le.transform(y_test)
y_val = le.transform(y_val)
```

0.6 Create the Feature Space (30 Points)

We experimented with many feature combinations and tried multiple dimensionality reduction techniques (i.e PCA, LDA) to try to capture trends and temporal patterns in speech emotion changes. We created two feature spaces, a 1-dimensional feature space, and a 2-dimensional one. Here are some features we tried:

1. Zero Crossing Rate : The rate of sign-changes of the signal during the duration of a particular frame.
2. Energy : The sum of squares of the signal values, normalized by the respective frame length.
3. Spectral Rolloff : The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
4. MFCCs: Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale.
5. Delta MFCC: In addition to the MFCC features, these features capture the changes in the MFCC coefficients over time, providing more information about the speech signal dynamics.

6. Chroma Features: Chroma features such as chroma energy and chroma deviation can provide useful information related to the pitch content or musicality of speech, which may be relevant for certain emotional states or variations in speech patterns.

7. Pitch mean, standard deviation, and range.

After many experimentations, we were able to yield the best results using ZCR, ENERGY, MFCCS, and SPECTRAL ROLLOFF with no dimensionality reduction. (1-dimensional feature space)

The 2-dimensional feature space consisted of the mel spectrogram of each audio with `n_mels=128`.

`Note: We tried using zcr, energy, and specral rolloff as arrays and as mean values. Mean values provide a global summary of the feature, while using an array of values captures the temporal dynamics. We settled on incorporating them as arrays.`

[ ]:
```
# OBSOLETE. LEFT FOR REFERENCE.

def create_feature_space(X,y=None,mel_spectrogram=False,training = False,
 ↪max_len_zcr=None,max_len_en=None,max_len_mfccs=None,max_len_mel=None,clf=None,mean=None,std
 ↪
    # labels = []
    if not mel_spectrogram: # time/freq domain feature space
        feature_space = []
        zcrs=[]
        energies=[]
        mfccs=[]
        rolloff=[]
        # lfccs = []
        # transform = transforms.LFCC(
        # speckwargs={"n_fft": 400, "hop_length": 512, "center": False})
        for i,data in enumerate(X):
            # audio = np.array(data[0][0],dtype='float')
            audio = data[0]
            fs = data[1]
            # zero_crosses = np.nonzero(np.diff(audio > 0))[0]
            # zcr = zero_crosses.size/len(audio) # zero crossing rate

            # zero crossing rate
            # zcr=np.squeeze(librosa.feature.
 ↪zero_crossing_rate(audio,frame_length=2048,hop_length=512))
            zcr=np.mean(librosa.feature.
 ↪zero_crossing_rate(audio,frame_length=2048,hop_length=512).T,axis=0)

            # print(zcr.shape)
            zcrs.append(zcr)

            # normalized energy
```

```
        # energy=np.squeeze(librosa.feature.
↪rms(y=audio,frame_length=2048,hop_length=512))
        # energy = np.array([sum(audio[j:j+2048]**2)/2048 for j in range(0,␣
↪len(audio), 512)])
        # print(energy.shape)
        energy=np.mean(librosa.feature.
↪rms(y=audio,frame_length=2048,hop_length=512).T,axis=0)
        energies.append(energy)

        # mel frequency cepstral coefficient (MFCC)
        mfcc=np.ravel(librosa.feature.mfcc(y=audio,sr=fs).T)
        # print(mfcc.shape)
        mfccs.append(mfcc)

        roll = np.squeeze(librosa.feature.spectral_rolloff(y=audio,␣
↪sr=fs,hop_length=512))
        rolloff.append(roll)

        # lfcc = transform(tf.convert_to_tensor(audio)[0])
        # lfccs.append(lfcc.numpy())

        # feature_space.append(np.concatenate((zcr,energy,mfcc), axis=None))
        # labels.append(data[1])

        if i%1000==0:
            print(f'audio #{i} checkpoint')

    # max_len = max([len(i) for i in zcrs]) # find the longest list of scr␣
↪to pad the others until they have equal length
    # if not training:
    #     zcrs,_ = pad(zcrs,max_len=max_len_zcr)
    #     energies,_ = pad(energies,max_len=max_len_en)
    #     mfccs,_ = pad(mfccs,max_len=max_len_mfccs)
    # else:
    #     zcrs,max_len_z = pad(zcrs)
    #     energies,max_len_e = pad(energies)
    #     mfccs,max_len_m = pad(mfccs)

    # lfccs = pad(lfccs)

    for i in range(X.shape[0]):
        # feature_space.append(np.
↪concatenate((zcrs[i],energies[i],mfccs[i],lfccs[i]), axis=None))
        feature_space.append(np.
↪concatenate((zcrs[i],energies[i],mfccs[i],rolloff[i]), axis=None))
```

```
        # labels = np.array(labels)
        feature_space = np.array(feature_space)
        # if training:
            # clf = LinearDiscriminantAnalysis(n_components=5).
↪fit(feature_space,y)
            # feature_space=clf.transform(feature_space)
            # return feature_space,max_len_z,max_len_e,max_len_m,clf
            # return feature_space,clf
        # else:
            # feature_space=clf.transform(feature_space)
        return feature_space
    else:
        mels=[]
        for i,data in enumerate(X):
            audio = data[0]
            fs = data[1]
            mel = librosa.feature.melspectrogram(y=audio, sr=fs,␣
↪n_fft=2048,n_mels=128).T
            # mel = librosa.feature.melspectrogram(y=audio, sr=fs,␣
↪n_fft=1024,n_mels=128)
            # mel_db = librosa.power_to_db(mel, ref=np.max)
            mels.append(mel)
            # labels.append(data[1])
            if i%1000==0:
                print(f'audio #{i} checkpoint')
        mels=np.array(mels)
        # if training:
            # mels,max_len_mels = pad(mels,mel_spectrogram=True)
            # mean_value = np.mean(mels,axis=0)
            # std_value = np.std(mels,axis=0)
            # norm_mels = (mels - mean_value) / std_value
            # return norm_mels,max_len_mels,mean_value,std_value
            # return norm_mels,mean_value,std_value

        # else:
            # mels,_ = pad(mels,mel_spectrogram=True,max_len=max_len_mel)
            # norm_mels = (mels - mean) / stddev
        return mels
```

[931]:
```
def extract_features(audio,mel_spectrogram=False):
    if not mel_spectrogram:
        result = np.array([])

        # zcr=np.mean(librosa.feature.
↪zero_crossing_rate(audio,frame_length=2048,hop_length=512).T,axis=0)
        zcr=np.squeeze(librosa.feature.
↪zero_crossing_rate(audio,frame_length=2048,hop_length=512))
```

```
        result = np.hstack((result,zcr))

        # rms = np.mean(librosa.feature.
↪rms(y=audio,frame_length=2048,hop_length=512).T ,axis=0)
        rms = np.squeeze(librosa.feature.
↪rms(y=audio,frame_length=2048,hop_length=512))

        result = np.hstack((result, rms))

        # rolloff = np.mean(librosa.feature.spectral_rolloff(y=audio,␣
↪sr=16000,n_fft=2048,hop_length=512).T,axis=0)
        rolloff = np.squeeze(librosa.feature.spectral_rolloff(y=audio,␣
↪sr=16000,n_fft=2048,hop_length=512))

        result = np.hstack((result, rolloff))

        # stft = np.abs(librosa.stft(audio))
        # chroma_stft = np.mean(librosa.feature.chroma_stft(S=stft, sr=16000).
↪T, axis=0)
        # result = np.hstack((result, chroma_stft))

        # mfcc = np.mean(librosa.feature.mfcc(y=audio, sr=16000).T, axis=0)
        mfcc = np.ravel(librosa.feature.mfcc(y=audio, sr=16000).T)
        result = np.hstack((result, mfcc))

        # delta_mfcc = np.mean(librosa.feature.delta(mfcc).T,axis=0)
        # result = np.hstack((result, delta_mfcc))

        # delta2_mfcc = np.mean(librosa.feature.delta(mfcc, order=2).T,axis=0)
        # result = np.hstack((result, delta2_mfcc))

        # Extract chroma features
        # chroma = librosa.feature.chroma_stft(y=audio, sr=16000)
        # chroma_energy = np.mean(chroma, axis=1)
        # # chroma_deviation = np.std(chroma, axis=1)
        # result = np.hstack((result, chroma_energy))
        # result = np.hstack((result, chroma_deviation))

        # pitch, magnitudes = librosa.core.piptrack(y=audio, sr=16000)
        # pitch_mean = np.mean(pitch, axis=0)
        # pitch_std = np.std(pitch, axis=0)
        # pitch_range = np.max(pitch, axis=0) - np.min(pitch, axis=0)
        # result = np.hstack((result, pitch_mean))
        # result = np.hstack((result, pitch_std))
        # result = np.hstack((result, pitch_range))
        return result
```

```
    else:
        mel = librosa.feature.melspectrogram(y=audio, sr=16000, n_fft=2048)
        return mel
```

[912]:
```
X_trainn = np.array(np.array([x for x,_ in X_train]))
X_testt = np.array(np.array([x for x,_ in X_test]))
X_vall = np.array(np.array([x for x,_ in X_val]))
```

[ ]:
```
sc=StandardScaler()
X_trainn=sc.fit_transform(X_trainn)
X_testt=sc.transform(X_testt)
X_vall=sc.transform(X_vall)
```

[ ]:
```
# Apply data augmentation techniques (example: random cropping)
def random_crop(mel_spectrogram, crop_size):
    h, w = mel_spectrogram.shape
    max_x = h - crop_size
    max_y = w - crop_size
    x = np.random.randint(0, max_x)
    y = np.random.randint(0, max_y)
    cropped = mel_spectrogram[x:x+crop_size, y:y+crop_size]
    return cropped
```

[ ]:
```
features_train = []
features_test = []
features_val = []

for i in range(X_trainn.shape[0]):
    features_train.append(extract_features(X_trainn[i]))
for i in range(X_testt.shape[0]):
    features_test.append(extract_features(X_testt[i]))
for i in range(X_vall.shape[0]):
    features_val.append(extract_features(X_vall[i]))
```

[ ]:
```
features_train = np.array(features_train)
features_test = np.array(features_test)
features_val = np.array(features_val)
```

[ ]:
```
X_trainn = np.array(np.array([x for x,_ in X_train]))
X_testt = np.array(np.array([x for x,_ in X_test]))
X_vall = np.array(np.array([x for x,_ in X_val]))
```

[932]:
```
mels_train = []
mels_test=[]
mels_val=[]
for i in range(X_trainn.shape[0]):
```

```
    mels_train.append(extract_features(X_trainn[i],mel_spectrogram=True))
for i in range(X_testt.shape[0]):
    mels_test.append(extract_features(X_testt[i],mel_spectrogram=True))
for i in range(X_vall.shape[0]):
    mels_val.append(extract_features(X_vall[i],mel_spectrogram=True))
```

[933]:
```
mels_train=np.array(mels_train)
mels_test = np.array(mels_test)
mels_val=np.array(mels_val)
```

[935]:
```
mels_train = np.expand_dims(mels_train,axis=3)
mels_test = np.expand_dims(mels_test,axis=3)
mels_val = np.expand_dims(mels_val,axis=3)
```

[936]:
```
mels_train.shape
```

[936]:
```
(4948, 128, 69, 1)
```

[ ]:
```
# lda = LinearDiscriminantAnalysis(n_components=5).fit(features_train,y_train)
# features_trainn = lda.transform(features_train)
# features_testt = lda.transform(features_test)
# features_vall = lda.transform(features_val)
from sklearn.decomposition import PCA
pca = PCA(n_components=50)
features_trainn = pca.fit_transform(X_trainn)
features_testt = pca.transform(X_testt)
features_vall = pca.transform(X_vall)
```

0.7 Building the Model (40 points)

We tried two different models for the 1-dimensional feature space. The major difference was that one of the models had LSTM layers while the other did not. Also, the model with no LSTM layers was much deeper, with 5,554,310 trainable parameters in contrast to just 189,190 trainable parameters. Obviously, The model with LSTM layers was much faster, and actually yielded better results. We were able to reach a test accuracy of 52.53% using that model. The other model yielded a test accuracy of 50.8%

[917]:
```
early_stop=EarlyStopping(monitor='val_acc',mode='auto',patience=20,restore_best_weights=True)
lr_reduction=ReduceLROnPlateau(monitor='val_acc',patience=5,verbose=1,factor=0.
↪75,min_lr=0.000001)
```

[ ]:
```
model=tf.keras.Sequential([
    L.Conv1D(512,kernel_size=5, strides=1,padding='same',␣
↪activation='relu',input_shape=(features_train.shape[1],1)),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    L.Conv1D(512,kernel_size=5,strides=1,padding='same',activation='relu'),
```

```
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    L.Conv1D(256,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    L.Conv1D(256,kernel_size=3,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    L.Dropout(0.3),
    L.Conv1D(128,kernel_size=3,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=3,strides=2,padding='same'),
    L.Flatten(),
    L.Dense(512,activation='relu'),
    L.BatchNormalization(),
    L.Dense(6,activation='softmax')
])
opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
model.compile(loss='sparse_categorical_crossentropy', optimizer=opt␣
 ↪,metrics=['acc'])
model.summary()
```

```
Model: "sequential_122"

_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 conv1d_386 (Conv1D)         (None, 1587, 512)         3072      

 batch_normalization_446 (Ba  (None, 1587, 512)        2048      
 tchNormalization)                                               

 max_pooling1d_377 (MaxPooli  (None, 794, 512)         0         
 ng1D)                                                           

 conv1d_387 (Conv1D)         (None, 794, 512)          1311232   

 batch_normalization_447 (Ba  (None, 794, 512)         2048      
 tchNormalization)                                               

 max_pooling1d_378 (MaxPooli  (None, 397, 512)         0         
 ng1D)                                                           

 conv1d_388 (Conv1D)         (None, 397, 256)          655616    

 batch_normalization_448 (Ba  (None, 397, 256)         1024      
 tchNormalization)                                               
```

```
 max_pooling1d_379 (MaxPooli  (None, 199, 256)         0
 ng1D)

 conv1d_389 (Conv1D)         (None, 199, 256)          196864

 batch_normalization_449 (Ba  (None, 199, 256)         1024
 tchNormalization)

 max_pooling1d_380 (MaxPooli  (None, 100, 256)         0
 ng1D)

 dropout_65 (Dropout)        (None, 100, 256)          0

 conv1d_390 (Conv1D)         (None, 100, 128)          98432

 batch_normalization_450 (Ba  (None, 100, 128)         512
 tchNormalization)

 max_pooling1d_381 (MaxPooli  (None, 50, 128)          0
 ng1D)

 flatten_52 (Flatten)        (None, 6400)              0

 dense_243 (Dense)           (None, 512)               3277312

 batch_normalization_451 (Ba  (None, 512)              2048
 tchNormalization)

 dense_244 (Dense)           (None, 6)                 3078

=================================================================
Total params: 5,554,310
Trainable params: 5,549,958
Non-trainable params: 4,352
_________________________________________________________________
```

[ ]:

