Speech Emotion Recognition

December 29, 2023

[1]:
```python
import pandas as pd
import numpy as np
import os
import sys
import librosa
import seaborn as sns

from matplotlib import pyplot as plt
from pathlib import Path as pth
from collections import Counter
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.model_selection import train_test_split
from IPython.display import Audio
import tensorflow as tf
import keras
from keras.callbacks import ReduceLROnPlateau
from keras.models import Sequential
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten, Dropout,␣
↪BatchNormalization, LSTM
from keras.utils import np_utils, to_categorical
from keras.callbacks import ModelCheckpoint

print("Cell Execution Completed.")
```

```
Cell Execution Completed.
```

[ ]:
```python
print(type(os.getcwd()))
print(os.listdir())
print(os.path.dirname("SAVEE"))
print(type(pth.cwd()))
print(list((pth.cwd()).iterdir()))
```

[ ]:
```python
currdir = os.getcwd()
ravdess = os.path.join(currdir, "RAVDESS\\audio_speech_actors_01-24")
crema = os.path.join(currdir, "CREMA-D\\AudioWAV")
tess = os.path.join(currdir, "TESS Toronto emotional speech set data\\TESS␣
↪Toronto emotional speech set data")
```

```
savee = os.path.join(currdir, "SAVEE\\ALL")
print(ravdess)
print(crema)
print(tess)
print(savee)
os.listdir(ravdess)
```

1. Ravdess Dataframe Here is the filename identifiers as per the official RAVDESS website:

Modality (01 = full-AV, 02 = video-only, 03 = audio-only). Vocal channel (01 = speech, 02 = song). Emotion (01 = neutral, 02 = calm, 03 = happy, 04 = sad, 05 = angry, 06 = fearful, 07 = disgust, 08 = surprised). Emotional intensity (01 = normal, 02 = strong). NOTE: There is no strong intensity for the 'neutral' emotion. Statement (01 = "Kids are talking by the door", 02 = "Dogs are sitting by the door"). Repetition (01 = 1st repetition, 02 = 2nd repetition). Actor (01 to 24. Odd numbered actors are male, even numbered actors are female). So, here's an example of an audio filename. 02-01-06-01-02-01-12.mp4 This means the meta data for the audio file is:

Video-only (02) Speech (01) Fearful (06) Normal intensity (01) Statement "dogs" (02) 1st Repetition (01) 12th Actor (12) - Female (as the actor ID number is even)

[ ]:
```
ravdess_directory_list = os.listdir(ravdess)
file_emotion = []
file_path = []
for actor_folder in ravdess_directory_list:
    audio_files = os.listdir(ravdess + "\\" + actor_folder)
    for file in audio_files:
        part = file.split('.')[0]
        part = part.split('-')
        file_emotion.append(int(part[2]))
        file_path.append(ravdess + "\\" + actor_folder + '\\' + file)

print("Number of Files in Ravdess:")
print(len(file_path))
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
ravdess_df = pd.concat([emotion_df, path_df], axis=1)

ravdess_df.Emotions.replace({1:'neutral', 2:'calm', 3:'happy', 4:'sad', 5:
 ↪'angry', 6:'fear', 7:'disgust', 8:'surprise'}, inplace=True)
print("Ravdess Database:")
ravdess_df.head()
```

4. CREMA-D dataset

Content CREMA-D is a data set of 7,442 original clips from 91 actors. These clips were from 48 male and 43 female actors between the ages of 20 and 74 coming from a variety of races and ethnicities (African America, Asian, Caucasian, Hispanic, and Unspecified). Actors spoke from a selection of 12 sentences. The sentences were presented using one of six different emotions (Anger, Disgust, Fear, Happy, Neutral, and Sad) and four different emotion levels (Low, Medium, High, and Unspecified).

The audio files in this dataset are named in such a way that the prefix letters describes the emotion classes as follows:

'a' = 'anger' 'd' = 'disgust' 'f' = 'fear' 'h' = 'happiness' 'n' = 'neutral' 'sa' = 'sadness' 'su' = 'surprise

Filename labeling conventions The Actor id is a 4 digit number at the start of the file. Each subsequent identifier is separated by an underscore (_).

Actors spoke from a selection of 12 sentences (in parentheses is the three letter acronym used in the second part of the filename):

It's eleven o'clock (IEO). That is exactly what happened (TIE). I'm on my way to the meeting (IOM). I wonder what this is about (IWW). The airplane is almost full (TAI). Maybe tomorrow it will be cold (MTI). I would like a new alarm clock (IWL) I think I have a doctor's appointment (ITH). Don't forget a jacket (DFA). I think I've seen this before (ITS). The surface is slick (TSI). We'll stop in a couple of minutes (WSI). The sentences were presented using different emotion (in parentheses is the three letter code used in the third part of the filename):

Anger (ANG) Disgust (DIS) Fear (FEA) Happy/Joy (HAP) Neutral (NEU) Sad (SAD) and emotion level (in parentheses is the two letter code used in the fourth part of the filename):

Low (LO) Medium (MD) High (HI) Unspecified (XX) The suffix of the filename is based on the type of file, flv for flash video used for presentation of both the video only, and the audio-visual clips. mp3 is used for the audio files used for the audio-only presentation of the clips. wav is used for files used for computational audio processing.

Audio Files MP3 Audio files used for presentation to the Raters are stored in the AudioMP3 directory.

Note: The following files do not have correct audio based on their filename, but they are most likely what the raters heard. (Thank you ruanxiu520 for finding these and pointing them out.):

1076_MTI_NEU_XX.mp3 - very short, no audio (WAV file is fine) 1076_MTI_SAD_XX.mp3/wav - no audio, very short 1064_TIE_SAD_XX.mp3 - file has no duration 1064_IEO_DIS_MD.mp3 - this file is actually 1 minute long and has all emotional displays for It's Eleven O'clock. Processed Audio WAV Audio files converted from the original video into a format appropriate for computational audio processing are stored in the AudioWAV directory.

[ ]:
```
crema_directory_list = os.listdir(crema)

file_emotion = []
file_path = []
print("Nunber of Audio Files:")
print(len(crema_directory_list))

for file in crema_directory_list:
    file_path.append(crema + "\\" + file)
    part=file.split('_')
    if part[2] == 'SAD':
        file_emotion.append('sad')
```

```
    elif part[2] == 'ANG':
        file_emotion.append('angry')
    elif part[2] == 'DIS':
        file_emotion.append('disgust')
    elif part[2] == 'FEA':
        file_emotion.append('fear')
    elif part[2] == 'HAP':
        file_emotion.append('happy')
    elif part[2] == 'NEU':
        file_emotion.append('neutral')
    else:
        file_emotion.append('Unknown')
emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
crema_df = pd.concat([emotion_df, path_df], axis=1)
crema_df.head()
```

Content There are a set of 200 target words were spoken in the carrier phrase "Say the word _' by two actresses (aged 26 and 64 years) and recordings were made of the set portraying each of seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral). There are 2800 data points (audio files) in total.

The dataset is organised such that each of the two female actor and their emotions are contain within its own folder. And within that, all 200 target words audio file can be found. The format of the audio file is a WAV format

[ ]:
```
tess_directory_list = os.listdir(tess)

file_emotion = []
file_path = []

for actor_folder in tess_directory_list:
    audio_files = os.listdir(tess + "\\" + actor_folder)
    for file in audio_files:
        part = file.split('.')[0]
        part = part.split('_')[2]
        if part=='ps':
            file_emotion.append('surprise')
        else:
            file_emotion.append(part)
        file_path.append(tess + "\\" + actor_folder + '\\' + file)

print("Number of Audio Files in Tess Dataset: ")
print(len(file_emotion))

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
path_df = pd.DataFrame(file_path, columns=['Path'])
tess_df = pd.concat([emotion_df, path_df], axis=1)
```

```
tess_df.head()
```

4. Surrey Audio-Visual Expressed Emotion (SAVEE) The audio files in this dataset are named in such a way that the prefix letters describes the emotion classes as follows:

'a' = 'anger' 'd' = 'disgust' 'f' = 'fear' 'h' = 'happiness' 'n' = 'neutral' 'sa' = 'sadness' 'su' = 'surprise'

The SAVEE database was recorded from four native English male speakers (identified as DC, JE, JK, KL), postgraduate students and researchers at the University of Surrey aged from 27 to 31 years. Emotion has been described psychologically in discrete categories: anger, disgust, fear, happiness, sadness and surprise. This is supported by the cross-cultural studies of Ekman [6] and studies of automatic emotion recognition tended to focus on recognizing these [12]. We added neutral to provide recordings of 7 emotion categories. The text material consisted of 15 TIMIT sentences per emotion: 3 common, 2 emotion-specific and 10 generic sentences that were different for each emotion and phonetically-balanced. The 3 common and 2 × 6 = 12 emotion-specific sentences were recorded as neutral to give 30 neutral sentences. This resulted in a total of 120 utterances per speaker, for example:

Common: She had your dark suit in greasy wash water all year. Anger: Who authorized the unlimited expense account? Disgust: Please take this dirty table cloth to the cleaners for me. Fear: Call an ambulance for medical assistance. Happiness: Those musicians harmonize marvelously. Sadness: The prospect of cutting back spending is an unpleasant one for any governor. Surprise: The carpet cleaners shampooed our oriental rug. Neutral: The best way to learn is to solve extra problems.

The distribution includes a complete list of sentences.

[ ]:
```
savee_directory_list = os.listdir(savee)

print("Number of files in Savee: ")
print(len(savee_directory_list))
file_emotion = []
file_path = []

for file in savee_directory_list:
    file_path.append(savee + "\\" + file)
    part = file.split('_')[1]
    emotion = part[:-6]
    if emotion=='a':
        file_emotion.append('angry')
    elif emotion=='d':
        file_emotion.append('disgust')
    elif emotion=='f':
        file_emotion.append('fear')
    elif emotion=='h':
        file_emotion.append('happy')
    elif emotion=='n':
        file_emotion.append('neutral')
```

```
    elif emotion=='sa':
        file_emotion.append('sad')
    else:
        file_emotion.append('surprise')

emotion_df = pd.DataFrame(file_emotion, columns=['Emotions'])
print(emotion_df)

path_df = pd.DataFrame(file_path, columns=['Path'])
print(path_df)
savee_df = pd.concat([emotion_df, path_df], axis=1)
savee_df.head()
```

[ ]:

```
# creating Dataframe using all the 4 dataframes we created so far.
emotion_dataset_path = pd.concat([ravdess_df, crema_df, tess_df, savee_df],␣
 ↪axis = 0)

emotion_dataset_path.to_csv("emotion_dataset_path.csv",index=False)
emotion_dataset_path.head()
```

[9]:

```
emotion_dataset_path.Emotions
```

[9]:

```
0        neutral
1        neutral
2        neutral
3        neutral
4           calm
          …
475     surprise
476     surprise
477     surprise
478     surprise
479     surprise
Name: Emotions, Length: 12162, dtype: object
```

[7]:

```
emotion_dataset_path = pd.read_csv("emotion_dataset_path.csv")
print(type(emotion_dataset_path.Emotions[1]))
emotion_dataset_path.Emotions
```

```
<class 'str'>
```

[7]:

```
0          neutral
1          neutral
2          neutral
3          neutral
4             calm
            …
12157     surprise
```

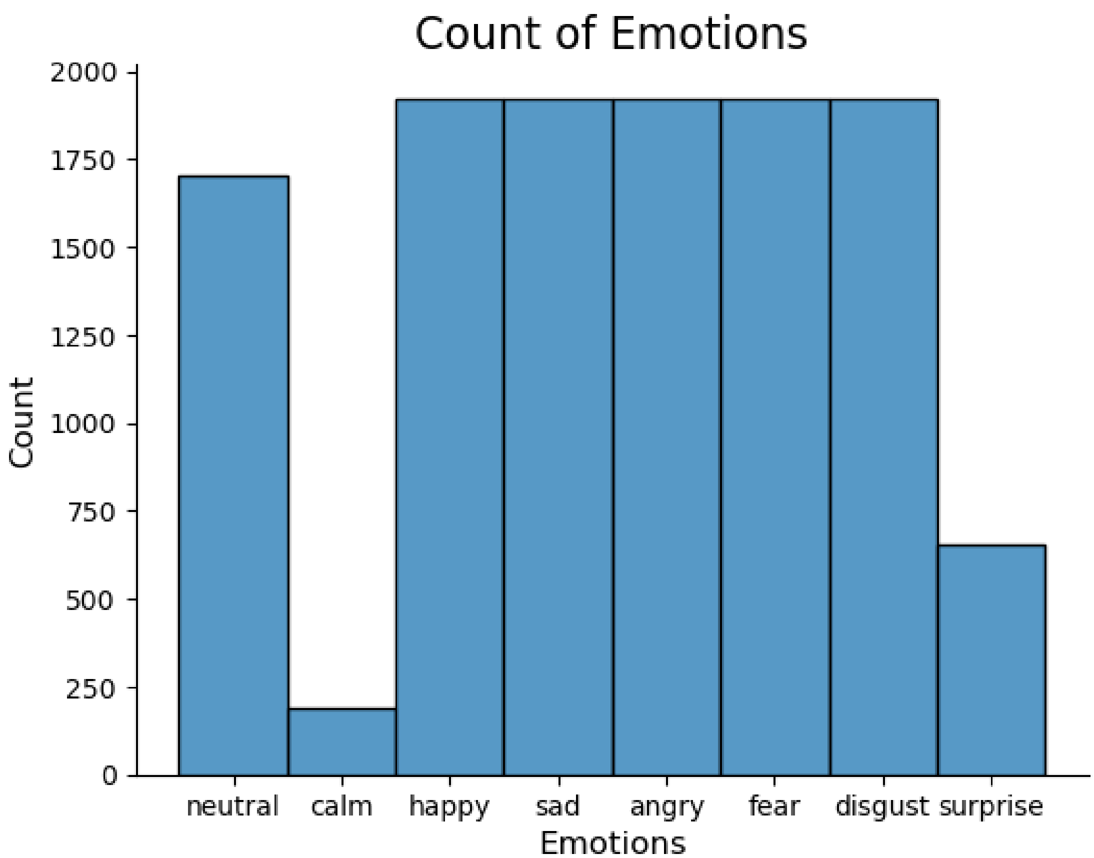
```
12158    surprise
12159    surprise
12160    surprise
12161    surprise
Name: Emotions, Length: 12162, dtype: object
```

[8]:
```
counter = Counter(emotion_dataset_path["Emotions"])
print(counter)op=True, right=True, left=False, bottom=False)
```

```
Counter({'happy': 1923, 'sad': 1923, 'angry': 1923, 'fear': 1923, 'disgust':
1923, 'neutral': 1703, 'surprise': 652, 'calm': 192})
```

[9]:
```
print(type(emotion_dataset_path.Emotions.value_counts()))
print(emotion_dataset_path.Emotions.value_counts())

plt.title('Count of Emotions', size=16)
sns.histplot(emotion_dataset_path.Emotions)
plt.ylabel('Count', size=12)
plt.xlabel('Emotions', size=12)
sns.despine(top=True, right=True, left=False, bottom=False)
plt.show()
```

```
<class 'pandas.core.series.Series'>
happy       1923
sad         1923
angry       1923
fear        1923
disgust     1923
neutral     1703
surprise     652
calm         192
Name: Emotions, dtype: int64
```



We can also plot waveplots and spectograms for audio signals

Waveplots - Waveplots let us know the loudness of the audio at a given time.

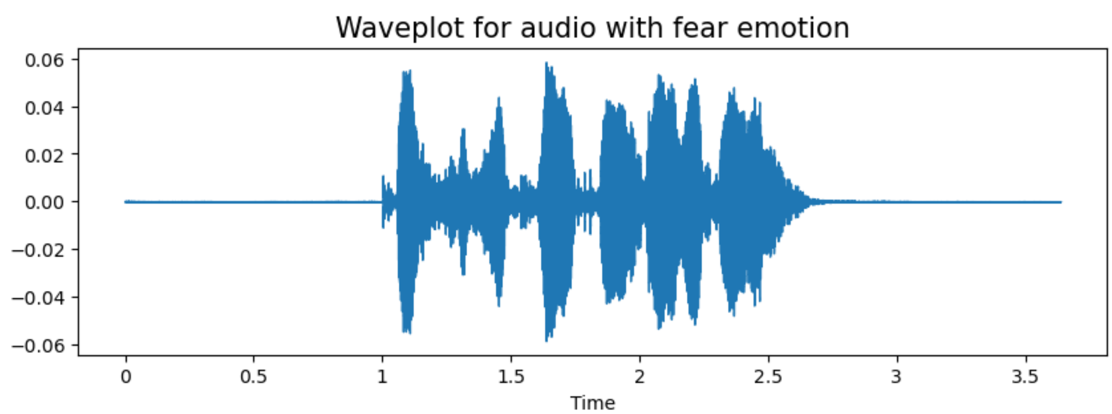
Spectograms - A spectrogram is a visual representation of the spectrum of frequencies of sound or other signals as they vary with time. It's a representation of frequencies changing with respect to time for given audio/music signals.