```
# Define the model architecture
modelopt = Sequential()
modelopt.add(Conv1D(filters=32, kernel_size=5, activation='relu',␣
 ↪input_shape=(features_train.shape[1], 1)))
modelopt.add(MaxPooling1D(pool_size=2,padding='same'))
# modelopt.add(BatchNormalization())
modelopt.add(Conv1D(filters=64, kernel_size=3, activation='relu'))
modelopt.add(MaxPooling1D(pool_size=2,padding='same'))
# modelopt.add(BatchNormalization())
modelopt.add(Conv1D(filters=256, kernel_size=3, activation='relu'))
modelopt.add(MaxPooling1D(pool_size=2,padding='same'))
```

```python
# modelopt.add(Conv1D(filters=256, kernel_size=3, activation='relu'))
# modelopt.add(MaxPooling1D(pool_size=2,padding='same'))
modelopt.add(LSTM(64, return_sequences=True))
modelopt.add(LSTM(64))
# modelopt.add(Dropout(0.5))
modelopt.add(Dense(256, activation='relu'))#, kernel_regularizer=tf.keras.
 ↪regularizers.l1_l2(l1=0.01, l2=0.01)))
# modelopt.add(Dropout(0.5))
modelopt.add(Dense(6, activation='softmax'))

opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
modelopt.compile(loss='sparse_categorical_crossentropy', optimizer=opt␣
 ↪,metrics=['acc'])
modelopt.summary()
```

```
Model: "sequential_123"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_391 (Conv1D)         (None, 1583, 32)          192

 max_pooling1d_382 (MaxPooli  (None, 792, 32)          0
 ng1D)

 conv1d_392 (Conv1D)         (None, 790, 64)           6208

 max_pooling1d_383 (MaxPooli  (None, 395, 64)          0
 ng1D)

 conv1d_393 (Conv1D)         (None, 393, 256)          49408

 max_pooling1d_384 (MaxPooli  (None, 197, 256)         0
 ng1D)

 lstm_88 (LSTM)              (None, 197, 64)           82176

 lstm_89 (LSTM)              (None, 64)                33024

 dense_245 (Dense)           (None, 256)               16640

 dense_246 (Dense)           (None, 6)                 1542

=================================================================
Total params: 189,190
Trainable params: 189,190
Non-trainable params: 0
_________________________________________________________________
```

[ ]:
```
histopt = modelopt.fit(features_train, y_train, batch_size=128, epochs=100,
    verbose=1, validation_data=(features_val, y_val),
    callbacks=[early_stop,lr_reduction])
```

```
Epoch 1/100
39/39 [==============================] - 79s 2s/step - loss: 1.7608 - acc:
0.2439 - val_loss: 1.7191 - val_acc: 0.2759 - lr: 1.0000e-04
Epoch 2/100
39/39 [==============================] - 79s 2s/step - loss: 1.6673 - acc:
0.3193 - val_loss: 1.5996 - val_acc: 0.3257 - lr: 1.0000e-04
Epoch 3/100
39/39 [==============================] - 74s 2s/step - loss: 1.5714 - acc:
0.3660 - val_loss: 1.5106 - val_acc: 0.4023 - lr: 1.0000e-04
Epoch 4/100
39/39 [==============================] - 74s 2s/step - loss: 1.5014 - acc:
0.4072 - val_loss: 1.4872 - val_acc: 0.3831 - lr: 1.0000e-04
Epoch 5/100
39/39 [==============================] - 74s 2s/step - loss: 1.4742 - acc:
0.4119 - val_loss: 1.4562 - val_acc: 0.3985 - lr: 1.0000e-04
Epoch 6/100
39/39 [==============================] - 74s 2s/step - loss: 1.4535 - acc:
0.4214 - val_loss: 1.4534 - val_acc: 0.4061 - lr: 1.0000e-04
Epoch 7/100
39/39 [==============================] - 74s 2s/step - loss: 1.4447 - acc:
0.4240 - val_loss: 1.4236 - val_acc: 0.4215 - lr: 1.0000e-04
Epoch 8/100
39/39 [==============================] - 73s 2s/step - loss: 1.4265 - acc:
0.4325 - val_loss: 1.4411 - val_acc: 0.4176 - lr: 1.0000e-04
Epoch 9/100
39/39 [==============================] - 73s 2s/step - loss: 1.4199 - acc:
0.4382 - val_loss: 1.4458 - val_acc: 0.4368 - lr: 1.0000e-04
Epoch 10/100
39/39 [==============================] - 74s 2s/step - loss: 1.4281 - acc:
0.4250 - val_loss: 1.4170 - val_acc: 0.4406 - lr: 1.0000e-04
Epoch 11/100
39/39 [==============================] - 73s 2s/step - loss: 1.4047 - acc:
0.4422 - val_loss: 1.3979 - val_acc: 0.4444 - lr: 1.0000e-04
Epoch 12/100
39/39 [==============================] - 74s 2s/step - loss: 1.3963 - acc:
0.4523 - val_loss: 1.4430 - val_acc: 0.3985 - lr: 1.0000e-04
Epoch 13/100
39/39 [==============================] - 73s 2s/step - loss: 1.3851 - acc:
0.4475 - val_loss: 1.4317 - val_acc: 0.3985 - lr: 1.0000e-04
Epoch 14/100
39/39 [==============================] - 73s 2s/step - loss: 1.3887 - acc:
0.4458 - val_loss: 1.4323 - val_acc: 0.4406 - lr: 1.0000e-04
Epoch 15/100
```

```
39/39 [==============================] - 72s 2s/step - loss: 1.3864 - acc:
0.4493 - val_loss: 1.4016 - val_acc: 0.4253 - lr: 1.0000e-04
Epoch 16/100
39/39 [==============================] - 75s 2s/step - loss: 1.3778 - acc:
0.4501 - val_loss: 1.3992 - val_acc: 0.4368 - lr: 1.0000e-04
Epoch 17/100
39/39 [==============================] - 77s 2s/step - loss: 1.3748 - acc:
0.4491 - val_loss: 1.4169 - val_acc: 0.3985 - lr: 1.0000e-04
Epoch 18/100
39/39 [==============================] - 75s 2s/step - loss: 1.3721 - acc:
0.4584 - val_loss: 1.4021 - val_acc: 0.4368 - lr: 1.0000e-04
Epoch 19/100
39/39 [==============================] - ETA: 0s - loss: 1.3557 - acc: 0.4689
Epoch 00019: ReduceLROnPlateau reducing learning rate to 7.499999810534064e-05.
39/39 [==============================] - 71s 2s/step - loss: 1.3557 - acc:
0.4689 - val_loss: 1.4187 - val_acc: 0.4138 - lr: 1.0000e-04
Epoch 20/100
39/39 [==============================] - 76s 2s/step - loss: 1.3503 - acc:
0.4614 - val_loss: 1.3958 - val_acc: 0.4330 - lr: 7.5000e-05
Epoch 21/100
39/39 [==============================] - 70s 2s/step - loss: 1.3608 - acc:
0.4557 - val_loss: 1.3781 - val_acc: 0.4444 - lr: 7.5000e-05
Epoch 22/100
39/39 [==============================] - 72s 2s/step - loss: 1.3443 - acc:
0.4713 - val_loss: 1.3920 - val_acc: 0.4444 - lr: 7.5000e-05
Epoch 23/100
39/39 [==============================] - 71s 2s/step - loss: 1.3564 - acc:
0.4604 - val_loss: 1.3880 - val_acc: 0.4521 - lr: 7.5000e-05
Epoch 24/100
39/39 [==============================] - 71s 2s/step - loss: 1.3353 - acc:
0.4679 - val_loss: 1.3746 - val_acc: 0.4215 - lr: 7.5000e-05
Epoch 25/100
39/39 [==============================] - 71s 2s/step - loss: 1.3265 - acc:
0.4729 - val_loss: 1.3750 - val_acc: 0.4559 - lr: 7.5000e-05
Epoch 26/100
39/39 [==============================] - 70s 2s/step - loss: 1.3233 - acc:
0.4780 - val_loss: 1.3621 - val_acc: 0.4483 - lr: 7.5000e-05
Epoch 27/100
39/39 [==============================] - 71s 2s/step - loss: 1.3349 - acc:
0.4634 - val_loss: 1.3901 - val_acc: 0.4368 - lr: 7.5000e-05
Epoch 28/100
39/39 [==============================] - 71s 2s/step - loss: 1.3293 - acc:
0.4671 - val_loss: 1.3899 - val_acc: 0.4598 - lr: 7.5000e-05
Epoch 29/100
39/39 [==============================] - 70s 2s/step - loss: 1.3296 - acc:
0.4695 - val_loss: 1.4047 - val_acc: 0.4330 - lr: 7.5000e-05
Epoch 30/100
39/39 [==============================] - 71s 2s/step - loss: 1.3379 - acc:
```

```
0.4612 - val_loss: 1.3621 - val_acc: 0.4713 - lr: 7.5000e-05
Epoch 31/100
39/39 [==============================] - 71s 2s/step - loss: 1.3180 - acc:
0.4818 - val_loss: 1.4135 - val_acc: 0.4061 - lr: 7.5000e-05
Epoch 32/100
39/39 [==============================] - 16514s 435s/step - loss: 1.3021 - acc:
0.4879 - val_loss: 1.3768 - val_acc: 0.4751 - lr: 7.5000e-05
Epoch 33/100
39/39 [==============================] - 74s 2s/step - loss: 1.3016 - acc:
0.4834 - val_loss: 1.3687 - val_acc: 0.4751 - lr: 7.5000e-05
Epoch 34/100
39/39 [==============================] - 71s 2s/step - loss: 1.2986 - acc:
0.4812 - val_loss: 1.3799 - val_acc: 0.4828 - lr: 7.5000e-05
Epoch 35/100
39/39 [==============================] - 71s 2s/step - loss: 1.3032 - acc:
0.4885 - val_loss: 1.3757 - val_acc: 0.4674 - lr: 7.5000e-05
Epoch 36/100
39/39 [==============================] - 72s 2s/step - loss: 1.3142 - acc:
0.4749 - val_loss: 1.3644 - val_acc: 0.4674 - lr: 7.5000e-05
Epoch 37/100
39/39 [==============================] - 76s 2s/step - loss: 1.2982 - acc:
0.4838 - val_loss: 1.4007 - val_acc: 0.4330 - lr: 7.5000e-05
Epoch 38/100
39/39 [==============================] - 72s 2s/step - loss: 1.2957 - acc:
0.4840 - val_loss: 1.3676 - val_acc: 0.4406 - lr: 7.5000e-05
Epoch 39/100
39/39 [==============================] - 72s 2s/step - loss: 1.3100 - acc:
0.4776 - val_loss: 1.3757 - val_acc: 0.4330 - lr: 7.5000e-05
Epoch 40/100
39/39 [==============================] - 72s 2s/step - loss: 1.2836 - acc:
0.4923 - val_loss: 1.3728 - val_acc: 0.4636 - lr: 7.5000e-05
Epoch 41/100
39/39 [==============================] - 72s 2s/step - loss: 1.2749 - acc:
0.4976 - val_loss: 1.3532 - val_acc: 0.4751 - lr: 7.5000e-05
Epoch 42/100
39/39 [==============================] - ETA: 0s - loss: 1.2689 - acc: 0.5044
Epoch 00042: ReduceLROnPlateau reducing learning rate to 5.6249997214763425e-05.
39/39 [==============================] - 72s 2s/step - loss: 1.2689 - acc:
0.5044 - val_loss: 1.4133 - val_acc: 0.4368 - lr: 7.5000e-05
Epoch 43/100
39/39 [==============================] - 73s 2s/step - loss: 1.2676 - acc:
0.4964 - val_loss: 1.3666 - val_acc: 0.4330 - lr: 5.6250e-05
Epoch 44/100
39/39 [==============================] - 72s 2s/step - loss: 1.2527 - acc:
0.5063 - val_loss: 1.3693 - val_acc: 0.4636 - lr: 5.6250e-05
Epoch 45/100
39/39 [==============================] - 72s 2s/step - loss: 1.2673 - acc:
0.4998 - val_loss: 1.3426 - val_acc: 0.4674 - lr: 5.6250e-05
```

```
Epoch 46/100
39/39 [==============================] - 71s 2s/step - loss: 1.2475 - acc:
0.5075 - val_loss: 1.3607 - val_acc: 0.4789 - lr: 5.6250e-05
Epoch 47/100
39/39 [==============================] - 71s 2s/step - loss: 1.2543 - acc:
0.5073 - val_loss: 1.3575 - val_acc: 0.4521 - lr: 5.6250e-05
Epoch 48/100
39/39 [==============================] - 70s 2s/step - loss: 1.2410 - acc:
0.5141 - val_loss: 1.3375 - val_acc: 0.5057 - lr: 5.6250e-05
Epoch 49/100
39/39 [==============================] - 69s 2s/step - loss: 1.2341 - acc:
0.5202 - val_loss: 1.3472 - val_acc: 0.4828 - lr: 5.6250e-05
Epoch 50/100
39/39 [==============================] - 69s 2s/step - loss: 1.2397 - acc:
0.5162 - val_loss: 1.3541 - val_acc: 0.4521 - lr: 5.6250e-05
Epoch 51/100
39/39 [==============================] - 69s 2s/step - loss: 1.2345 - acc:
0.5172 - val_loss: 1.3473 - val_acc: 0.4828 - lr: 5.6250e-05
Epoch 52/100
39/39 [==============================] - 74s 2s/step - loss: 1.2367 - acc:
0.5109 - val_loss: 1.3486 - val_acc: 0.4674 - lr: 5.6250e-05
Epoch 53/100
39/39 [==============================] - 69s 2s/step - loss: 1.2397 - acc:
0.5188 - val_loss: 1.3298 - val_acc: 0.4828 - lr: 5.6250e-05
Epoch 54/100
39/39 [==============================] - 69s 2s/step - loss: 1.2158 - acc:
0.5301 - val_loss: 1.3329 - val_acc: 0.4828 - lr: 5.6250e-05
Epoch 55/100
39/39 [==============================] - 73s 2s/step - loss: 1.2189 - acc:
0.5249 - val_loss: 1.3547 - val_acc: 0.4943 - lr: 5.6250e-05
Epoch 56/100
39/39 [==============================] - ETA: 0s - loss: 1.2259 - acc: 0.5257
Epoch 00056: ReduceLROnPlateau reducing learning rate to 4.218749927531462e-05.
39/39 [==============================] - 81s 2s/step - loss: 1.2259 - acc:
0.5257 - val_loss: 1.3703 - val_acc: 0.4866 - lr: 5.6250e-05
Epoch 57/100
39/39 [==============================] - 70s 2s/step - loss: 1.2160 - acc:
0.5329 - val_loss: 1.3372 - val_acc: 0.4789 - lr: 4.2188e-05
Epoch 58/100
39/39 [==============================] - 69s 2s/step - loss: 1.2112 - acc:
0.5279 - val_loss: 1.3572 - val_acc: 0.4713 - lr: 4.2188e-05
Epoch 59/100
39/39 [==============================] - 69s 2s/step - loss: 1.2145 - acc:
0.5249 - val_loss: 1.3572 - val_acc: 0.4636 - lr: 4.2188e-05
Epoch 60/100
39/39 [==============================] - 70s 2s/step - loss: 1.1977 - acc:
0.5378 - val_loss: 1.3386 - val_acc: 0.4598 - lr: 4.2188e-05
Epoch 61/100
```