[10]:
```
def create_waveplot(data, sr, emotion_name):
    plt.figure(figsize=(10, 3))
    plt.title('Waveplot for audio with {} emotion'.format(emotion_name),␣
↪size=15)
    librosa.display.waveshow(data, sr=sr)
    plt.show()

def create_spectrogram(data, sr, emotion_name):
    stft_matrix = librosa.stft(data)
    print("In Function:")
    stft_matrix_db = librosa.amplitude_to_db(abs(stft_matrix))
    plt.figure(figsize=(12, 3))
```

```
    plt.title('Spectrogram for audio with {} emotion on linear scale'.
 ↪format(emotion_name), size=15)
    librosa.display.specshow(stft_matrix_db, sr=sr, x_axis='s', y_axis='hz')
    plt.colorbar()
    plt.figure(figsize=(12, 3))
    plt.title('Spectrogram for audio with {} emotion on a log scale'.
 ↪format(emotion_name), size=15)
    librosa.display.specshow(stft_matrix_db, sr=sr, x_axis='s', y_axis='log')
    plt.colorbar()
```

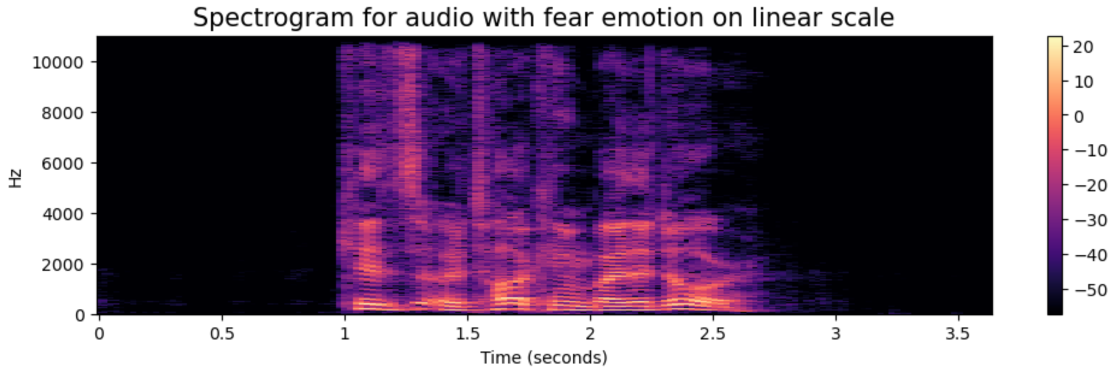
[11]:
```
emotion='fear'
path = np.array(emotion_dataset_path.Path[emotion_dataset_path.
 ↪Emotions==emotion])[1]
#print(path)
data, sampling_rate = librosa.load(path)
print(data)
print(data.shape)
print(sampling_rate)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```

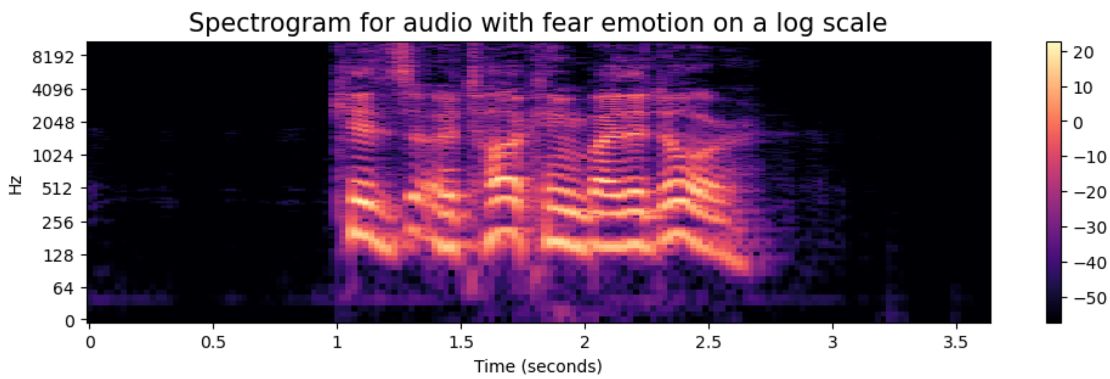
```
[ 9.3495757e-05  1.6403565e-04  1.1930163e-04 …  1.6012858e-11
 -1.1257498e-11  0.0000000e+00]
(80196,)
22050
```



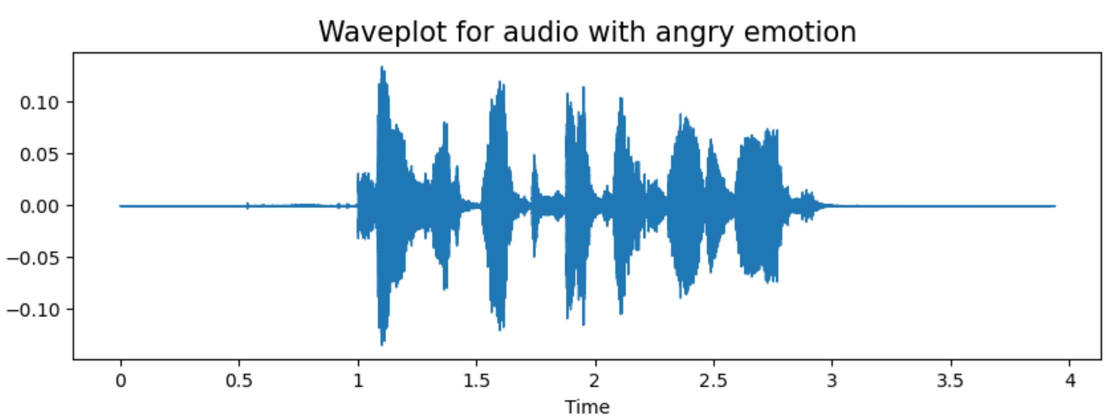

```
In Function:
```

[11]: `<IPython.lib.display.Audio object>`



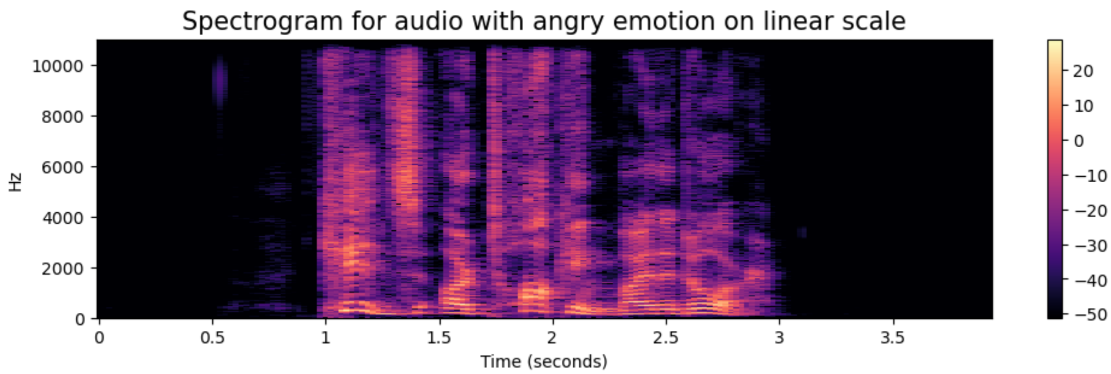


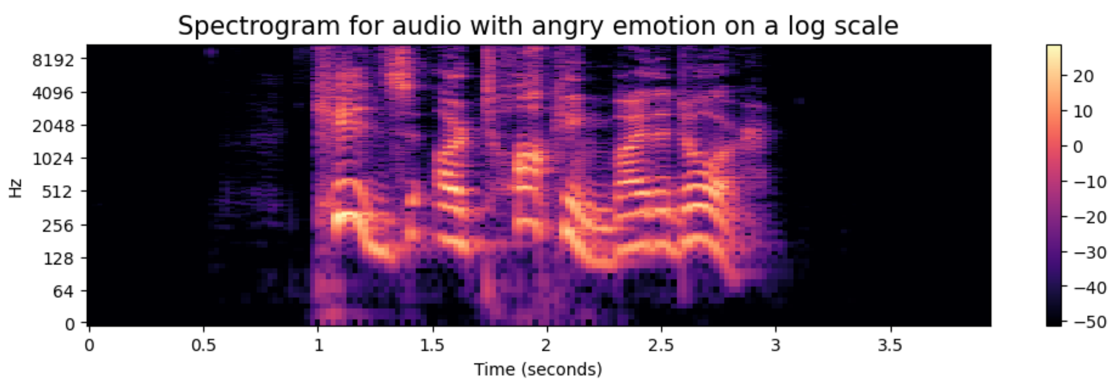
[12]:
```
emotion='angry'
path = np.array(emotion_dataset_path.Path[emotion_dataset_path.
  ↪Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```



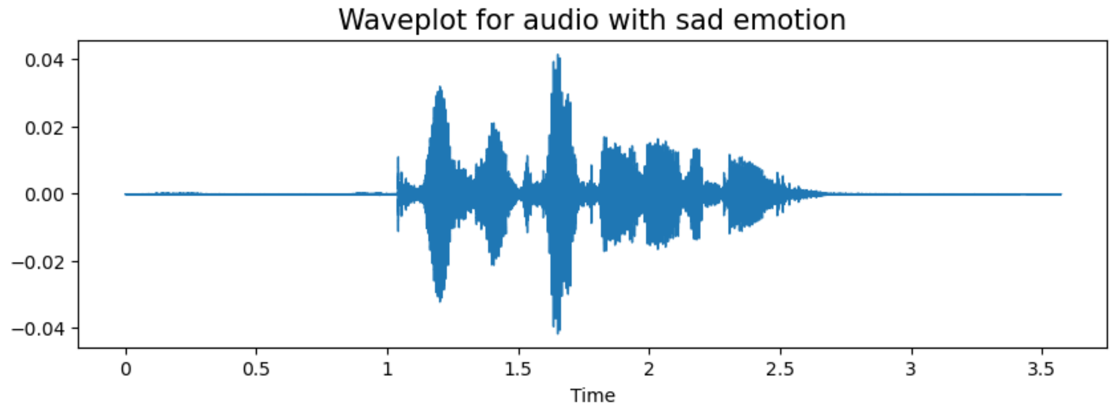
In Function:

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



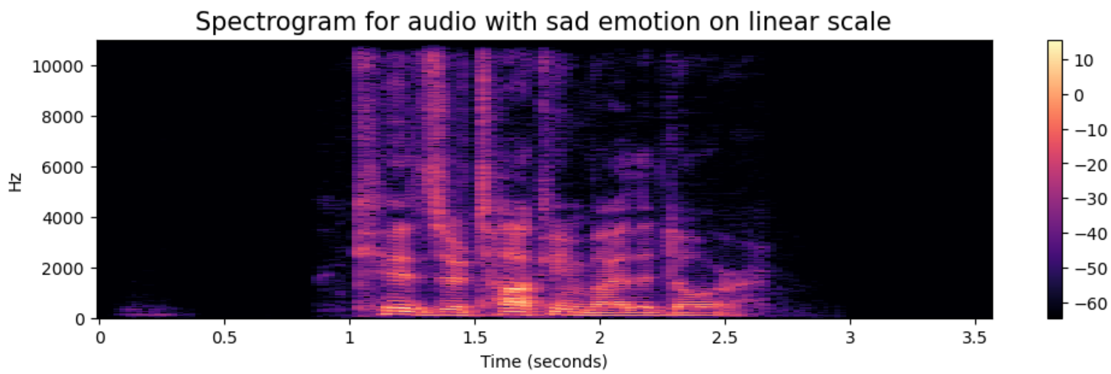


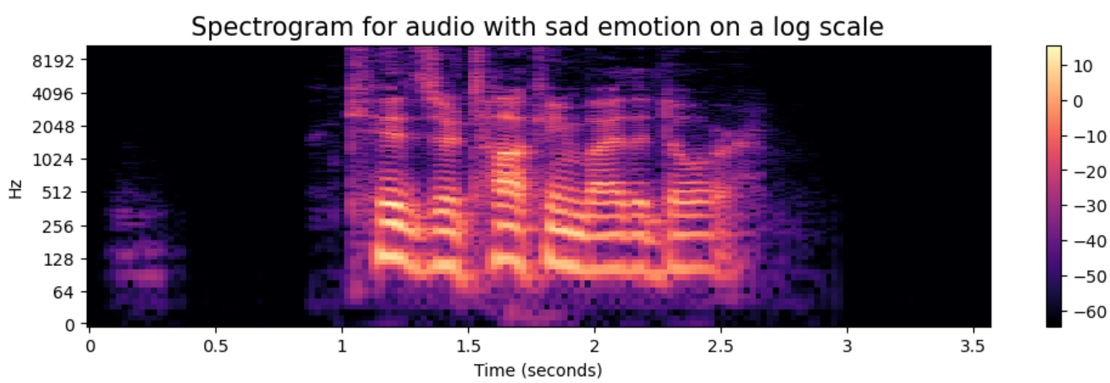
[13]:
```
emotion='sad'
path = np.array(emotion_dataset_path.Path[emotion_dataset_path.
 ↪Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```



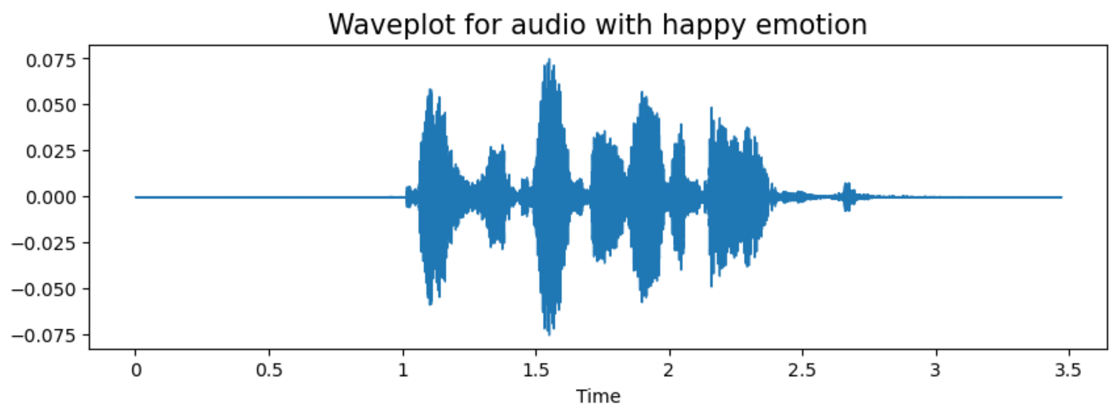
In Function:

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



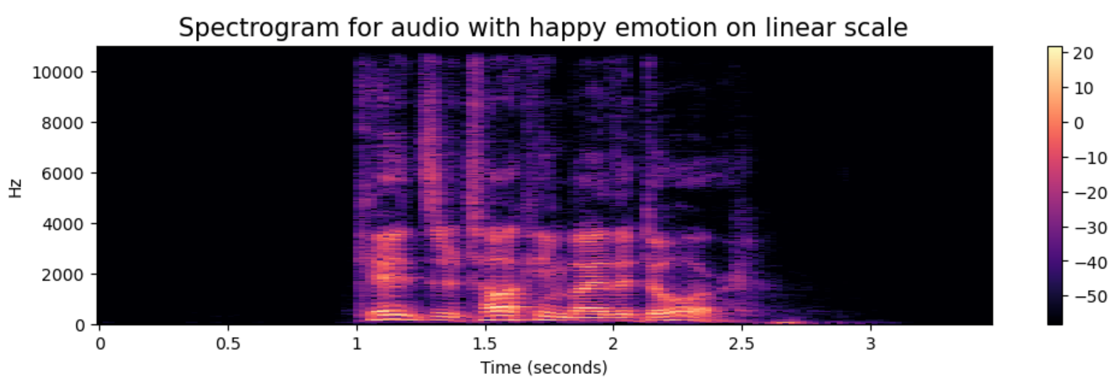


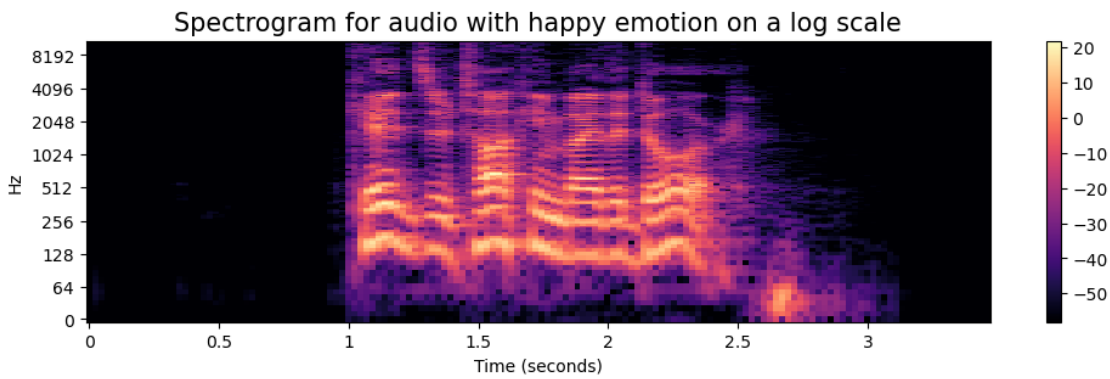
```
[14]: emotion='happy'
path = np.array(emotion_dataset_path.Path[emotion_dataset_path.
  ↪Emotions==emotion])[1]
data, sampling_rate = librosa.load(path)
create_waveplot(data, sampling_rate, emotion)
create_spectrogram(data, sampling_rate, emotion)
Audio(path)
```



In Function:

```
[14]: <IPython.lib.display.Audio object>
```





Data Augmentation Data augmentation is the process by which we create new synthetic data samples by adding small perturbations on our initial training set. To generate syntactic data for audio, we can apply noise injection, shifting time, changing pitch and speed. The objective is to make our model invariant to those perturbations and enhace its ability to generalize. In order for this to work, adding the perturbations must conserve the same label as the original training sample. In images data augmentation can be performed by shifting the image, zooming, rotating … First, let's check which augmentation techniques works better for our dataset.

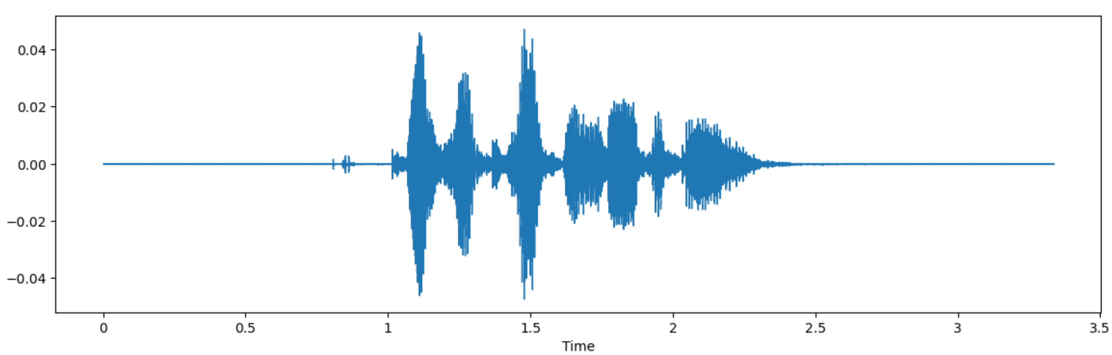
[ ]:
```
def noise(data):
    noise_amp = 0.035*np.random.uniform()*np.amax(data)
    data = data + noise_amp*np.random.normal(size=data.shape[0])
    return data

def stretch(data, rate=0.8):
    return librosa.effects.time_stretch(data, rate = rate)

def shift(data):
    shift_range = int(np.random.uniform(low=-5, high = 5)*1000)
    x = np.roll(data, shift_range)
    return x

def pitch(data, sampling_rate, pitch_factor=0.7):
    x = librosa.effects.pitch_shift(data, sr = sampling_rate, n_steps =
 ↪pitch_factor)
    return x

path = np.array(emotion_dataset_path.Path)[1]
print(path)
data, sample_rate = librosa.load(path)
print(data)
print(type(data))
print(data.shape)
print(sampling_rate)
```

[16]:
```python
from numpy.random import default_rng
rng = default_rng()
vals = rng.uniform()
more_vals = rng.uniform()
print("*"*50)
print("Vals: {}".format(vals))
print("More_Vals: {}".format(more_vals))
print("*"*50)
# instead of this (legacy version)
from numpy import random
vals = random.uniform()
more_vals = random.uniform()
print("*"*50)
print("Vals: {}".format(vals))
print("More_Vals: {}".format(more_vals))
print("*"*50)
```