```
39/39 [==============================] - 74s 2s/step - loss: 1.2118 - acc:
0.5263 - val_loss: 1.3393 - val_acc: 0.4789 - lr: 4.2188e-05
Epoch 62/100
39/39 [==============================] - 76s 2s/step - loss: 1.1874 - acc:
0.5382 - val_loss: 1.3154 - val_acc: 0.4789 - lr: 4.2188e-05
Epoch 63/100
39/39 [==============================] - 75s 2s/step - loss: 1.1819 - acc:
0.5473 - val_loss: 1.3151 - val_acc: 0.5057 - lr: 4.2188e-05
Epoch 64/100
39/39 [==============================] - ETA: 0s - loss: 1.1825 - acc: 0.5471
Epoch 00064: ReduceLROnPlateau reducing learning rate to 3.164062582072802e-05.
39/39 [==============================] - 74s 2s/step - loss: 1.1825 - acc:
0.5471 - val_loss: 1.3297 - val_acc: 0.4828 - lr: 4.2188e-05
Epoch 65/100
39/39 [==============================] - 78s 2s/step - loss: 1.1725 - acc:
0.5493 - val_loss: 1.3112 - val_acc: 0.5172 - lr: 3.1641e-05
Epoch 66/100
39/39 [==============================] - 73s 2s/step - loss: 1.1690 - acc:
0.5477 - val_loss: 1.2970 - val_acc: 0.5172 - lr: 3.1641e-05
Epoch 67/100
39/39 [==============================] - 71s 2s/step - loss: 1.1552 - acc:
0.5604 - val_loss: 1.3335 - val_acc: 0.4866 - lr: 3.1641e-05
Epoch 68/100
39/39 [==============================] - 70s 2s/step - loss: 1.1634 - acc:
0.5491 - val_loss: 1.3186 - val_acc: 0.5134 - lr: 3.1641e-05
Epoch 69/100
39/39 [==============================] - 70s 2s/step - loss: 1.1543 - acc:
0.5568 - val_loss: 1.3337 - val_acc: 0.4904 - lr: 3.1641e-05
Epoch 70/100
39/39 [==============================] - 72s 2s/step - loss: 1.1555 - acc:
0.5572 - val_loss: 1.3495 - val_acc: 0.4866 - lr: 3.1641e-05
Epoch 71/100
39/39 [==============================] - 70s 2s/step - loss: 1.1506 - acc:
0.5523 - val_loss: 1.3895 - val_acc: 0.4598 - lr: 3.1641e-05
Epoch 72/100
39/39 [==============================] - 71s 2s/step - loss: 1.1671 - acc:
0.5509 - val_loss: 1.3216 - val_acc: 0.4981 - lr: 3.1641e-05
Epoch 73/100
39/39 [==============================] - ETA: 0s - loss: 1.1488 - acc: 0.5588
Epoch 00073: ReduceLROnPlateau reducing learning rate to 2.3730469365546014e-05.
39/39 [==============================] - 72s 2s/step - loss: 1.1488 - acc:
0.5588 - val_loss: 1.3497 - val_acc: 0.4674 - lr: 3.1641e-05
Epoch 74/100
39/39 [==============================] - 72s 2s/step - loss: 1.1378 - acc:
0.5641 - val_loss: 1.3296 - val_acc: 0.4866 - lr: 2.3730e-05
Epoch 75/100
39/39 [==============================] - 72s 2s/step - loss: 1.1328 - acc:
0.5679 - val_loss: 1.3408 - val_acc: 0.4674 - lr: 2.3730e-05
```

```
Epoch 76/100
39/39 [==============================] - 71s 2s/step - loss: 1.1539 - acc:
0.5592 - val_loss: 1.3126 - val_acc: 0.5019 - lr: 2.3730e-05
Epoch 77/100
39/39 [==============================] - 70s 2s/step - loss: 1.1407 - acc:
0.5622 - val_loss: 1.3163 - val_acc: 0.4943 - lr: 2.3730e-05
Epoch 78/100
39/39 [==============================] - 71s 2s/step - loss: 1.1272 - acc:
0.5645 - val_loss: 1.3257 - val_acc: 0.4904 - lr: 2.3730e-05
Epoch 79/100
39/39 [==============================] - 72s 2s/step - loss: 1.1239 - acc:
0.5663 - val_loss: 1.3001 - val_acc: 0.5211 - lr: 2.3730e-05
Epoch 80/100
39/39 [==============================] - 77s 2s/step - loss: 1.1184 - acc:
0.5713 - val_loss: 1.3142 - val_acc: 0.4943 - lr: 2.3730e-05
Epoch 81/100
39/39 [==============================] - 72s 2s/step - loss: 1.1129 - acc:
0.5707 - val_loss: 1.3346 - val_acc: 0.4866 - lr: 2.3730e-05
Epoch 82/100
39/39 [==============================] - 72s 2s/step - loss: 1.1189 - acc:
0.5695 - val_loss: 1.3052 - val_acc: 0.5172 - lr: 2.3730e-05
Epoch 83/100
39/39 [==============================] - 72s 2s/step - loss: 1.1215 - acc:
0.5669 - val_loss: 1.2936 - val_acc: 0.5249 - lr: 2.3730e-05
Epoch 84/100
39/39 [==============================] - 73s 2s/step - loss: 1.1091 - acc:
0.5780 - val_loss: 1.3268 - val_acc: 0.5172 - lr: 2.3730e-05
Epoch 85/100
39/39 [==============================] - 72s 2s/step - loss: 1.1031 - acc:
0.5740 - val_loss: 1.3105 - val_acc: 0.5057 - lr: 2.3730e-05
Epoch 86/100
39/39 [==============================] - 73s 2s/step - loss: 1.1124 - acc:
0.5756 - val_loss: 1.3395 - val_acc: 0.4751 - lr: 2.3730e-05
Epoch 87/100
39/39 [==============================] - 73s 2s/step - loss: 1.1198 - acc:
0.5699 - val_loss: 1.3355 - val_acc: 0.4866 - lr: 2.3730e-05
Epoch 88/100
39/39 [==============================] - 73s 2s/step - loss: 1.1083 - acc:
0.5770 - val_loss: 1.3077 - val_acc: 0.5211 - lr: 2.3730e-05
Epoch 89/100
39/39 [==============================] - 71s 2s/step - loss: 1.1043 - acc:
0.5762 - val_loss: 1.3249 - val_acc: 0.5057 - lr: 2.3730e-05
Epoch 90/100
39/39 [==============================] - 70s 2s/step - loss: 1.0990 - acc:
0.5798 - val_loss: 1.3275 - val_acc: 0.4981 - lr: 2.3730e-05
Epoch 91/100
39/39 [==============================] - ETA: 0s - loss: 1.1012 - acc: 0.5770
Epoch 00091: ReduceLROnPlateau reducing learning rate to 1.7797852706280537e-05.
```

```
39/39 [==============================] - 70s 2s/step - loss: 1.1012 - acc:
0.5770 - val_loss: 1.3347 - val_acc: 0.4904 - lr: 2.3730e-05
Epoch 92/100
39/39 [==============================] - 70s 2s/step - loss: 1.0884 - acc:
0.5792 - val_loss: 1.3217 - val_acc: 0.4866 - lr: 1.7798e-05
Epoch 93/100
39/39 [==============================] - 70s 2s/step - loss: 1.0901 - acc:
0.5839 - val_loss: 1.3240 - val_acc: 0.4904 - lr: 1.7798e-05
Epoch 94/100
39/39 [==============================] - 70s 2s/step - loss: 1.0927 - acc:
0.5837 - val_loss: 1.3017 - val_acc: 0.5172 - lr: 1.7798e-05
Epoch 95/100
39/39 [==============================] - 72s 2s/step - loss: 1.0813 - acc:
0.5887 - val_loss: 1.2979 - val_acc: 0.5096 - lr: 1.7798e-05
Epoch 96/100
39/39 [==============================] - 72s 2s/step - loss: 1.0755 - acc:
0.5897 - val_loss: 1.3096 - val_acc: 0.4943 - lr: 1.7798e-05
Epoch 97/100
39/39 [==============================] - 73s 2s/step - loss: 1.0863 - acc:
0.5869 - val_loss: 1.3011 - val_acc: 0.5172 - lr: 1.7798e-05
Epoch 98/100
39/39 [==============================] - 73s 2s/step - loss: 1.0799 - acc:
0.5859 - val_loss: 1.3044 - val_acc: 0.5211 - lr: 1.7798e-05
Epoch 99/100
39/39 [==============================] - ETA: 0s - loss: 1.0733 - acc: 0.5907
Epoch 00099: ReduceLROnPlateau reducing learning rate to 1.3348389529710403e-05.
39/39 [==============================] - 75s 2s/step - loss: 1.0733 - acc:
0.5907 - val_loss: 1.3056 - val_acc: 0.5057 - lr: 1.7798e-05
Epoch 100/100
39/39 [==============================] - 72s 2s/step - loss: 1.0719 - acc:
0.5901 - val_loss: 1.3062 - val_acc: 0.5134 - lr: 1.3348e-05
```

[ ]:
```
histopt = modelopt.fit(features_train, y_train, batch_size=128, epochs=50,␣
↪verbose=1, validation_data=(features_val, y_val),␣
↪callbacks=[early_stop,lr_reduction])
```

```
Epoch 1/50
39/39 [==============================] - 71s 2s/step - loss: 1.6248 - acc:
0.3207 - val_loss: 1.4396 - val_acc: 0.4253 - lr: 1.0000e-04
Epoch 2/50
39/39 [==============================] - 71s 2s/step - loss: 1.4346 - acc:
0.4236 - val_loss: 1.3934 - val_acc: 0.4713 - lr: 1.0000e-04
Epoch 3/50
39/39 [==============================] - 73s 2s/step - loss: 1.3835 - acc:
0.4491 - val_loss: 1.3718 - val_acc: 0.4598 - lr: 1.0000e-04
Epoch 4/50
39/39 [==============================] - 70s 2s/step - loss: 1.3553 - acc:
0.4721 - val_loss: 1.3697 - val_acc: 0.4751 - lr: 1.0000e-04
```

```
Epoch 5/50
39/39 [==============================] - 70s 2s/step - loss: 1.3730 - acc:
0.4612 - val_loss: 1.3567 - val_acc: 0.4483 - lr: 1.0000e-04
Epoch 6/50
39/39 [==============================] - 72s 2s/step - loss: 1.3154 - acc:
0.4800 - val_loss: 1.3859 - val_acc: 0.4406 - lr: 1.0000e-04
Epoch 7/50
39/39 [==============================] - 72s 2s/step - loss: 1.3030 - acc:
0.4861 - val_loss: 1.3745 - val_acc: 0.4598 - lr: 1.0000e-04
Epoch 8/50
39/39 [==============================] - 73s 2s/step - loss: 1.3520 - acc:
0.4642 - val_loss: 1.4411 - val_acc: 0.3870 - lr: 1.0000e-04
Epoch 9/50
39/39 [==============================] - 71s 2s/step - loss: 1.3400 - acc:
0.4685 - val_loss: 1.3733 - val_acc: 0.4291 - lr: 1.0000e-04
Epoch 10/50
39/39 [==============================] - 70s 2s/step - loss: 1.3025 - acc:
0.4875 - val_loss: 1.3715 - val_acc: 0.4406 - lr: 1.0000e-04
Epoch 11/50
39/39 [==============================] - 71s 2s/step - loss: 1.2826 - acc:
0.5030 - val_loss: 1.3559 - val_acc: 0.4636 - lr: 1.0000e-04
Epoch 12/50
39/39 [==============================] - ETA: 0s - loss: 1.2862 - acc: 0.4986
Epoch 00012: ReduceLROnPlateau reducing learning rate to 7.499999810534064e-05.
39/39 [==============================] - 70s 2s/step - loss: 1.2862 - acc:
0.4986 - val_loss: 1.3488 - val_acc: 0.4751 - lr: 1.0000e-04
Epoch 13/50
39/39 [==============================] - 73s 2s/step - loss: 1.2600 - acc:
0.5101 - val_loss: 1.3433 - val_acc: 0.4636 - lr: 7.5000e-05
Epoch 14/50
39/39 [==============================] - 73s 2s/step - loss: 1.2497 - acc:
0.5143 - val_loss: 1.3913 - val_acc: 0.4406 - lr: 7.5000e-05
Epoch 15/50
39/39 [==============================] - 75s 2s/step - loss: 1.2668 - acc:
0.5093 - val_loss: 1.3841 - val_acc: 0.4483 - lr: 7.5000e-05
Epoch 16/50
39/39 [==============================] - 72s 2s/step - loss: 1.2420 - acc:
0.5150 - val_loss: 1.3157 - val_acc: 0.4789 - lr: 7.5000e-05
Epoch 17/50
39/39 [==============================] - 73s 2s/step - loss: 1.2300 - acc:
0.5271 - val_loss: 1.3416 - val_acc: 0.4713 - lr: 7.5000e-05
Epoch 18/50
39/39 [==============================] - 74s 2s/step - loss: 1.2199 - acc:
0.5327 - val_loss: 1.3366 - val_acc: 0.4828 - lr: 7.5000e-05
Epoch 19/50
39/39 [==============================] - 72s 2s/step - loss: 1.2276 - acc:
0.5190 - val_loss: 1.3272 - val_acc: 0.4789 - lr: 7.5000e-05
Epoch 20/50
```