```
**************************************************
Vals: 0.23159435402914585
More_Vals: 0.8713390150812016
**************************************************
**************************************************
Vals: 0.2555688501693628
More_Vals: 0.4307683890112539
**************************************************
```

[17]:
```python
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=data, sr=sample_rate)
Audio(path)
```

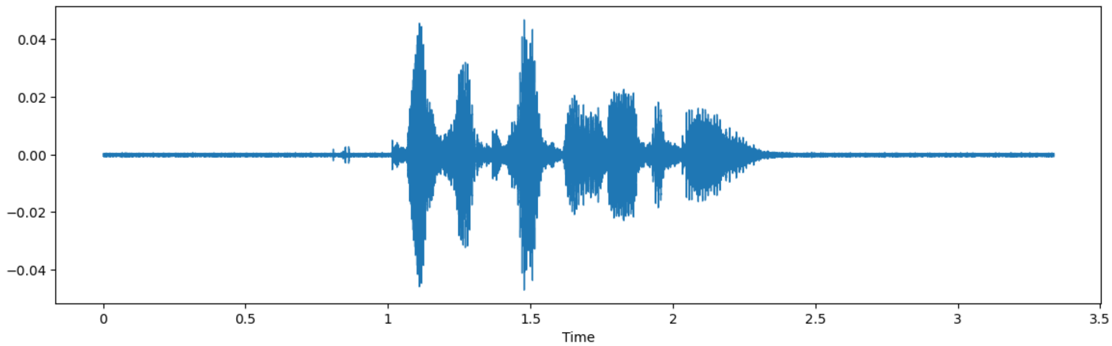
[17]:
```
<IPython.lib.display.Audio object>
```



[18]:
```python
# Adding noise
x = noise(data)
```

```
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

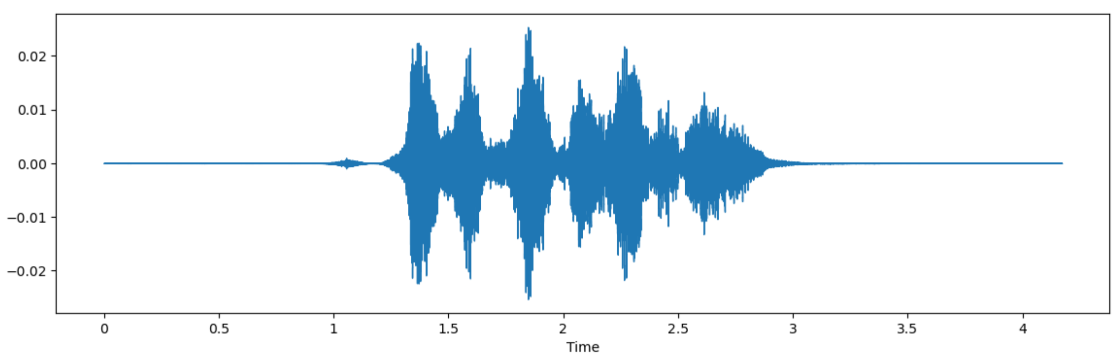
[18]: <IPython.lib.display.Audio object>



[19]:
```
# Stretching
x = stretch(data)
print("Stretched Data: ")
print(x)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

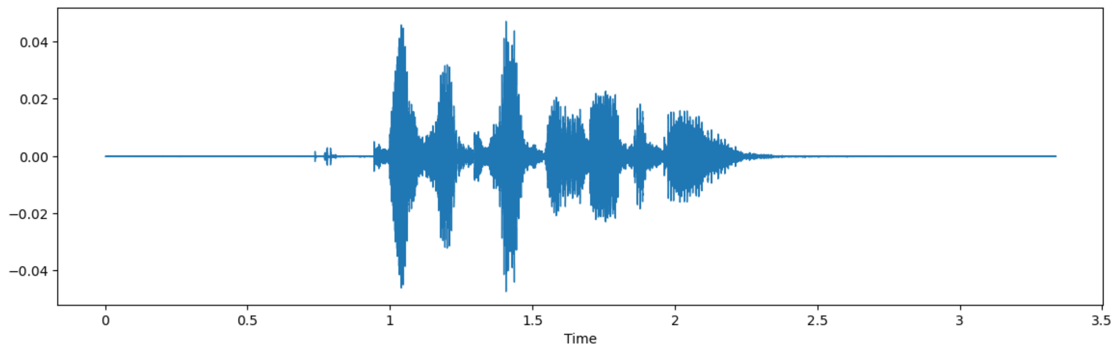
```
Stretched Data:
[-2.7293686e-06 -1.2543627e-05 -1.0094540e-06 …  1.5635393e-07
 -4.5327849e-07 -1.4834899e-06]
```

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



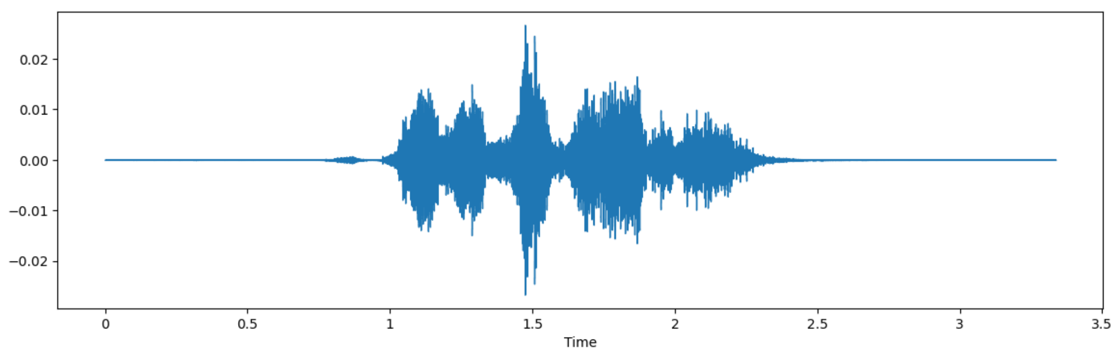
[20]:
```
# Shifting
x = shift(data)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

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



[21]:
```
# Pitch
x = pitch(data, sample_rate)
plt.figure(figsize=(14,4))
librosa.display.waveshow(y=x, sr=sample_rate)
Audio(x, rate=sample_rate)
```

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



1 Feature Extraction

Zero Crossing Rate : The rate of sign-changes of the signal during the duration of a particular frame. Energy : The sum of squares of the signal values, normalized by the respective frame length. Entropy of Energy : The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes. Spectral Centroid : The center of gravity of the spectrum. Spectral Spread : The second central moment of the spectrum. Spectral Entropy : Entropy of the normalized spectral energies for a set of sub-frames. Spectral Flux : The squared difference between the normalized magnitudes of the spectra of the two successive frames. Spectral Rolloff : The frequency below which 90% of the magnitude distribution of the spectrum is concentrated. MFCCs Mel Frequency Cepstral Coefficients form a cepstral representation where the frequency bands are not linear but distributed according to the mel-scale. Chroma Vector : A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing). Chroma Deviation : The standard deviation of the 12 chroma coefficients. In this project i am not going deep in feature selection process to check which features are good for our dataset rather i am only extracting 5 features:

Zero Crossing Rate Chroma_stft MFCC RMS(root mean square) value MelSpectogram to train our model.