```
39/39 [==============================] - 73s 2s/step - loss: 1.2037 - acc:
0.5342 - val_loss: 1.3081 - val_acc: 0.4789 - lr: 7.5000e-05
Epoch 21/50
39/39 [==============================] - 72s 2s/step - loss: 1.1936 - acc:
0.5392 - val_loss: 1.3111 - val_acc: 0.4713 - lr: 7.5000e-05
Epoch 22/50
39/39 [==============================] - 73s 2s/step - loss: 1.2014 - acc:
0.5366 - val_loss: 1.3371 - val_acc: 0.4713 - lr: 7.5000e-05
Epoch 23/50
39/39 [==============================] - 72s 2s/step - loss: 1.1941 - acc:
0.5348 - val_loss: 1.3229 - val_acc: 0.4406 - lr: 7.5000e-05
Epoch 24/50
39/39 [==============================] - 75s 2s/step - loss: 1.1841 - acc:
0.5432 - val_loss: 1.2731 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 25/50
39/39 [==============================] - 72s 2s/step - loss: 1.1647 - acc:
0.5511 - val_loss: 1.3193 - val_acc: 0.4828 - lr: 7.5000e-05
Epoch 26/50
39/39 [==============================] - 72s 2s/step - loss: 1.1589 - acc:
0.5558 - val_loss: 1.2705 - val_acc: 0.4789 - lr: 7.5000e-05
Epoch 27/50
39/39 [==============================] - 72s 2s/step - loss: 1.1623 - acc:
0.5501 - val_loss: 1.2958 - val_acc: 0.4981 - lr: 7.5000e-05
Epoch 28/50
39/39 [==============================] - 72s 2s/step - loss: 1.1451 - acc:
0.5594 - val_loss: 1.3664 - val_acc: 0.4330 - lr: 7.5000e-05
Epoch 29/50
39/39 [==============================] - 72s 2s/step - loss: 1.1337 - acc:
0.5635 - val_loss: 1.3036 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 30/50
39/39 [==============================] - 72s 2s/step - loss: 1.1676 - acc:
0.5475 - val_loss: 1.3894 - val_acc: 0.4751 - lr: 7.5000e-05
Epoch 31/50
39/39 [==============================] - 72s 2s/step - loss: 1.1703 - acc:
0.5580 - val_loss: 1.2555 - val_acc: 0.5096 - lr: 7.5000e-05
Epoch 32/50
39/39 [==============================] - 73s 2s/step - loss: 1.1269 - acc:
0.5675 - val_loss: 1.3003 - val_acc: 0.4904 - lr: 7.5000e-05
Epoch 33/50
39/39 [==============================] - 72s 2s/step - loss: 1.1819 - acc:
0.5453 - val_loss: 1.3105 - val_acc: 0.4751 - lr: 7.5000e-05
Epoch 34/50
39/39 [==============================] - 72s 2s/step - loss: 1.1323 - acc:
0.5594 - val_loss: 1.2994 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 35/50
39/39 [==============================] - 72s 2s/step - loss: 1.1229 - acc:
0.5683 - val_loss: 1.3316 - val_acc: 0.4828 - lr: 7.5000e-05
Epoch 36/50
```

```
39/39 [==============================] - 72s 2s/step - loss: 1.1113 - acc:
0.5728 - val_loss: 1.2785 - val_acc: 0.5096 - lr: 7.5000e-05
Epoch 37/50
39/39 [==============================] - 72s 2s/step - loss: 1.1232 - acc:
0.5661 - val_loss: 1.2963 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 38/50
39/39 [==============================] - 72s 2s/step - loss: 1.1154 - acc:
0.5616 - val_loss: 1.2602 - val_acc: 0.5211 - lr: 7.5000e-05
Epoch 39/50
39/39 [==============================] - 72s 2s/step - loss: 1.1221 - acc:
0.5699 - val_loss: 1.2877 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 40/50
39/39 [==============================] - 72s 2s/step - loss: 1.0818 - acc:
0.5901 - val_loss: 1.2908 - val_acc: 0.4866 - lr: 7.5000e-05
Epoch 41/50
39/39 [==============================] - 127s 3s/step - loss: 1.1038 - acc:
0.5754 - val_loss: 1.2206 - val_acc: 0.5326 - lr: 7.5000e-05
Epoch 42/50
39/39 [==============================] - 90s 2s/step - loss: 1.0764 - acc:
0.5930 - val_loss: 1.2551 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 43/50
39/39 [==============================] - 90s 2s/step - loss: 1.0636 - acc:
0.5938 - val_loss: 1.2587 - val_acc: 0.5287 - lr: 7.5000e-05
Epoch 44/50
39/39 [==============================] - 91s 2s/step - loss: 1.0599 - acc:
0.5930 - val_loss: 1.3312 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 45/50
39/39 [==============================] - 91s 2s/step - loss: 1.0584 - acc:
0.5974 - val_loss: 1.3186 - val_acc: 0.4943 - lr: 7.5000e-05
Epoch 46/50
39/39 [==============================] - 92s 2s/step - loss: 1.0611 - acc:
0.5914 - val_loss: 1.2842 - val_acc: 0.5057 - lr: 7.5000e-05
Epoch 47/50
39/39 [==============================] - 92s 2s/step - loss: 1.0554 - acc:
0.6013 - val_loss: 1.2967 - val_acc: 0.5096 - lr: 7.5000e-05
Epoch 48/50
39/39 [==============================] - 92s 2s/step - loss: 1.0637 - acc:
0.5968 - val_loss: 1.2619 - val_acc: 0.5249 - lr: 7.5000e-05
Epoch 49/50
39/39 [==============================] - ETA: 0s - loss: 1.0475 - acc: 0.5984
Epoch 00049: ReduceLROnPlateau reducing learning rate to 5.6249997214763425e-05.
39/39 [==============================] - 567s 15s/step - loss: 1.0475 - acc:
0.5984 - val_loss: 1.3311 - val_acc: 0.4713 - lr: 7.5000e-05
Epoch 50/50
39/39 [==============================] - 110s 3s/step - loss: 1.0432 - acc:
0.6065 - val_loss: 1.2516 - val_acc: 0.5211 - lr: 5.6250e-05
```

```
[ ]: histopt = modelopt.fit(features_train, y_train, batch_size=128, epochs=50,␣
      ↪verbose=1, validation_data=(features_val, y_val),␣
      ↪callbacks=[early_stop,lr_reduction])
```

0.8 Big Picture (20 Points)

1. TESTING THE FIRST MODEL ON 1-DIMENSIONAL FEATURE VECTORS

We compute the accuracy, f1-score, precision, recall and loss on test data based on our model.

```
[ ]: loss,accuracy=modelopt.evaluate(features_test,y_test,verbose=0)
     print(f'Test Loss: {loss}')
     print(f'Test Accuracy: {accuracy*100}%')
```

```
Test Loss: 1.2662875652313232
Test Accuracy: 52.53022909164429%
```

```
[ ]: import plotly.express as px
     fig=px.line(histopt.history,y=['acc','val_acc'],
                 labels={'index':'epoch','value':'acc'},
                 title=f'Epoch accuracy and validation accuracy chart for the model')
     fig.show()
```

```
[ ]: fig=px.line(histopt.history,y=['loss','val_loss'],
                 labels={'index':'epoch','value':'loss'},
                 title=(f'Epoch loss and validation loss chart for the model'))
     fig.show()
```

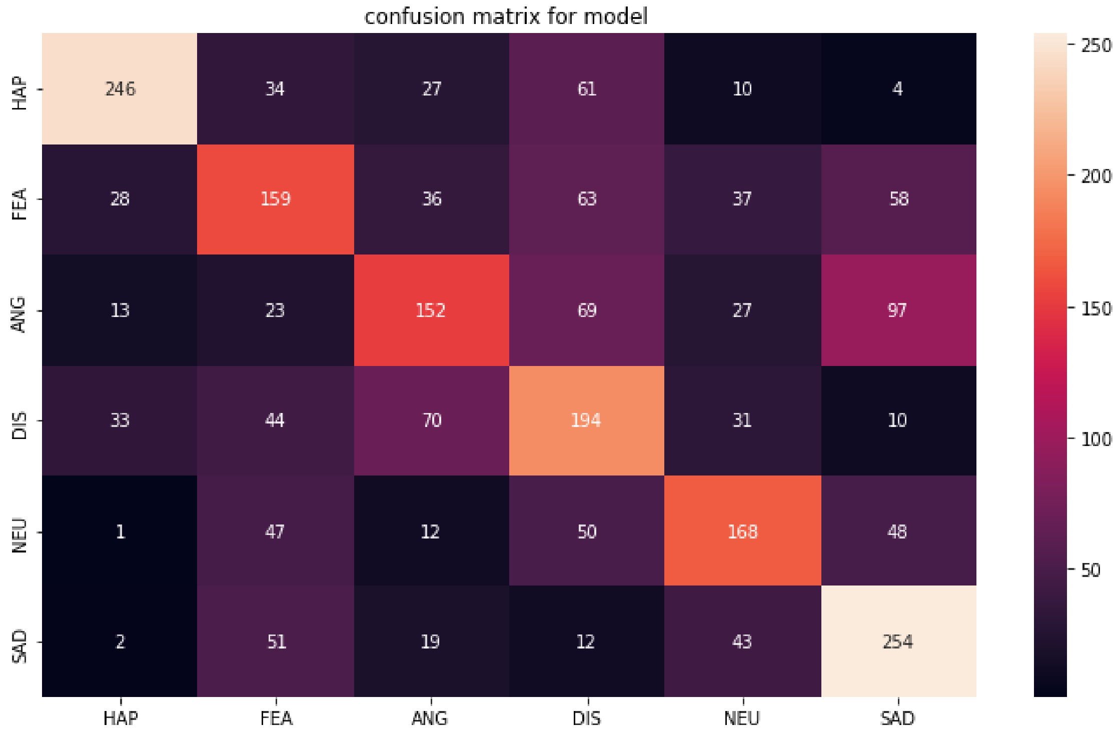
```
[ ]: import plotly.express as px
     fig=px.line(histopt.history,y=['acc','val_acc'],
                 labels={'index':'epoch','value':'acc'},
                 title=f'Epoch accuracy and validation accuracy chart for the model')
     fig.show()
```

```
[ ]: fig=px.line(histopt.history,y=['loss','val_loss'],
                 labels={'index':'epoch','value':'loss'},
                 title=(f'Epoch loss and validation loss chart for the model'))
     fig.show()
```

```
[ ]: from sklearn.metrics import confusion_matrix
     from sklearn.metrics import classification_report
     y_pred = modelopt.predict(features_test)
     y_pred = np.argmax(y_pred, axis=1)
     conf=confusion_matrix(y_test,y_pred)
     cm=pd.DataFrame(
         conf,index=[i for i in set(labels)],
         columns=[i for i in set(labels)]
     )
```

```
plt.figure(figsize=(12,7))
ax=sns.heatmap(cm,annot=True,fmt='d')
ax.set_title(f'confusion matrix for model ')
plt.show()
```

```
70/70 [==============================] - 19s 239ms/step
```




```
[ ]: from sklearn.metrics import f1_score,precision_score,recall_score
prec=precision_score(y_test,y_pred,average='weighted')
rec=recall_score(y_test,y_pred,average='weighted')
f1score=f1_score(y_test,y_pred,average='weighted')
print(f'Test precision: {prec}')
print(f'Test recall: {rec}')
print(f'Test f1-score: {f1score}')
```

```
Test precision: 0.5316831795330076
Test recall: 0.5253022839229736
Test f1-score: 0.5252374785837246
```

2. TESTING THE SECOND MODEL ON 1-DIMENSIONAL FEATURE VECTORS

```
[ ]: hist = model.fit(features_train, y_train, batch_size=128, epochs=100,
  ↪verbose=1, validation_data=(features_val, y_val),
  ↪callbacks=[early_stop,lr_reduction])
```

```
Epoch 1/100
39/39 [==============================] - 23s 450ms/step - loss: 1.9096 - acc:
0.3013 - val_loss: 5.0133 - val_acc: 0.1724 - lr: 1.0000e-04
Epoch 2/100
39/39 [==============================] - 17s 433ms/step - loss: 1.7263 - acc:
0.3373 - val_loss: 2.0859 - val_acc: 0.2299 - lr: 1.0000e-04
Epoch 3/100
39/39 [==============================] - 17s 443ms/step - loss: 1.6632 - acc:
0.3559 - val_loss: 1.8614 - val_acc: 0.1724 - lr: 1.0000e-04
Epoch 4/100
39/39 [==============================] - 17s 430ms/step - loss: 1.6164 - acc:
0.3612 - val_loss: 1.6996 - val_acc: 0.2567 - lr: 1.0000e-04
Epoch 5/100
39/39 [==============================] - 17s 433ms/step - loss: 1.5683 - acc:
0.3775 - val_loss: 1.6211 - val_acc: 0.3333 - lr: 1.0000e-04
Epoch 6/100
39/39 [==============================] - 17s 426ms/step - loss: 1.5800 - acc:
0.3705 - val_loss: 1.5418 - val_acc: 0.3180 - lr: 1.0000e-04
Epoch 7/100
39/39 [==============================] - 17s 429ms/step - loss: 1.5425 - acc:
0.3800 - val_loss: 1.5074 - val_acc: 0.3678 - lr: 1.0000e-04
Epoch 8/100
39/39 [==============================] - 17s 438ms/step - loss: 1.5215 - acc:
0.3789 - val_loss: 1.5081 - val_acc: 0.3563 - lr: 1.0000e-04
Epoch 9/100
39/39 [==============================] - 17s 438ms/step - loss: 1.5213 - acc:
0.3899 - val_loss: 1.5001 - val_acc: 0.3487 - lr: 1.0000e-04
Epoch 10/100
39/39 [==============================] - 17s 428ms/step - loss: 1.4954 - acc:
0.3965 - val_loss: 1.4750 - val_acc: 0.3755 - lr: 1.0000e-04
Epoch 11/100
39/39 [==============================] - 18s 453ms/step - loss: 1.4971 - acc:
0.3897 - val_loss: 1.4807 - val_acc: 0.4023 - lr: 1.0000e-04
Epoch 12/100
39/39 [==============================] - 18s 452ms/step - loss: 1.4854 - acc:
0.3921 - val_loss: 1.4441 - val_acc: 0.4100 - lr: 1.0000e-04
Epoch 13/100
39/39 [==============================] - 17s 441ms/step - loss: 1.4871 - acc:
0.3812 - val_loss: 1.5706 - val_acc: 0.3716 - lr: 1.0000e-04
Epoch 14/100
39/39 [==============================] - 17s 434ms/step - loss: 1.4663 - acc:
0.3955 - val_loss: 1.4787 - val_acc: 0.3678 - lr: 1.0000e-04
Epoch 15/100
39/39 [==============================] - 18s 465ms/step - loss: 1.4631 - acc:
0.4022 - val_loss: 1.4492 - val_acc: 0.3793 - lr: 1.0000e-04
Epoch 16/100
39/39 [==============================] - 19s 482ms/step - loss: 1.4561 - acc:
0.4004 - val_loss: 1.4540 - val_acc: 0.3908 - lr: 1.0000e-04
```

```
Epoch 17/100
39/39 [==============================] - 19s 496ms/step - loss: 1.4445 - acc:
0.4044 - val_loss: 1.5814 - val_acc: 0.3333 - lr: 1.0000e-04
Epoch 18/100
39/39 [==============================] - 20s 513ms/step - loss: 1.4543 - acc:
0.4000 - val_loss: 1.4826 - val_acc: 0.3870 - lr: 1.0000e-04
Epoch 19/100
39/39 [==============================] - 20s 503ms/step - loss: 1.4348 - acc:
0.4115 - val_loss: 1.4606 - val_acc: 0.3716 - lr: 1.0000e-04
Epoch 20/100
39/39 [==============================] - 19s 478ms/step - loss: 1.4196 - acc:
0.4121 - val_loss: 1.4335 - val_acc: 0.3870 - lr: 1.0000e-04
Epoch 21/100
39/39 [==============================] - 18s 466ms/step - loss: 1.4214 - acc:
0.4192 - val_loss: 1.4160 - val_acc: 0.4138 - lr: 1.0000e-04
Epoch 22/100
39/39 [==============================] - 17s 443ms/step - loss: 1.4150 - acc:
0.4198 - val_loss: 1.4437 - val_acc: 0.4253 - lr: 1.0000e-04
Epoch 23/100
39/39 [==============================] - 17s 437ms/step - loss: 1.4100 - acc:
0.4276 - val_loss: 1.5748 - val_acc: 0.3218 - lr: 1.0000e-04
Epoch 24/100
39/39 [==============================] - 17s 439ms/step - loss: 1.4059 - acc:
0.4361 - val_loss: 1.4105 - val_acc: 0.4483 - lr: 1.0000e-04
Epoch 25/100
39/39 [==============================] - 17s 441ms/step - loss: 1.4214 - acc:
0.4186 - val_loss: 1.3992 - val_acc: 0.4368 - lr: 1.0000e-04
Epoch 26/100
39/39 [==============================] - 17s 444ms/step - loss: 1.3990 - acc:
0.4266 - val_loss: 1.4498 - val_acc: 0.3831 - lr: 1.0000e-04
Epoch 27/100
39/39 [==============================] - 17s 437ms/step - loss: 1.3942 - acc:
0.4327 - val_loss: 1.4232 - val_acc: 0.3946 - lr: 1.0000e-04
Epoch 28/100
39/39 [==============================] - 17s 440ms/step - loss: 1.3725 - acc:
0.4418 - val_loss: 1.4173 - val_acc: 0.4100 - lr: 1.0000e-04
Epoch 29/100
39/39 [==============================] - 18s 455ms/step - loss: 1.3771 - acc:
0.4337 - val_loss: 1.4052 - val_acc: 0.4253 - lr: 1.0000e-04
Epoch 30/100
39/39 [==============================] - 22s 561ms/step - loss: 1.3780 - acc:
0.4329 - val_loss: 1.3921 - val_acc: 0.4215 - lr: 1.0000e-04
Epoch 31/100
39/39 [==============================] - 18s 471ms/step - loss: 1.3802 - acc:
0.4307 - val_loss: 1.4065 - val_acc: 0.4291 - lr: 1.0000e-04
Epoch 32/100
39/39 [==============================] - 18s 459ms/step - loss: 1.3698 - acc:
0.4460 - val_loss: 1.3753 - val_acc: 0.4444 - lr: 1.0000e-04
```

```
Epoch 33/100
39/39 [==============================] - 17s 443ms/step - loss: 1.3834 - acc:
0.4335 - val_loss: 1.4012 - val_acc: 0.4368 - lr: 1.0000e-04
Epoch 34/100
39/39 [==============================] - ETA: 0s - loss: 1.3730 - acc: 0.4378
Epoch 00034: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-05.
39/39 [==============================] - 17s 439ms/step - loss: 1.3730 - acc:
0.4378 - val_loss: 1.4264 - val_acc: 0.4023 - lr: 1.0000e-04
Epoch 35/100
39/39 [==============================] - 17s 442ms/step - loss: 1.3586 - acc:
0.4483 - val_loss: 1.4051 - val_acc: 0.4253 - lr: 5.0000e-05
Epoch 36/100
39/39 [==============================] - 17s 445ms/step - loss: 1.3429 - acc:
0.4572 - val_loss: 1.4039 - val_acc: 0.4368 - lr: 5.0000e-05
Epoch 37/100
39/39 [==============================] - 17s 443ms/step - loss: 1.3374 - acc:
0.4450 - val_loss: 1.3481 - val_acc: 0.4406 - lr: 5.0000e-05
Epoch 38/100
39/39 [==============================] - 17s 442ms/step - loss: 1.3403 - acc:
0.4503 - val_loss: 1.3652 - val_acc: 0.4751 - lr: 5.0000e-05
Epoch 39/100
39/39 [==============================] - 17s 439ms/step - loss: 1.3268 - acc:
0.4624 - val_loss: 1.3727 - val_acc: 0.4674 - lr: 5.0000e-05
Epoch 40/100
39/39 [==============================] - 17s 438ms/step - loss: 1.3262 - acc:
0.4675 - val_loss: 1.3795 - val_acc: 0.4138 - lr: 5.0000e-05
Epoch 41/100
39/39 [==============================] - 17s 439ms/step - loss: 1.3196 - acc:
0.4600 - val_loss: 1.4155 - val_acc: 0.4023 - lr: 5.0000e-05
Epoch 42/100
39/39 [==============================] - 17s 447ms/step - loss: 1.3127 - acc:
0.4739 - val_loss: 1.3670 - val_acc: 0.4368 - lr: 5.0000e-05
Epoch 43/100
39/39 [==============================] - 17s 443ms/step - loss: 1.3204 - acc:
0.4656 - val_loss: 1.3820 - val_acc: 0.4291 - lr: 5.0000e-05
Epoch 44/100
39/39 [==============================] - 17s 441ms/step - loss: 1.3144 - acc:
0.4667 - val_loss: 1.4513 - val_acc: 0.3793 - lr: 5.0000e-05
Epoch 45/100
39/39 [==============================] - 18s 459ms/step - loss: 1.3115 - acc:
0.4665 - val_loss: 1.3616 - val_acc: 0.4598 - lr: 5.0000e-05
Epoch 46/100
39/39 [==============================] - 18s 453ms/step - loss: 1.3057 - acc:
0.4762 - val_loss: 1.3781 - val_acc: 0.4483 - lr: 5.0000e-05
Epoch 47/100
39/39 [==============================] - 17s 445ms/step - loss: 1.3065 - acc:
0.4784 - val_loss: 1.3501 - val_acc: 0.4598 - lr: 5.0000e-05
Epoch 48/100
```

```
39/39 [==============================] - ETA: 0s - loss: 1.3004 - acc: 0.4747
Epoch 00048: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
39/39 [==============================] - 17s 444ms/step - loss: 1.3004 - acc:
0.4747 - val_loss: 1.3730 - val_acc: 0.4100 - lr: 5.0000e-05
Epoch 49/100
39/39 [==============================] - 17s 448ms/step - loss: 1.2847 - acc:
0.4852 - val_loss: 1.3641 - val_acc: 0.4444 - lr: 2.5000e-05
Epoch 50/100
39/39 [==============================] - 19s 492ms/step - loss: 1.2709 - acc:
0.4871 - val_loss: 1.3500 - val_acc: 0.4789 - lr: 2.5000e-05
Epoch 51/100
39/39 [==============================] - 18s 451ms/step - loss: 1.2664 - acc:
0.4913 - val_loss: 1.3780 - val_acc: 0.4215 - lr: 2.5000e-05
Epoch 52/100
39/39 [==============================] - 17s 448ms/step - loss: 1.2612 - acc:
0.4883 - val_loss: 1.3833 - val_acc: 0.4061 - lr: 2.5000e-05
Epoch 53/100
39/39 [==============================] - 17s 444ms/step - loss: 1.2643 - acc:
0.4905 - val_loss: 1.4071 - val_acc: 0.4253 - lr: 2.5000e-05
Epoch 54/100
39/39 [==============================] - 18s 452ms/step - loss: 1.2534 - acc:
0.5020 - val_loss: 1.3653 - val_acc: 0.4483 - lr: 2.5000e-05
Epoch 55/100
39/39 [==============================] - 18s 468ms/step - loss: 1.2508 - acc:
0.5002 - val_loss: 1.3916 - val_acc: 0.4521 - lr: 2.5000e-05
Epoch 56/100
39/39 [==============================] - 17s 443ms/step - loss: 1.2366 - acc:
0.5044 - val_loss: 1.3818 - val_acc: 0.4330 - lr: 2.5000e-05
Epoch 57/100
39/39 [==============================] - 19s 479ms/step - loss: 1.2433 - acc:
0.5010 - val_loss: 1.3588 - val_acc: 0.4636 - lr: 2.5000e-05
Epoch 58/100
39/39 [==============================] - 19s 474ms/step - loss: 1.2355 - acc:
0.5067 - val_loss: 1.3703 - val_acc: 0.4406 - lr: 2.5000e-05
Epoch 59/100
39/39 [==============================] - 18s 461ms/step - loss: 1.2369 - acc:
0.4956 - val_loss: 1.3678 - val_acc: 0.4559 - lr: 2.5000e-05
Epoch 60/100
39/39 [==============================] - ETA: 0s - loss: 1.2284 - acc: 0.5087
Epoch 00060: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-05.
39/39 [==============================] - 18s 463ms/step - loss: 1.2284 - acc:
0.5087 - val_loss: 1.3908 - val_acc: 0.4253 - lr: 2.5000e-05
Epoch 61/100
39/39 [==============================] - 18s 452ms/step - loss: 1.2170 - acc:
0.5135 - val_loss: 1.3579 - val_acc: 0.4406 - lr: 1.2500e-05
Epoch 62/100
39/39 [==============================] - 18s 457ms/step - loss: 1.2091 - acc:
0.5182 - val_loss: 1.3775 - val_acc: 0.4483 - lr: 1.2500e-05
```

```
Epoch 63/100
39/39 [==============================] - 17s 448ms/step - loss: 1.2054 - acc:
0.5188 - val_loss: 1.3598 - val_acc: 0.4521 - lr: 1.2500e-05
Epoch 64/100
39/39 [==============================] - 18s 449ms/step - loss: 1.2069 - acc:
0.5168 - val_loss: 1.3858 - val_acc: 0.4559 - lr: 1.2500e-05
Epoch 65/100
39/39 [==============================] - 20s 511ms/step - loss: 1.2043 - acc:
0.5180 - val_loss: 1.3824 - val_acc: 0.4291 - lr: 1.2500e-05
Epoch 66/100
39/39 [==============================] - 19s 499ms/step - loss: 1.1927 - acc:
0.5224 - val_loss: 1.3670 - val_acc: 0.4713 - lr: 1.2500e-05
Epoch 67/100
39/39 [==============================] - 17s 443ms/step - loss: 1.1929 - acc:
0.5261 - val_loss: 1.3805 - val_acc: 0.3985 - lr: 1.2500e-05
Epoch 68/100
39/39 [==============================] - 17s 438ms/step - loss: 1.1970 - acc:
0.5176 - val_loss: 1.3579 - val_acc: 0.4483 - lr: 1.2500e-05
Epoch 69/100
39/39 [==============================] - 17s 441ms/step - loss: 1.1962 - acc:
0.5232 - val_loss: 1.3822 - val_acc: 0.4444 - lr: 1.2500e-05
Epoch 70/100
39/39 [==============================] - ETA: 0s - loss: 1.1823 - acc: 0.5342
Epoch 00070: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-06.
39/39 [==============================] - 17s 444ms/step - loss: 1.1823 - acc:
0.5342 - val_loss: 1.3690 - val_acc: 0.4444 - lr: 1.2500e-05
```

[ ]:
```
import plotly.express as px
fig=px.line(hist.history,y=['acc','val_acc'],
           labels={'index':'epoch','value':'acc'},
           title=f'According to the epoch accuracy and validation accuracy␣
 ↪chart for the model')
fig.show()
```

[ ]:
```
fig=px.line(hist.history,y=['loss','val_loss'],
           labels={'index':'epoch','value':'loss'},
           title=f'According to the epoch loss and validation loss chart for␣
 ↪the model')
fig.show()
```

[ ]:
```
hist2 = model.fit(features_train, y_train, batch_size=64, epochs=50, verbose=1,␣
 ↪validation_data=(features_val, y_val), callbacks=[early_stop,lr_reduction])
```

```
Epoch 1/50
78/78 [==============================] - 167s 2s/step - loss: 1.9794 - acc:
0.2850 - val_loss: 2.4579 - val_acc: 0.2452 - lr: 1.0000e-04
Epoch 2/50
78/78 [==============================] - 167s 2s/step - loss: 1.7202 - acc:
```