[56]:
```
def extract_features(data, type_data):

    ############################################################################
    # ZCR
    result = np.array([])
    x = librosa.feature.zero_crossing_rate(y=data)
    zcr = np.mean(x.T, axis=0)
    result = np.hstack((result, zcr))
    ############################################################################

    ############################################################################
    # Chroma_stft
    stft = np.abs(librosa.stft(data))
    x = librosa.feature.chroma_stft(S=stft, sr=sample_rate)
    x_T = librosa.feature.chroma_stft(S=stft, sr=sample_rate).T
    chroma_stft = np.mean(x_T, axis=0)
    result = np.hstack((result, chroma_stft))
    ############################################################################

    ############################################################################
    # MFCC
    x = librosa.feature.mfcc(y=data, sr=sample_rate)
    mfcc = np.mean(x.T, axis=0)
    result = np.hstack((result, mfcc))
    ############################################################################

    ############################################################################
    # Root Mean Square Value
    rms = np.mean(librosa.feature.rms(y=data).T, axis=0)
    result = np.hstack((result, rms))
    ############################################################################

    ############################################################################
```

```
    # Spectral Centroid
    sp_cent = np.mean(librosa.feature.spectral_centroid(y=data, sr=sample_rate).
 ↪T, axis=0)
    result = np.hstack((result, sp_cent)) # stacking horizontally
    ##########################################################################

    ##########################################################################
    # Spectral Bandwidth
    sp_band = np.mean(librosa.feature.spectral_bandwidth(y=data,␣
 ↪sr=sample_rate, p=2).T, axis = 0)
    ##########################################################################

    ##########################################################################
    # Tonnetz
    tonnetz = np.mean(librosa.feature.tonnetz(y=data, sr=sample_rate).T, axis=0)
    result = np.hstack((result, tonnetz))
    ##########################################################################

    ##########################################################################
    # MelSpectogram
    mel = np.mean(librosa.feature.melspectrogram(y=data, sr=sample_rate).T,␣
 ↪axis=0)
    result = np.hstack((result, mel))
    ##########################################################################

    return result

def get_features(path):
    result = np.array([])
    # duration and offset are used to take care of the no audio in start and␣
 ↪the ending of each audio files as seen above.
    data, sample_rate = librosa.load(path, duration=2.5, offset=0.6)

    # without augmentation
    res1 = extract_features(data, type_data = "Normal")
    result = np.array(res1)

    # data with noise
    noise_data = noise(data)
    res2 = extract_features(noise_data, type_data = "Noisy")
    result = np.vstack((result, res2)) # stacking vertically

    # data with stretching and pitching
    new_data = stretch(data)
    data_stretch_pitch = pitch(new_data, sample_rate)
    res3 = extract_features(data_stretch_pitch, type_data = "Stretched and␣
 ↪Pitched")
```

```
    result = np.vstack((result, res3)) # stacking vertically

    return result
```

[55]:
```
result = np.array([])

print("Spectral Bandwidth: ")
print(librosa.feature.spectral_bandwidth(y=data, sr=sample_rate, p=2).shape)
print(librosa.feature.spectral_bandwidth(y=data, sr=sample_rate, p=2))
print("Spectral Bandwidth Mean: ")
print(np.mean(librosa.feature.spectral_bandwidth(y=data, sr=sample_rate, p=2).
 ↪T, axis = 0).shape)
print(np.mean(librosa.feature.spectral_bandwidth(y=data, sr=sample_rate, p=2).
 ↪T, axis = 0))
#print(data.shape)
```

```
Spectral Bandwidth:
(1, 144)
[[3303.78085208 3274.44971318 3177.76947568 3206.96647617 3353.36471121
  3308.70090269 3299.87924658 3444.81772226 3412.57586825 3358.78186151
  3394.37813299 3358.59529578 3348.75443118 3437.85529144 3490.23148979
  3419.0946381  3258.20743153 3179.15767842 3248.81130867 3314.81704415
  3359.07287253 3350.01420581 3378.68052222 3485.33147722 3400.47164758
  3220.60676581 3127.27020542 3303.99256486 3334.85636797 3311.67210471
  3297.59320277 3215.29211276 3190.33786456 3109.97038787 2310.31736291
  2285.60940231 2218.11869621 2344.52755378 2612.12668631 2951.64064257
  3051.90453605 3194.66155122 2916.65418367 2721.62941783 2799.53094018
  2968.09364205 2203.58304824 2013.44135361 2116.75150486 2107.15475593
  2504.76866421 3518.11254154 3017.55097698 2889.72580571 2349.80524747
  1939.4673643  2261.66802956 2476.15724486 3059.99272727 2877.31693646
  2798.5984489  2582.85899456 1896.26734717 1560.96923044 1553.9588608
  1652.64196101 1648.40232264 2116.65919529 2954.20107761 2722.56502814
  2210.259698   2129.9283097  1901.62579841 1554.23464617 1408.58566823
  1276.13566486 1368.61331614 1359.97906448 1420.28809685 1621.9124835
  1789.35442016 1802.78889834 2273.55131536 1976.90745659 1723.34807921
  1612.89999618 1607.6111545  2167.98276432 1944.48322262 1478.58287735
  1243.32109163 1236.30702769 1257.43801156 1287.65053956 1312.66654487
  1283.39926732 1328.3044336  1430.17295389 1542.61150025 1638.60686369
  1905.80158354 2030.08749157 2090.90239223 2120.81889163 2306.64871604
  2784.62880364 3052.79349013 3184.45387714 3172.80891871 3154.30136148
  3184.29082108 3162.20952373 3171.7825674  3148.94431668 3141.25669395
  3181.5418026  3258.66808049 3270.22479869 3313.16778207 3379.27548729
  3330.89509969 3354.48169383 3308.60155998 3301.97743802 3286.2842948
  3227.22031037 3100.19500831 3039.60183776 3112.55567247 3201.77163909
  3125.61241001 3140.56319354 3110.48370368 3234.64029393 3237.4863374
  3268.72782738 3284.43800121 3241.52958609 3324.36964926 3339.41271066
  3226.62071719 3125.08736608 3151.39492471 3168.7063067 ]]
Spectral Bandwidth Mean:
```

```
(1,)
[2675.83612016]
```

[ ]:
```
X, y = [], []
for path, emotion in zip(emotion_dataset_path.Path, emotion_dataset_path.
↪Emotions):
    feature = get_features(path)
    for normal_plus_additional_features in feature:
        X.append(normal_plus_additional_features)
        # appending emotion 3 times as we have made 3 augmentation techniques␣
↪on each audio file.
        y.append(emotion)
```

[58]:
```
print("Input Variable: ")
print(len(X))
#print(X)
print("Output Variable: ")
print(len(y))
#print(y)
print(emotion_dataset_path.Path.shape)
```

```
Input Variable:
36486
Output Variable:
36486
(12162,)
```

[59]:
```
Features_db = pd.DataFrame(X)
Features_db['labels'] = y
Features_db.to_csv('Features_Database_Extended.csv', index=False)
Features_db.head()
```

[59]:

	0	1	2	3	4	5	6	\
0	0.321275	0.729664	0.750032	0.730624	0.735275	0.713529	0.660531	
1	0.326891	0.804860	0.832834	0.817533	0.827433	0.817272	0.720943	
2	0.188256	0.622128	0.699219	0.753332	0.721223	0.701737	0.682353	
3	0.293566	0.673896	0.722096	0.723508	0.682302	0.680533	0.675352	
4	0.309109	0.768068	0.812219	0.806472	0.774351	0.782551	0.753839	

	7	8	9	…	161	162	\
0	0.684966	0.733049	0.753971	…	4.310903e-06	3.291511e-06	
1	0.691741	0.734845	0.762425	…	8.158390e-05	8.470013e-05	
2	0.662828	0.686490	0.733966	…	8.576332e-07	9.576413e-07	
3	0.628977	0.679179	0.707283	…	6.984504e-06	7.034949e-06	
4	0.649087	0.676392	0.728209	…	5.875406e-05	6.157950e-05	

	163	164	165	166	167	168	\
0	2.148075e-06	2.279739e-06	5.116493e-06	8.190282e-06	0.000007	0.000005	

```
1  8.842301e-05  8.285536e-05  8.676062e-05  9.658957e-05  0.000086  0.000081
2  7.733594e-07  5.233102e-07  3.592796e-07  9.261687e-07  0.000002  0.000001
3  6.654923e-06  6.979548e-06  1.214236e-05  9.640183e-06  0.000011  0.000006
4  6.059422e-05  5.843470e-05  6.257253e-05  5.941999e-05  0.000065  0.000058

            169   labels
0  4.245834e-07  neutral
1  8.449430e-05  neutral
2  7.753984e-08  neutral
3  4.254087e-07  neutral
4  5.034854e-05  neutral

[5 rows x 171 columns]
```

[2]:
```python
Features_db = pd.read_csv('Features_Database_Extended.csv')
Features_db.head()
```

[2]:
```
          0         1         2         3         4         5         6  \
0  0.321275  0.729664  0.750032  0.730624  0.735275  0.713529  0.660531
1  0.326891  0.804860  0.832834  0.817533  0.827433  0.817272  0.720943
2  0.188256  0.622128  0.699219  0.753332  0.721223  0.701737  0.682353
3  0.293566  0.673896  0.722096  0.723508  0.682302  0.680533  0.675352
4  0.309109  0.768068  0.812219  0.806472  0.774351  0.782551  0.753839

          7         8         9  …           161           162  \
0  0.684966  0.733049  0.753971  …  4.310903e-06  3.291511e-06
1  0.691741  0.734845  0.762425  …  8.158390e-05  8.470013e-05
2  0.662828  0.686490  0.733966  …  8.576332e-07  9.576413e-07
3  0.628977  0.679179  0.707283  …  6.984504e-06  7.034949e-06
4  0.649087  0.676392  0.728209  …  5.875406e-05  6.157950e-05

            163           164           165           166       167       168  \
0  2.148075e-06  2.279739e-06  5.116493e-06  8.190282e-06  0.000007  0.000005
1  8.842301e-05  8.285536e-05  8.676062e-05  9.658957e-05  0.000086  0.000081
2  7.733594e-07  5.233102e-07  3.592796e-07  9.261687e-07  0.000002  0.000001
3  6.654923e-06  6.979548e-06  1.214236e-05  9.640183e-06  0.000011  0.000006
4  6.059422e-05  5.843470e-05  6.257253e-05  5.941999e-05  0.000065  0.000058

            169   labels
0  4.245834e-07  neutral
1  8.449430e-05  neutral
2  7.753984e-08  neutral
3  4.254087e-07  neutral
4  5.034854e-05  neutral

[5 rows x 171 columns]
```