```
0.3432 - val_loss: 1.9398 - val_acc: 0.2452 - lr: 1.0000e-04
Epoch 3/50
78/78 [==============================] - 169s 2s/step - loss: 1.6271 - acc:
0.3652 - val_loss: 1.7943 - val_acc: 0.2375 - lr: 1.0000e-04
Epoch 4/50
78/78 [==============================] - 167s 2s/step - loss: 1.5450 - acc:
0.3884 - val_loss: 1.7942 - val_acc: 0.2912 - lr: 1.0000e-04
Epoch 5/50
78/78 [==============================] - 170s 2s/step - loss: 1.5145 - acc:
0.4014 - val_loss: 1.4952 - val_acc: 0.3602 - lr: 1.0000e-04
Epoch 6/50
78/78 [==============================] - 163s 2s/step - loss: 1.4731 - acc:
0.4101 - val_loss: 1.5317 - val_acc: 0.3602 - lr: 1.0000e-04
Epoch 7/50
78/78 [==============================] - 163s 2s/step - loss: 1.4392 - acc:
0.4169 - val_loss: 1.5331 - val_acc: 0.3333 - lr: 1.0000e-04
Epoch 8/50
78/78 [==============================] - 163s 2s/step - loss: 1.4100 - acc:
0.4337 - val_loss: 1.5030 - val_acc: 0.3793 - lr: 1.0000e-04
Epoch 9/50
78/78 [==============================] - 162s 2s/step - loss: 1.3856 - acc:
0.4424 - val_loss: 1.4835 - val_acc: 0.3755 - lr: 1.0000e-04
Epoch 10/50
78/78 [==============================] - 163s 2s/step - loss: 1.3561 - acc:
0.4545 - val_loss: 1.4623 - val_acc: 0.4215 - lr: 1.0000e-04
Epoch 11/50
78/78 [==============================] - 168s 2s/step - loss: 1.3426 - acc:
0.4636 - val_loss: 1.5009 - val_acc: 0.3870 - lr: 1.0000e-04
Epoch 12/50
78/78 [==============================] - 169s 2s/step - loss: 1.3316 - acc:
0.4656 - val_loss: 1.4026 - val_acc: 0.4215 - lr: 1.0000e-04
Epoch 13/50
78/78 [==============================] - 173s 2s/step - loss: 1.3084 - acc:
0.4755 - val_loss: 1.4503 - val_acc: 0.3678 - lr: 1.0000e-04
Epoch 14/50
78/78 [==============================] - 153s 2s/step - loss: 1.3112 - acc:
0.4832 - val_loss: 1.4790 - val_acc: 0.4138 - lr: 1.0000e-04
Epoch 15/50
78/78 [==============================] - 146s 2s/step - loss: 1.2806 - acc:
0.4956 - val_loss: 1.3788 - val_acc: 0.4406 - lr: 1.0000e-04
Epoch 16/50
78/78 [==============================] - 170s 2s/step - loss: 1.2738 - acc:
0.4897 - val_loss: 1.4003 - val_acc: 0.4674 - lr: 1.0000e-04
Epoch 17/50
78/78 [==============================] - 147s 2s/step - loss: 1.2495 - acc:
0.5032 - val_loss: 1.4312 - val_acc: 0.3985 - lr: 1.0000e-04
Epoch 18/50
78/78 [==============================] - 147s 2s/step - loss: 1.2283 - acc:
```

```
0.5119 - val_loss: 1.4717 - val_acc: 0.3908 - lr: 1.0000e-04
Epoch 19/50
78/78 [==============================] - 144s 2s/step - loss: 1.2026 - acc:
0.5212 - val_loss: 1.4051 - val_acc: 0.4444 - lr: 1.0000e-04
Epoch 20/50
78/78 [==============================] - 159s 2s/step - loss: 1.1922 - acc:
0.5261 - val_loss: 1.3725 - val_acc: 0.4521 - lr: 1.0000e-04
Epoch 21/50
78/78 [==============================] - ETA: 0s - loss: 1.1762 - acc: 0.5325
Epoch 00021: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-05.
78/78 [==============================] - 165s 2s/step - loss: 1.1762 - acc:
0.5325 - val_loss: 1.3543 - val_acc: 0.4330 - lr: 1.0000e-04
Epoch 22/50
78/78 [==============================] - 169s 2s/step - loss: 1.1310 - acc:
0.5588 - val_loss: 1.4055 - val_acc: 0.4138 - lr: 5.0000e-05
Epoch 23/50
78/78 [==============================] - 163s 2s/step - loss: 1.0941 - acc:
0.5649 - val_loss: 1.3242 - val_acc: 0.5019 - lr: 5.0000e-05
Epoch 24/50
78/78 [==============================] - 159s 2s/step - loss: 1.0742 - acc:
0.5825 - val_loss: 1.3460 - val_acc: 0.4636 - lr: 5.0000e-05
Epoch 25/50
78/78 [==============================] - 172s 2s/step - loss: 1.0549 - acc:
0.5823 - val_loss: 1.2937 - val_acc: 0.4751 - lr: 5.0000e-05
Epoch 26/50
78/78 [==============================] - 154s 2s/step - loss: 1.0298 - acc:
0.6051 - val_loss: 1.4161 - val_acc: 0.4406 - lr: 5.0000e-05
Epoch 27/50
78/78 [==============================] - 155s 2s/step - loss: 1.0186 - acc:
0.6043 - val_loss: 1.2856 - val_acc: 0.5172 - lr: 5.0000e-05
Epoch 28/50
78/78 [==============================] - 155s 2s/step - loss: 1.0078 - acc:
0.6085 - val_loss: 1.3790 - val_acc: 0.4483 - lr: 5.0000e-05
Epoch 29/50
78/78 [==============================] - 153s 2s/step - loss: 0.9803 - acc:
0.6318 - val_loss: 1.4974 - val_acc: 0.4176 - lr: 5.0000e-05
Epoch 30/50
78/78 [==============================] - 153s 2s/step - loss: 0.9711 - acc:
0.6261 - val_loss: 1.3993 - val_acc: 0.4521 - lr: 5.0000e-05
Epoch 31/50
78/78 [==============================] - 153s 2s/step - loss: 0.9475 - acc:
0.6372 - val_loss: 1.3605 - val_acc: 0.4751 - lr: 5.0000e-05
Epoch 32/50
78/78 [==============================] - ETA: 0s - loss: 0.9601 - acc: 0.6308
Epoch 00032: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
78/78 [==============================] - 153s 2s/step - loss: 0.9601 - acc:
0.6308 - val_loss: 1.4399 - val_acc: 0.4368 - lr: 5.0000e-05
Epoch 33/50
```

```
78/78 [==============================] - 153s 2s/step - loss: 0.8949 - acc:
0.6681 - val_loss: 1.3813 - val_acc: 0.4598 - lr: 2.5000e-05
Epoch 34/50
78/78 [==============================] - 153s 2s/step - loss: 0.8526 - acc:
0.6886 - val_loss: 1.3938 - val_acc: 0.4368 - lr: 2.5000e-05
Epoch 35/50
78/78 [==============================] - 154s 2s/step - loss: 0.8350 - acc:
0.6886 - val_loss: 1.3697 - val_acc: 0.4444 - lr: 2.5000e-05
Epoch 36/50
78/78 [==============================] - 153s 2s/step - loss: 0.8131 - acc:
0.7025 - val_loss: 1.4371 - val_acc: 0.4176 - lr: 2.5000e-05
Epoch 37/50
78/78 [==============================] - ETA: 0s - loss: 0.7939 - acc: 0.7100
Epoch 00037: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-05.
78/78 [==============================] - 153s 2s/step - loss: 0.7939 - acc:
0.7100 - val_loss: 1.3735 - val_acc: 0.4904 - lr: 2.5000e-05
Epoch 38/50
78/78 [==============================] - 154s 2s/step - loss: 0.7610 - acc:
0.7270 - val_loss: 1.4090 - val_acc: 0.4713 - lr: 1.2500e-05
Epoch 39/50
78/78 [==============================] - 153s 2s/step - loss: 0.7336 - acc:
0.7346 - val_loss: 1.3952 - val_acc: 0.4598 - lr: 1.2500e-05
Epoch 40/50
78/78 [==============================] - 154s 2s/step - loss: 0.7290 - acc:
0.7371 - val_loss: 1.4452 - val_acc: 0.4789 - lr: 1.2500e-05
Epoch 41/50
78/78 [==============================] - 153s 2s/step - loss: 0.7064 - acc:
0.7506 - val_loss: 1.4392 - val_acc: 0.4828 - lr: 1.2500e-05
Epoch 42/50
78/78 [==============================] - ETA: 0s - loss: 0.6931 - acc: 0.7510
Epoch 00042: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-06.
78/78 [==============================] - 162s 2s/step - loss: 0.6931 - acc:
0.7510 - val_loss: 1.4309 - val_acc: 0.4598 - lr: 1.2500e-05
Epoch 43/50
78/78 [==============================] - 163s 2s/step - loss: 0.6755 - acc:
0.7593 - val_loss: 1.4500 - val_acc: 0.4483 - lr: 6.2500e-06
Epoch 44/50
78/78 [==============================] - 159s 2s/step - loss: 0.6569 - acc:
0.7680 - val_loss: 1.3964 - val_acc: 0.5019 - lr: 6.2500e-06
Epoch 45/50
78/78 [==============================] - 160s 2s/step - loss: 0.6571 - acc:
0.7668 - val_loss: 1.3912 - val_acc: 0.4943 - lr: 6.2500e-06
Epoch 46/50
78/78 [==============================] - 160s 2s/step - loss: 0.6436 - acc:
0.7771 - val_loss: 1.3933 - val_acc: 0.4713 - lr: 6.2500e-06
Epoch 47/50
78/78 [==============================] - ETA: 0s - loss: 0.6565 - acc: 0.7658
Epoch 00047: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-06.
```

```
78/78 [==============================] - 159s 2s/step - loss: 0.6565 - acc:
0.7658 - val_loss: 1.4162 - val_acc: 0.4713 - lr: 6.2500e-06
```

[ ]:
```
fig=px.line(hist2.history,y=['acc','val_acc'],
           labels={'index':'epoch','value':'acc'},
           title=f'According to the epoch accuracy and validation accuracy␣
 ↪chart for the model')
fig.show()
```

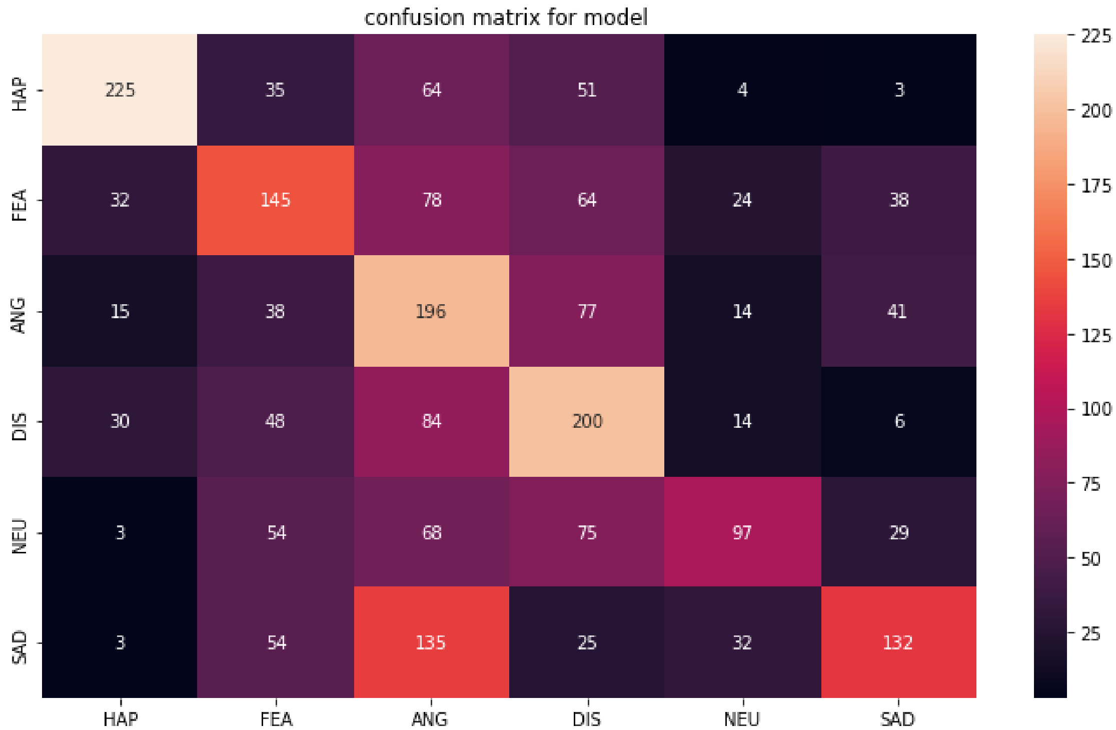
[ ]:
```
fig=px.line(hist2.history,y=['loss','val_loss'],
           labels={'index':'epoch','value':'loss'},
           title=f'According to the epoch loss and validation loss chart for␣
 ↪the model')
fig.show()
```

[ ]:
```
loss,accuracy=model.evaluate(features_test,y_test,verbose=0)
print(f'Test Loss: {loss}')
print(f'Test Accuracy: {accuracy}')
```

```
Test Loss: 1.3888615369796753
Test Accuracy: 0.4455888867378235
```

[ ]:
```
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
y_pred = model.predict(features_test)
y_pred = np.argmax(y_pred, axis=1)
conf=confusion_matrix(y_test,y_pred)
cm=pd.DataFrame(
    conf,index=[i for i in set(labels)],
    columns=[i for i in set(labels)]
)
plt.figure(figsize=(12,7))
ax=sns.heatmap(cm,annot=True,fmt='d')
ax.set_title(f'confusion matrix for model ')
plt.show()
```

```
70/70 [==============================] - 14s 192ms/step
```



[938]:
```
model2D=tf.keras.Sequential([
    L.Conv2D(32,kernel_size=5, strides=1,padding='same',␣
↪activation='relu',input_shape=(mels_train.shape[1],mels_train.shape[2],1)),
    L.BatchNormalization(),
    L.MaxPool2D(pool_size=5,strides=2,padding='same'),
    L.Conv2D(64,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool2D(pool_size=5,strides=2,padding='same'),
    # L.Conv2D(256,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.Conv2D(128,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool2D(pool_size=3,strides=2,padding='same'),
    L.Conv2D(256,kernel_size=3,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool2D(pool_size=3,strides=2,padding='same'),

    # L.Conv2D(256,kernel_size=3,strides=1,padding='same',activation='relu'),
    # L.BatchNormalization(),
    # L.MaxPool2D(pool_size=5,strides=2,padding='same'),
    # L.Conv2D(128,kernel_size=3,strides=1,padding='same',activation='relu'),
    # L.BatchNormalization(),
    # L.GlobalAveragePooling2D(),
```

```
    L.Flatten(),
    # L.Dropout(rate=0.1),
    # L.Dense(512,activation='relu'),
    L.Dense(512,activation='relu'),
    # L.BatchNormalization(),
    # L.Dense(64,activation='relu'),
    # L.BatchNormalization(),
    L.Dropout(0.3),
    L.Dense(6,activation='softmax')
])
# model2D.
 ↪compile(optimizer='adam',loss='sparse_categorical_crossentropy',metrics='acc')
# model2D.summary()
opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
model2D.compile(loss='sparse_categorical_crossentropy', optimizer=opt␣
 ↪,metrics=['acc'])
model2D.summary()
```

```
Model: "sequential_140"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_169 (Conv2D)         (None, 128, 69, 32)       832

 batch_normalization_521 (Ba  (None, 128, 69, 32)      128
 tchNormalization)

 max_pooling2d_143 (MaxPooli  (None, 64, 35, 32)       0
 ng2D)

 conv2d_170 (Conv2D)         (None, 64, 35, 64)        51264

 batch_normalization_522 (Ba  (None, 64, 35, 64)       256
 tchNormalization)

 max_pooling2d_144 (MaxPooli  (None, 32, 18, 64)       0
 ng2D)

 conv2d_171 (Conv2D)         (None, 32, 18, 128)       204928

 batch_normalization_523 (Ba  (None, 32, 18, 128)      512
 tchNormalization)

 max_pooling2d_145 (MaxPooli  (None, 16, 9, 128)       0
 ng2D)
```

```
 conv2d_172 (Conv2D)         (None, 16, 9, 256)        295168

 batch_normalization_524 (Ba  (None, 16, 9, 256)       1024
 tchNormalization)

 max_pooling2d_146 (MaxPooli  (None, 8, 5, 256)        0
 ng2D)

 flatten_69 (Flatten)        (None, 10240)             0

 dense_279 (Dense)           (None, 512)               5243392

 dropout_72 (Dropout)        (None, 512)               0

 dense_280 (Dense)           (None, 6)                 3078

=================================================================
Total params: 5,800,582
Trainable params: 5,799,622
Non-trainable params: 960
_________________________________________________________________
```

[939]:

```
early_stop=EarlyStopping(monitor='val_acc',mode='auto',patience=20,restore_best_weights=True)
lr_reduction=ReduceLROnPlateau(monitor='val_acc',patience=5,verbose=1,factor=0.
 ↪75,min_lr=0.000001)
```

[922]:

```
history2D = model2D.fit(mels_train, y_train, batch_size=128, epochs=70,␣
 ↪verbose=1, validation_data=(mels_val, y_val), callbacks = [early_stop,␣
 ↪lr_reduction])
```

```
Epoch 1/70
39/39 [==============================] - 118s 3s/step - loss: 88.2459 - acc:
0.3250 - val_loss: 84.9503 - val_acc: 0.3027 - lr: 1.0000e-04
Epoch 2/70
39/39 [==============================] - 112s 3s/step - loss: 80.0816 - acc:
0.3965 - val_loss: 76.0143 - val_acc: 0.3103 - lr: 1.0000e-04
Epoch 3/70
39/39 [==============================] - 117s 3s/step - loss: 71.8992 - acc:
0.4190 - val_loss: 68.1409 - val_acc: 0.2874 - lr: 1.0000e-04
Epoch 4/70
39/39 [==============================] - 122s 3s/step - loss: 63.9588 - acc:
0.4274 - val_loss: 60.2869 - val_acc: 0.2605 - lr: 1.0000e-04
Epoch 5/70
39/39 [==============================] - 122s 3s/step - loss: 56.3771 - acc:
0.4376 - val_loss: 52.8989 - val_acc: 0.3372 - lr: 1.0000e-04
Epoch 6/70
39/39 [==============================] - 121s 3s/step - loss: 49.3156 - acc:
0.4523 - val_loss: 46.1609 - val_acc: 0.3027 - lr: 1.0000e-04
```

```
Epoch 7/70
39/39 [==============================] - 121s 3s/step - loss: 42.8162 - acc:
0.4527 - val_loss: 39.9457 - val_acc: 0.2950 - lr: 1.0000e-04
Epoch 8/70
39/39 [==============================] - 120s 3s/step - loss: 36.8822 - acc:
0.4792 - val_loss: 34.3501 - val_acc: 0.3103 - lr: 1.0000e-04
Epoch 9/70
39/39 [==============================] - 122s 3s/step - loss: 31.5940 - acc:
0.4814 - val_loss: 29.4316 - val_acc: 0.3333 - lr: 1.0000e-04
Epoch 10/70
39/39 [==============================] - ETA: 0s - loss: 26.9128 - acc: 0.4905
Epoch 00010: ReduceLROnPlateau reducing learning rate to 4.999999873689376e-05.
39/39 [==============================] - 124s 3s/step - loss: 26.9128 - acc:
0.4905 - val_loss: 25.1072 - val_acc: 0.2950 - lr: 1.0000e-04
Epoch 11/70
39/39 [==============================] - 115s 3s/step - loss: 23.6820 - acc:
0.4984 - val_loss: 23.0734 - val_acc: 0.3487 - lr: 5.0000e-05
Epoch 12/70
39/39 [==============================] - 99s 3s/step - loss: 21.6802 - acc:
0.5253 - val_loss: 21.2599 - val_acc: 0.3372 - lr: 5.0000e-05
Epoch 13/70
39/39 [==============================] - 97s 2s/step - loss: 19.9207 - acc:
0.5317 - val_loss: 19.5197 - val_acc: 0.3410 - lr: 5.0000e-05
Epoch 14/70
39/39 [==============================] - 98s 3s/step - loss: 18.2971 - acc:
0.5358 - val_loss: 18.0314 - val_acc: 0.3218 - lr: 5.0000e-05
Epoch 15/70
39/39 [==============================] - 103s 3s/step - loss: 16.8446 - acc:
0.5350 - val_loss: 16.7022 - val_acc: 0.3333 - lr: 5.0000e-05
Epoch 16/70
39/39 [==============================] - ETA: 0s - loss: 15.5007 - acc: 0.5453
Epoch 00016: ReduceLROnPlateau reducing learning rate to 2.499999936844688e-05.
39/39 [==============================] - 109s 3s/step - loss: 15.5007 - acc:
0.5453 - val_loss: 15.3416 - val_acc: 0.3487 - lr: 5.0000e-05
Epoch 17/70
39/39 [==============================] - 109s 3s/step - loss: 14.4897 - acc:
0.5602 - val_loss: 14.7186 - val_acc: 0.3563 - lr: 2.5000e-05
Epoch 18/70
39/39 [==============================] - 102s 3s/step - loss: 13.8423 - acc:
0.5705 - val_loss: 14.0835 - val_acc: 0.3716 - lr: 2.5000e-05
Epoch 19/70
39/39 [==============================] - 102s 3s/step - loss: 13.2773 - acc:
0.5766 - val_loss: 13.4665 - val_acc: 0.3563 - lr: 2.5000e-05
Epoch 20/70
39/39 [==============================] - 97s 2s/step - loss: 12.7429 - acc:
0.5859 - val_loss: 12.9965 - val_acc: 0.3678 - lr: 2.5000e-05
Epoch 21/70
39/39 [==============================] - 95s 2s/step - loss: 12.2648 - acc:
```

```
0.5853 - val_loss: 12.4989 - val_acc: 0.3678 - lr: 2.5000e-05
Epoch 22/70
39/39 [==============================] - 100s 3s/step - loss: 11.8219 - acc:
0.5821 - val_loss: 12.0580 - val_acc: 0.3563 - lr: 2.5000e-05
Epoch 23/70
39/39 [==============================] - 101s 3s/step - loss: 11.3762 - acc:
0.5837 - val_loss: 11.7204 - val_acc: 0.3870 - lr: 2.5000e-05
Epoch 24/70
39/39 [==============================] - 101s 3s/step - loss: 10.9508 - acc:
0.5891 - val_loss: 11.2257 - val_acc: 0.3678 - lr: 2.5000e-05
Epoch 25/70
39/39 [==============================] - 103s 3s/step - loss: 10.5617 - acc:
0.5829 - val_loss: 10.7376 - val_acc: 0.3985 - lr: 2.5000e-05
Epoch 26/70
39/39 [==============================] - 102s 3s/step - loss: 10.1722 - acc:
0.5948 - val_loss: 10.4245 - val_acc: 0.3946 - lr: 2.5000e-05
Epoch 27/70
39/39 [==============================] - 102s 3s/step - loss: 9.8463 - acc:
0.5859 - val_loss: 10.1512 - val_acc: 0.3793 - lr: 2.5000e-05
Epoch 28/70
39/39 [==============================] - 113s 3s/step - loss: 9.5216 - acc:
0.6031 - val_loss: 9.9324 - val_acc: 0.4061 - lr: 2.5000e-05
Epoch 29/70
39/39 [==============================] - 105s 3s/step - loss: 9.2746 - acc:
0.5903 - val_loss: 9.5365 - val_acc: 0.4100 - lr: 2.5000e-05
Epoch 30/70
39/39 [==============================] - 105s 3s/step - loss: 8.9320 - acc:
0.6008 - val_loss: 9.2244 - val_acc: 0.4215 - lr: 2.5000e-05
Epoch 31/70
39/39 [==============================] - 103s 3s/step - loss: 8.6864 - acc:
0.6091 - val_loss: 9.0418 - val_acc: 0.4330 - lr: 2.5000e-05
Epoch 32/70
39/39 [==============================] - 103s 3s/step - loss: 8.3927 - acc:
0.6164 - val_loss: 8.8639 - val_acc: 0.3602 - lr: 2.5000e-05
Epoch 33/70
39/39 [==============================] - 100s 3s/step - loss: 8.1464 - acc:
0.6174 - val_loss: 8.6749 - val_acc: 0.3563 - lr: 2.5000e-05
Epoch 34/70
39/39 [==============================] - 94s 2s/step - loss: 7.8954 - acc:
0.6227 - val_loss: 8.5880 - val_acc: 0.3755 - lr: 2.5000e-05
Epoch 35/70
39/39 [==============================] - 105s 3s/step - loss: 7.7224 - acc:
0.6112 - val_loss: 8.1109 - val_acc: 0.4138 - lr: 2.5000e-05
Epoch 36/70
39/39 [==============================] - ETA: 0s - loss: 7.5364 - acc: 0.6150
Epoch 00036: ReduceLROnPlateau reducing learning rate to 1.249999968422344e-05.
39/39 [==============================] - 102s 3s/step - loss: 7.5364 - acc:
0.6150 - val_loss: 8.1473 - val_acc: 0.3870 - lr: 2.5000e-05
```

```
Epoch 37/70
39/39 [==============================] - 95s 2s/step - loss: 7.2876 - acc:
0.6336 - val_loss: 7.8307 - val_acc: 0.4138 - lr: 1.2500e-05
Epoch 38/70
39/39 [==============================] - 98s 3s/step - loss: 7.1252 - acc:
0.6409 - val_loss: 7.7058 - val_acc: 0.4253 - lr: 1.2500e-05
Epoch 39/70
39/39 [==============================] - 97s 2s/step - loss: 7.0328 - acc:
0.6358 - val_loss: 7.6868 - val_acc: 0.4291 - lr: 1.2500e-05
Epoch 40/70
39/39 [==============================] - 95s 2s/step - loss: 6.9453 - acc:
0.6413 - val_loss: 7.4741 - val_acc: 0.4100 - lr: 1.2500e-05
Epoch 41/70
39/39 [==============================] - ETA: 0s - loss: 6.8242 - acc: 0.6465
Epoch 00041: ReduceLROnPlateau reducing learning rate to 6.24999984211172e-06.
39/39 [==============================] - 96s 2s/step - loss: 6.8242 - acc:
0.6465 - val_loss: 7.3652 - val_acc: 0.4023 - lr: 1.2500e-05
Epoch 42/70
39/39 [==============================] - 88s 2s/step - loss: 6.7365 - acc:
0.6597 - val_loss: 7.3747 - val_acc: 0.4176 - lr: 6.2500e-06
Epoch 43/70
39/39 [==============================] - 88s 2s/step - loss: 6.6745 - acc:
0.6550 - val_loss: 7.2937 - val_acc: 0.4215 - lr: 6.2500e-06
Epoch 44/70
39/39 [==============================] - 102s 3s/step - loss: 6.6303 - acc:
0.6568 - val_loss: 7.2149 - val_acc: 0.4023 - lr: 6.2500e-06
Epoch 45/70
39/39 [==============================] - 101s 3s/step - loss: 6.5642 - acc:
0.6607 - val_loss: 7.2294 - val_acc: 0.4253 - lr: 6.2500e-06
Epoch 46/70
39/39 [==============================] - ETA: 0s - loss: 6.5481 - acc: 0.6562
Epoch 00046: ReduceLROnPlateau reducing learning rate to 3.12499992105586e-06.
39/39 [==============================] - 102s 3s/step - loss: 6.5481 - acc:
0.6562 - val_loss: 7.2276 - val_acc: 0.4023 - lr: 6.2500e-06
Epoch 47/70
39/39 [==============================] - 103s 3s/step - loss: 6.5098 - acc:
0.6603 - val_loss: 7.1733 - val_acc: 0.4176 - lr: 3.1250e-06
Epoch 48/70
39/39 [==============================] - 102s 3s/step - loss: 6.4842 - acc:
0.6647 - val_loss: 7.0787 - val_acc: 0.4100 - lr: 3.1250e-06
Epoch 49/70
39/39 [==============================] - 93s 2s/step - loss: 6.4605 - acc:
0.6649 - val_loss: 7.0740 - val_acc: 0.4061 - lr: 3.1250e-06
Epoch 50/70
39/39 [==============================] - 96s 2s/step - loss: 6.4300 - acc:
0.6603 - val_loss: 7.1167 - val_acc: 0.4023 - lr: 3.1250e-06
Epoch 51/70
39/39 [==============================] - ETA: 0s - loss: 6.4155 - acc: 0.6556
```

```
Epoch 00051: ReduceLROnPlateau reducing learning rate to 1.56249996052793e-06.
39/39 [==============================] - 101s 3s/step - loss: 6.4155 - acc:
0.6556 - val_loss: 7.1132 - val_acc: 0.4023 - lr: 3.1250e-06
```

[940]:

```python
# opt = tf.keras.optimizers.Adam(learning_rate=0.0001)
# model2D.compile(loss='sparse_categorical_crossentropy', optimizer=opt␣
 ↪,metrics=['acc'])
history2D = model2D.fit(mels_train, y_train, batch_size=128, epochs=50,␣
 ↪verbose=1, validation_data=(mels_val, y_val), callbacks = [early_stop,␣
 ↪lr_reduction])
```