[3]:
```python
X = Features_db.iloc[: ,:-1].values
y = Features_db['labels'].values
```

[4]:
```python
# As this is a multiclass classification problem onehotencoding our Y.
encoder = OneHotEncoder()
y = encoder.fit_transform(np.array(y).reshape(-1,1)).toarray()
```

[75]:
```python
print(asd.shape)
asd = asd.reshape(-1,1)
print(asd.shape)
print(encoder)
zxc = encoder.fit_transform(asd)
print(zxc)
```

```
OneHotEncoder()
  (0, 5)        1.0
  (1, 5)        1.0
  (2, 5)        1.0
  (3, 5)        1.0
  (4, 5)        1.0
  (5, 5)        1.0
  (6, 5)        1.0
  (7, 5)        1.0
  (8, 5)        1.0
  (9, 5)        1.0
  (10, 5)       1.0
  (11, 5)       1.0
  (12, 1)       1.0
  (13, 1)       1.0
  (14, 1)       1.0
  (15, 1)       1.0
  (16, 1)       1.0
  (17, 1)       1.0
  (18, 1)       1.0
  (19, 1)       1.0
  (20, 1)       1.0
  (21, 1)       1.0
  (22, 1)       1.0
  (23, 1)       1.0
  (24, 1)       1.0
  :     :
  (36461, 7)    1.0
  (36462, 7)    1.0
  (36463, 7)    1.0
  (36464, 7)    1.0
  (36465, 7)    1.0
  (36466, 7)    1.0
  (36467, 7)    1.0
```

```
  (36468, 7)    1.0
  (36469, 7)    1.0
  (36470, 7)    1.0
  (36471, 7)    1.0
  (36472, 7)    1.0
  (36473, 7)    1.0
  (36474, 7)    1.0
  (36475, 7)    1.0
  (36476, 7)    1.0
  (36477, 7)    1.0
  (36478, 7)    1.0
  (36479, 7)    1.0
  (36480, 7)    1.0
  (36481, 7)    1.0
  (36482, 7)    1.0
  (36483, 7)    1.0
  (36484, 7)    1.0
  (36485, 7)    1.0
```

[5]:
```python
# splitting data
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0,␣
 ↪shuffle=True)
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

[5]: ((27364, 170), (27364, 8), (9122, 170), (9122, 8))

[6]:
```python
# scaling our data with sklearn's Standard scaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
X_train.shape, y_train.shape, X_test.shape, y_test.shape
```

[6]: ((27364, 170), (27364, 8), (9122, 170), (9122, 8))

[7]:
```python
# making our data compatible to model.
X_train = np.expand_dims(X_train, axis=2)
X_test = np.expand_dims(X_test, axis=2)
print(X_train.shape, y_train.shape, X_test.shape, y_test.shape)
print(X_train)
```

```
(27364, 170, 1) (27364, 8) (9122, 170, 1) (9122, 8)
[[[ 0.82521273]
  [ 0.57463458]
  [ 0.3981078 ]
  …
  [-0.17440619]
  [-0.1797417 ]
  [-0.20402521]]
```

```
[[ 1.10739479]
 [ 0.90333688]
 [ 1.14994919]
 …
 [-0.20375293]
 [-0.20674512]
 [-0.20534766]]

[[ 1.39898172]
 [-0.1518979 ]
 [-0.35460417]
 …
 [ 0.04822136]
 [ 0.02812277]
 [-0.18475306]]

…

[[ 0.20721122]
 [-0.62076758]
 [-0.7327683 ]
 …
 [-0.21412062]
 [-0.20843802]
 [-0.1974524 ]]

[[-0.83929816]
 [ 0.44684531]
 [-0.08212176]
 …
 [-0.23187261]
 [-0.22332798]
 [-0.20737452]]

[[ 0.78003991]
 [-0.24550737]
 [-0.43700153]
 …
 [-0.22652189]
 [-0.21848669]
 [-0.20659249]]]
```

[84]:
```
X_train[0][0]
```

[84]:
```
array([0.82944584])
```

[8]:

```
model=Sequential()
model.add(Conv1D(256, kernel_size=5, strides=1, padding='same',␣
 ↪activation='relu', input_shape=(X_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(256, kernel_size=5, strides=1, padding='same',␣
 ↪activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same',␣
 ↪activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))
model.add(Dropout(0.2))

model.add(Conv1D(64, kernel_size=5, strides=1, padding='same',␣
 ↪activation='relu'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Flatten())
model.add(Dense(units=32, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=8, activation='softmax'))
#model.add(Dense(units=8, activation='relu'))
model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy' , metrics␣
 ↪= ['accuracy'])
#model.compile(optimizer = 'adam' , loss = tf.nn.ctc_loss(labels=y_train,logits␣
 ↪= X_train) , metrics = ['accuracy'])
model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 170, 256)          1536

 max_pooling1d (MaxPooling1D  (None, 85, 256)          0
 )

 conv1d_1 (Conv1D)           (None, 85, 256)           327936

 max_pooling1d_1 (MaxPooling  (None, 43, 256)          0
 1D)

 conv1d_2 (Conv1D)           (None, 43, 128)           163968

 max_pooling1d_2 (MaxPooling  (None, 22, 128)          0
```

```
 1D)

 dropout (Dropout)          (None, 22, 128)          0

 conv1d_3 (Conv1D)          (None, 22, 64)           41024

 max_pooling1d_3 (MaxPooling  (None, 11, 64)         0
 1D)

 flatten (Flatten)          (None, 704)              0

 dense (Dense)              (None, 32)               22560

 dropout_1 (Dropout)        (None, 32)               0

 dense_1 (Dense)            (None, 8)                264

=================================================================
Total params: 557,288
Trainable params: 557,288
Non-trainable params: 0
_________________________________________________________________
```

[9]:
```
rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=1, patience=2,␣
 ↪min_lr=0.0000000001)
history=model.fit(X_train, y_train, batch_size=64, epochs=150,␣
 ↪validation_data=(X_test, y_test), callbacks=[rlrp])
```

```
Epoch 1/150
428/428 [==============================] - 48s 110ms/step - loss: 1.7182 -
accuracy: 0.2933 - val_loss: 1.4962 - val_accuracy: 0.3870 - lr: 0.0010
Epoch 2/150
428/428 [==============================] - 51s 119ms/step - loss: 1.5045 -
accuracy: 0.3871 - val_loss: 1.3960 - val_accuracy: 0.4315 - lr: 0.0010
Epoch 3/150
428/428 [==============================] - 56s 131ms/step - loss: 1.4160 -
accuracy: 0.4270 - val_loss: 1.3122 - val_accuracy: 0.4690 - lr: 0.0010
Epoch 4/150
428/428 [==============================] - 49s 114ms/step - loss: 1.3583 -
accuracy: 0.4525 - val_loss: 1.2754 - val_accuracy: 0.4881 - lr: 0.0010
Epoch 5/150
428/428 [==============================] - 48s 113ms/step - loss: 1.3136 -
accuracy: 0.4707 - val_loss: 1.2363 - val_accuracy: 0.4941 - lr: 0.0010
Epoch 6/150
428/428 [==============================] - 49s 114ms/step - loss: 1.2745 -
accuracy: 0.4872 - val_loss: 1.2236 - val_accuracy: 0.5073 - lr: 0.0010
Epoch 7/150
428/428 [==============================] - 47s 109ms/step - loss: 1.2439 -
```

```
accuracy: 0.5016 - val_loss: 1.1864 - val_accuracy: 0.5205 - lr: 0.0010
Epoch 8/150
428/428 [==============================] - 47s 109ms/step - loss: 1.2248 -
accuracy: 0.5049 - val_loss: 1.1718 - val_accuracy: 0.5331 - lr: 0.0010
Epoch 9/150
428/428 [==============================] - 47s 109ms/step - loss: 1.2045 -
accuracy: 0.5152 - val_loss: 1.1551 - val_accuracy: 0.5361 - lr: 0.0010
Epoch 10/150
428/428 [==============================] - 47s 109ms/step - loss: 1.1814 -
accuracy: 0.5248 - val_loss: 1.1600 - val_accuracy: 0.5275 - lr: 0.0010
Epoch 11/150
428/428 [==============================] - 47s 109ms/step - loss: 1.1709 -
accuracy: 0.5289 - val_loss: 1.1599 - val_accuracy: 0.5266 - lr: 0.0010
Epoch 12/150
428/428 [==============================] - 47s 110ms/step - loss: 1.1541 -
accuracy: 0.5359 - val_loss: 1.1485 - val_accuracy: 0.5453 - lr: 0.0010
Epoch 13/150
428/428 [==============================] - 46s 107ms/step - loss: 1.1427 -
accuracy: 0.5437 - val_loss: 1.1201 - val_accuracy: 0.5491 - lr: 0.0010
Epoch 14/150
428/428 [==============================] - 46s 108ms/step - loss: 1.1255 -
accuracy: 0.5481 - val_loss: 1.1028 - val_accuracy: 0.5605 - lr: 0.0010
Epoch 15/150
428/428 [==============================] - 46s 107ms/step - loss: 1.1128 -
accuracy: 0.5537 - val_loss: 1.1300 - val_accuracy: 0.5536 - lr: 0.0010
Epoch 16/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0987 -
accuracy: 0.5597 - val_loss: 1.1379 - val_accuracy: 0.5583 - lr: 0.0010
Epoch 17/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0945 -
accuracy: 0.5626 - val_loss: 1.1003 - val_accuracy: 0.5654 - lr: 0.0010
Epoch 18/150
428/428 [==============================] - 45s 106ms/step - loss: 1.0809 -
accuracy: 0.5667 - val_loss: 1.1060 - val_accuracy: 0.5638 - lr: 0.0010
Epoch 19/150
428/428 [==============================] - 46s 108ms/step - loss: 1.0701 -
accuracy: 0.5740 - val_loss: 1.0881 - val_accuracy: 0.5633 - lr: 0.0010
Epoch 20/150
428/428 [==============================] - 45s 106ms/step - loss: 1.0579 -
accuracy: 0.5752 - val_loss: 1.0834 - val_accuracy: 0.5709 - lr: 0.0010
Epoch 21/150
428/428 [==============================] - 46s 106ms/step - loss: 1.0490 -
accuracy: 0.5823 - val_loss: 1.0965 - val_accuracy: 0.5721 - lr: 0.0010
Epoch 22/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0314 -
accuracy: 0.5906 - val_loss: 1.0941 - val_accuracy: 0.5732 - lr: 0.0010
Epoch 23/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0211 -
```