```
Epoch 1/50
39/39 [==============================] - 110s 3s/step - loss: 2.1346 - acc:
0.3454 - val_loss: 4.3059 - val_acc: 0.2529 - lr: 1.0000e-04
Epoch 2/50
39/39 [==============================] - 109s 3s/step - loss: 1.6300 - acc:
0.3842 - val_loss: 1.6857 - val_acc: 0.3295 - lr: 1.0000e-04
Epoch 3/50
39/39 [==============================] - 110s 3s/step - loss: 1.4706 - acc:
0.4143 - val_loss: 1.6260 - val_acc: 0.3027 - lr: 1.0000e-04
Epoch 4/50
39/39 [==============================] - 109s 3s/step - loss: 1.4532 - acc:
0.4347 - val_loss: 1.7235 - val_acc: 0.2874 - lr: 1.0000e-04
Epoch 5/50
39/39 [==============================] - 105s 3s/step - loss: 1.4218 - acc:
0.4359 - val_loss: 1.5401 - val_acc: 0.3372 - lr: 1.0000e-04
Epoch 6/50
39/39 [==============================] - 102s 3s/step - loss: 1.3717 - acc:
0.4525 - val_loss: 1.6128 - val_acc: 0.2835 - lr: 1.0000e-04
Epoch 7/50
39/39 [==============================] - 102s 3s/step - loss: 1.3391 - acc:
0.4600 - val_loss: 1.6166 - val_acc: 0.3257 - lr: 1.0000e-04
Epoch 8/50
39/39 [==============================] - 102s 3s/step - loss: 1.3086 - acc:
0.4788 - val_loss: 1.6206 - val_acc: 0.3257 - lr: 1.0000e-04
Epoch 9/50
39/39 [==============================] - 101s 3s/step - loss: 1.2911 - acc:
0.4830 - val_loss: 1.6273 - val_acc: 0.3257 - lr: 1.0000e-04
Epoch 10/50
39/39 [==============================] - ETA: 0s - loss: 1.2400 - acc: 0.5067
Epoch 00010: ReduceLROnPlateau reducing learning rate to 7.499999810534064e-05.
39/39 [==============================] - 97s 2s/step - loss: 1.2400 - acc:
0.5067 - val_loss: 1.5841 - val_acc: 0.3372 - lr: 1.0000e-04
Epoch 11/50
39/39 [==============================] - 101s 3s/step - loss: 1.2279 - acc:
0.5101 - val_loss: 1.5841 - val_acc: 0.3410 - lr: 7.5000e-05
Epoch 12/50
39/39 [==============================] - 99s 3s/step - loss: 1.1861 - acc:
```

```
0.5236 - val_loss: 1.7961 - val_acc: 0.3180 - lr: 7.5000e-05
Epoch 13/50
39/39 [==============================] - 98s 3s/step - loss: 1.1555 - acc:
0.5483 - val_loss: 1.4447 - val_acc: 0.4483 - lr: 7.5000e-05
Epoch 14/50
39/39 [==============================] - 98s 2s/step - loss: 1.1183 - acc:
0.5578 - val_loss: 1.6422 - val_acc: 0.3793 - lr: 7.5000e-05
Epoch 15/50
39/39 [==============================] - 98s 3s/step - loss: 1.0936 - acc:
0.5831 - val_loss: 1.6323 - val_acc: 0.3908 - lr: 7.5000e-05
Epoch 16/50
39/39 [==============================] - 98s 3s/step - loss: 1.0701 - acc:
0.5837 - val_loss: 1.6098 - val_acc: 0.3755 - lr: 7.5000e-05
Epoch 17/50
39/39 [==============================] - 99s 3s/step - loss: 1.0462 - acc:
0.5889 - val_loss: 1.4675 - val_acc: 0.4483 - lr: 7.5000e-05
Epoch 18/50
39/39 [==============================] - ETA: 0s - loss: 1.0347 - acc: 0.5998
Epoch 00018: ReduceLROnPlateau reducing learning rate to 5.6249997214763425e-05.
39/39 [==============================] - 101s 3s/step - loss: 1.0347 - acc:
0.5998 - val_loss: 1.5620 - val_acc: 0.3640 - lr: 7.5000e-05
Epoch 19/50
39/39 [==============================] - 107s 3s/step - loss: 0.9899 - acc:
0.6209 - val_loss: 1.6272 - val_acc: 0.3908 - lr: 5.6250e-05
Epoch 20/50
39/39 [==============================] - 120s 3s/step - loss: 0.9588 - acc:
0.6332 - val_loss: 1.9032 - val_acc: 0.3333 - lr: 5.6250e-05
Epoch 21/50
39/39 [==============================] - 109s 3s/step - loss: 0.9472 - acc:
0.6300 - val_loss: 1.8207 - val_acc: 0.3870 - lr: 5.6250e-05
Epoch 22/50
39/39 [==============================] - 112s 3s/step - loss: 0.9061 - acc:
0.6512 - val_loss: 1.5841 - val_acc: 0.4138 - lr: 5.6250e-05
Epoch 23/50
39/39 [==============================] - ETA: 0s - loss: 0.8947 - acc: 0.6570
Epoch 00023: ReduceLROnPlateau reducing learning rate to 4.218749927531462e-05.
39/39 [==============================] - 117s 3s/step - loss: 0.8947 - acc:
0.6570 - val_loss: 1.6974 - val_acc: 0.3908 - lr: 5.6250e-05
Epoch 24/50
39/39 [==============================] - 114s 3s/step - loss: 0.8522 - acc:
0.6791 - val_loss: 1.6231 - val_acc: 0.3985 - lr: 4.2188e-05
Epoch 25/50
39/39 [==============================] - 105s 3s/step - loss: 0.8353 - acc:
0.6855 - val_loss: 1.5725 - val_acc: 0.4138 - lr: 4.2188e-05
Epoch 26/50
39/39 [==============================] - 102s 3s/step - loss: 0.8007 - acc:
0.7019 - val_loss: 1.8773 - val_acc: 0.3563 - lr: 4.2188e-05
Epoch 27/50
```

```
39/39 [==============================] - 103s 3s/step - loss: 0.7954 - acc:
0.7041 - val_loss: 1.6710 - val_acc: 0.4061 - lr: 4.2188e-05
Epoch 28/50
39/39 [==============================] - ETA: 0s - loss: 0.7809 - acc: 0.7001
Epoch 00028: ReduceLROnPlateau reducing learning rate to 3.164062582072802e-05.
39/39 [==============================] - 108s 3s/step - loss: 0.7809 - acc:
0.7001 - val_loss: 1.6689 - val_acc: 0.4100 - lr: 4.2188e-05
Epoch 29/50
39/39 [==============================] - 107s 3s/step - loss: 0.7531 - acc:
0.7209 - val_loss: 1.7262 - val_acc: 0.3870 - lr: 3.1641e-05
Epoch 30/50
39/39 [==============================] - 108s 3s/step - loss: 0.7288 - acc:
0.7262 - val_loss: 1.5889 - val_acc: 0.4521 - lr: 3.1641e-05
Epoch 31/50
39/39 [==============================] - 112s 3s/step - loss: 0.7395 - acc:
0.7268 - val_loss: 1.6569 - val_acc: 0.3946 - lr: 3.1641e-05
Epoch 32/50
39/39 [==============================] - 114s 3s/step - loss: 0.6962 - acc:
0.7395 - val_loss: 1.6024 - val_acc: 0.4521 - lr: 3.1641e-05
Epoch 33/50
39/39 [==============================] - 112s 3s/step - loss: 0.6754 - acc:
0.7492 - val_loss: 1.7905 - val_acc: 0.3602 - lr: 3.1641e-05
Epoch 34/50
39/39 [==============================] - 110s 3s/step - loss: 0.6683 - acc:
0.7540 - val_loss: 1.6673 - val_acc: 0.4406 - lr: 3.1641e-05
Epoch 35/50
39/39 [==============================] - ETA: 0s - loss: 0.6521 - acc: 0.7635
Epoch 00035: ReduceLROnPlateau reducing learning rate to 2.3730469365546014e-05.
39/39 [==============================] - 116s 3s/step - loss: 0.6521 - acc:
0.7635 - val_loss: 1.6467 - val_acc: 0.4406 - lr: 3.1641e-05
Epoch 36/50
39/39 [==============================] - 103s 3s/step - loss: 0.6409 - acc:
0.7690 - val_loss: 1.7007 - val_acc: 0.4100 - lr: 2.3730e-05
Epoch 37/50
39/39 [==============================] - 110s 3s/step - loss: 0.6335 - acc:
0.7656 - val_loss: 1.7215 - val_acc: 0.4061 - lr: 2.3730e-05
Epoch 38/50
39/39 [==============================] - 107s 3s/step - loss: 0.6249 - acc:
0.7650 - val_loss: 1.7074 - val_acc: 0.4291 - lr: 2.3730e-05
Epoch 39/50
39/39 [==============================] - 103s 3s/step - loss: 0.6077 - acc:
0.7821 - val_loss: 1.6960 - val_acc: 0.3946 - lr: 2.3730e-05
Epoch 40/50
39/39 [==============================] - 104s 3s/step - loss: 0.6071 - acc:
0.7732 - val_loss: 1.7206 - val_acc: 0.4636 - lr: 2.3730e-05
Epoch 41/50
39/39 [==============================] - 101s 3s/step - loss: 0.5743 - acc:
0.7928 - val_loss: 1.7366 - val_acc: 0.4023 - lr: 2.3730e-05
```

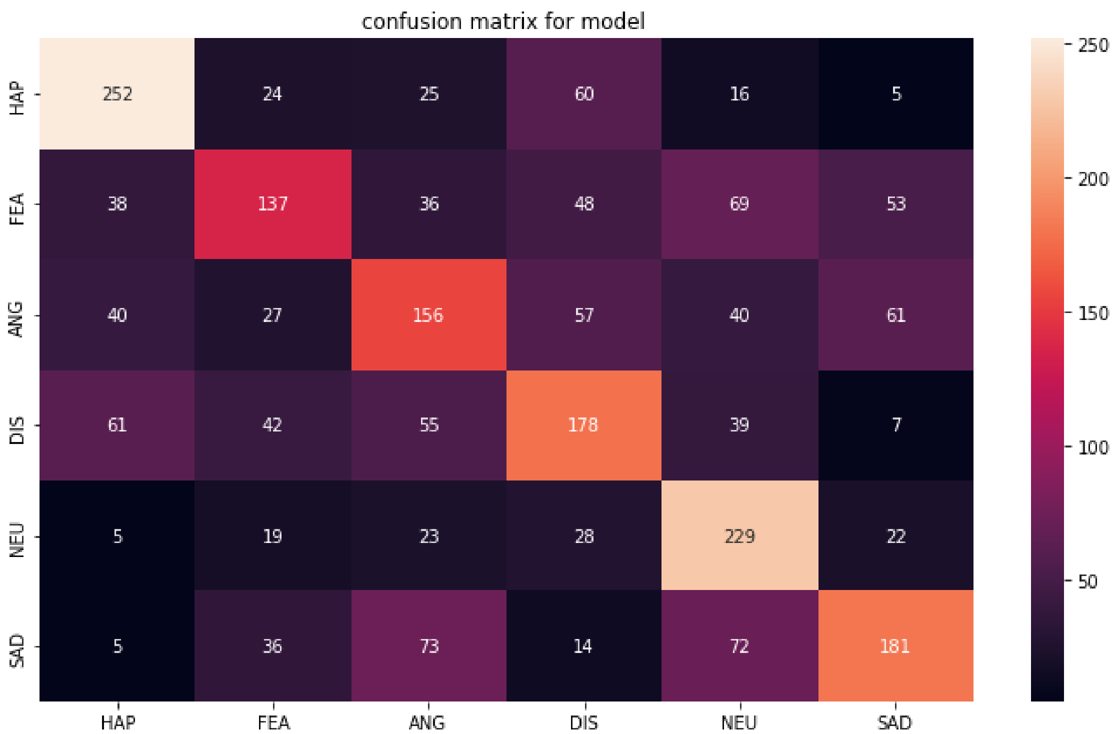
```
Epoch 42/50
39/39 [==============================] - 108s 3s/step - loss: 0.5833 - acc:
0.7878 - val_loss: 2.0084 - val_acc: 0.3563 - lr: 2.3730e-05
Epoch 43/50
39/39 [==============================] - 108s 3s/step - loss: 0.5705 - acc:
0.7955 - val_loss: 1.7986 - val_acc: 0.4291 - lr: 2.3730e-05
Epoch 44/50
39/39 [==============================] - 115s 3s/step - loss: 0.5426 - acc:
0.8080 - val_loss: 1.6625 - val_acc: 0.4598 - lr: 2.3730e-05
Epoch 45/50
39/39 [==============================] - ETA: 0s - loss: 0.5422 - acc: 0.7971
Epoch 00045: ReduceLROnPlateau reducing learning rate to 1.7797852706280537e-05.
39/39 [==============================] - 110s 3s/step - loss: 0.5422 - acc:
0.7971 - val_loss: 1.7655 - val_acc: 0.4253 - lr: 2.3730e-05
Epoch 46/50
39/39 [==============================] - 108s 3s/step - loss: 0.5342 - acc:
0.8090 - val_loss: 1.8823 - val_acc: 0.3870 - lr: 1.7798e-05
Epoch 47/50
39/39 [==============================] - 102s 3s/step - loss: 0.5313 - acc:
0.8086 - val_loss: 1.8564 - val_acc: 0.4521 - lr: 1.7798e-05
Epoch 48/50
39/39 [==============================] - 102s 3s/step - loss: 0.5177 - acc:
0.8222 - val_loss: 1.7960 - val_acc: 0.4521 - lr: 1.7798e-05
Epoch 49/50
39/39 [==============================] - 108s 3s/step - loss: 0.5178 - acc:
0.8096 - val_loss: 1.8814 - val_acc: 0.4138 - lr: 1.7798e-05
Epoch 50/50
39/39 [==============================] - 106s 3s/step - loss: 0.5097 - acc:
0.8171 - val_loss: 1.7588 - val_acc: 0.4674 - lr: 1.7798e-05
```

3. TESTING THE 2D CNN MODEL ON THE 2-DIMENSIONAL FEATURE MATRICES

[941]:

```
loss,accuracy=model2D.evaluate(mels_test,y_test,verbose=0)
print(f'Test Loss: {loss}')
print(f'Test Accuracy: {accuracy*100}%')
y_pred = model2D.predict(mels_test)
y_pred = np.argmax(y_pred, axis=1)
conf=confusion_matrix(y_test,y_pred)
cm=pd.DataFrame(
    conf,index=[i for i in set(labels)],
    columns=[i for i in set(labels)]
)
plt.figure(figsize=(12,7))
ax=sns.heatmap(cm,annot=True,fmt='d')
ax.set_title(f'confusion matrix for model ')
plt.show()
```

```
Test Loss: 1.5859235525131226
Test Accuracy: 50.73891878128052%
70/70 [==============================] - 12s 149ms/step
```



[943]:

```
from sklearn.metrics import f1_score,precision_score,recall_score
prec=precision_score(y_test,y_pred,average='weighted')
rec=recall_score(y_test,y_pred,average='weighted')
f1score=f1_score(y_test,y_pred,average='weighted')
print(f'Test precision: {prec}')
print(f'Test recall: {rec}')
print(f'Test f1-score: {f1score}')
```

```
Test precision: 0.5067107992243727
Test recall: 0.5073891625615764
Test f1-score: 0.5023103752175271
```

[944]:

```
fig=px.line(history2D.history,y=['acc','val_acc'],
           labels={'index':'epoch','value':'acc'},
           title=f'Epoch train accuracy and validation accuracy chart')
fig.show()
```

[945]:

```
fig=px.line(history2D.history,y=['loss','val_loss'],
           labels={'index':'epoch','value':'loss'},
           title=f'Epoch train loss and validation loss chart')
```

```
fig.show()
```