```
accuracy: 0.5906 - val_loss: 1.0768 - val_accuracy: 0.5741 - lr: 0.0010
Epoch 24/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0191 -
accuracy: 0.5945 - val_loss: 1.0842 - val_accuracy: 0.5784 - lr: 0.0010
Epoch 25/150
428/428 [==============================] - 45s 105ms/step - loss: 1.0223 -
accuracy: 0.5925 - val_loss: 1.0941 - val_accuracy: 0.5694 - lr: 0.0010
Epoch 26/150
428/428 [==============================] - 45s 106ms/step - loss: 1.0039 -
accuracy: 0.6027 - val_loss: 1.0900 - val_accuracy: 0.5734 - lr: 0.0010
Epoch 27/150
428/428 [==============================] - 45s 106ms/step - loss: 0.9835 -
accuracy: 0.6096 - val_loss: 1.0663 - val_accuracy: 0.5859 - lr: 0.0010
Epoch 28/150
428/428 [==============================] - 45s 106ms/step - loss: 0.9798 -
accuracy: 0.6143 - val_loss: 1.0702 - val_accuracy: 0.5807 - lr: 0.0010
Epoch 29/150
428/428 [==============================] - 46s 107ms/step - loss: 0.9711 -
accuracy: 0.6149 - val_loss: 1.0661 - val_accuracy: 0.5783 - lr: 0.0010
Epoch 30/150
428/428 [==============================] - 46s 108ms/step - loss: 0.9532 -
accuracy: 0.6217 - val_loss: 1.0761 - val_accuracy: 0.5819 - lr: 0.0010
Epoch 31/150
428/428 [==============================] - 47s 109ms/step - loss: 0.9579 -
accuracy: 0.6213 - val_loss: 1.0622 - val_accuracy: 0.5825 - lr: 0.0010
Epoch 32/150
428/428 [==============================] - 53s 123ms/step - loss: 0.9479 -
accuracy: 0.6261 - val_loss: 1.0673 - val_accuracy: 0.5790 - lr: 0.0010
Epoch 33/150
428/428 [==============================] - 48s 113ms/step - loss: 0.9328 -
accuracy: 0.6326 - val_loss: 1.0757 - val_accuracy: 0.5762 - lr: 0.0010
Epoch 34/150
428/428 [==============================] - 46s 108ms/step - loss: 0.9176 -
accuracy: 0.6359 - val_loss: 1.0890 - val_accuracy: 0.5755 - lr: 0.0010
Epoch 35/150
428/428 [==============================] - 46s 108ms/step - loss: 0.9186 -
accuracy: 0.6375 - val_loss: 1.0645 - val_accuracy: 0.5873 - lr: 0.0010
Epoch 36/150
428/428 [==============================] - 46s 107ms/step - loss: 0.9070 -
accuracy: 0.6402 - val_loss: 1.1039 - val_accuracy: 0.5841 - lr: 0.0010
Epoch 37/150
428/428 [==============================] - 46s 106ms/step - loss: 0.9090 -
accuracy: 0.6440 - val_loss: 1.0909 - val_accuracy: 0.5831 - lr: 0.0010
Epoch 38/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8858 -
accuracy: 0.6497 - val_loss: 1.0922 - val_accuracy: 0.5818 - lr: 0.0010
Epoch 39/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8870 -
```

```
accuracy: 0.6517 - val_loss: 1.0543 - val_accuracy: 0.5930 - lr: 0.0010
Epoch 40/150
428/428 [==============================] - 46s 107ms/step - loss: 0.8766 -
accuracy: 0.6530 - val_loss: 1.0790 - val_accuracy: 0.5975 - lr: 0.0010
Epoch 41/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8724 -
accuracy: 0.6549 - val_loss: 1.0728 - val_accuracy: 0.5890 - lr: 0.0010
Epoch 42/150
428/428 [==============================] - 45s 104ms/step - loss: 0.8733 -
accuracy: 0.6571 - val_loss: 1.0906 - val_accuracy: 0.5862 - lr: 0.0010
Epoch 43/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8619 -
accuracy: 0.6612 - val_loss: 1.0792 - val_accuracy: 0.5989 - lr: 0.0010
Epoch 44/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8532 -
accuracy: 0.6665 - val_loss: 1.0972 - val_accuracy: 0.5868 - lr: 0.0010
Epoch 45/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8440 -
accuracy: 0.6688 - val_loss: 1.1514 - val_accuracy: 0.5858 - lr: 0.0010
Epoch 46/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8417 -
accuracy: 0.6733 - val_loss: 1.1010 - val_accuracy: 0.5944 - lr: 0.0010
Epoch 47/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8345 -
accuracy: 0.6721 - val_loss: 1.1102 - val_accuracy: 0.5913 - lr: 0.0010
Epoch 48/150
428/428 [==============================] - 45s 106ms/step - loss: 0.8316 -
accuracy: 0.6780 - val_loss: 1.0880 - val_accuracy: 0.5971 - lr: 0.0010
Epoch 49/150
428/428 [==============================] - 48s 112ms/step - loss: 0.8234 -
accuracy: 0.6794 - val_loss: 1.0992 - val_accuracy: 0.5949 - lr: 0.0010
Epoch 50/150
428/428 [==============================] - 47s 109ms/step - loss: 0.8270 -
accuracy: 0.6776 - val_loss: 1.1058 - val_accuracy: 0.5934 - lr: 0.0010
Epoch 51/150
428/428 [==============================] - 46s 108ms/step - loss: 0.8009 -
accuracy: 0.6873 - val_loss: 1.1092 - val_accuracy: 0.6013 - lr: 0.0010
Epoch 52/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7961 -
accuracy: 0.6906 - val_loss: 1.1345 - val_accuracy: 0.5993 - lr: 0.0010
Epoch 53/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7844 -
accuracy: 0.6932 - val_loss: 1.1335 - val_accuracy: 0.5975 - lr: 0.0010
Epoch 54/150
428/428 [==============================] - 46s 108ms/step - loss: 0.7822 -
accuracy: 0.6925 - val_loss: 1.1324 - val_accuracy: 0.5953 - lr: 0.0010
Epoch 55/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7794 -
```

```
accuracy: 0.6980 - val_loss: 1.1758 - val_accuracy: 0.5885 - lr: 0.0010
Epoch 56/150
428/428 [==============================] - 48s 111ms/step - loss: 0.7703 -
accuracy: 0.7010 - val_loss: 1.1579 - val_accuracy: 0.5891 - lr: 0.0010
Epoch 57/150
428/428 [==============================] - 46s 108ms/step - loss: 0.7706 -
accuracy: 0.6996 - val_loss: 1.1226 - val_accuracy: 0.5946 - lr: 0.0010
Epoch 58/150
428/428 [==============================] - 46s 108ms/step - loss: 0.7585 -
accuracy: 0.7052 - val_loss: 1.1373 - val_accuracy: 0.5991 - lr: 0.0010
Epoch 59/150
428/428 [==============================] - 46s 108ms/step - loss: 0.7549 -
accuracy: 0.7029 - val_loss: 1.1594 - val_accuracy: 0.5983 - lr: 0.0010
Epoch 60/150
428/428 [==============================] - 47s 109ms/step - loss: 0.7740 -
accuracy: 0.7018 - val_loss: 1.1185 - val_accuracy: 0.5942 - lr: 0.0010
Epoch 61/150
428/428 [==============================] - 45s 105ms/step - loss: 0.7419 -
accuracy: 0.7121 - val_loss: 1.1381 - val_accuracy: 0.6045 - lr: 0.0010
Epoch 62/150
428/428 [==============================] - 45s 106ms/step - loss: 0.7359 -
accuracy: 0.7148 - val_loss: 1.1761 - val_accuracy: 0.5990 - lr: 0.0010
Epoch 63/150
428/428 [==============================] - 45s 105ms/step - loss: 0.7323 -
accuracy: 0.7154 - val_loss: 1.1664 - val_accuracy: 0.6000 - lr: 0.0010
Epoch 64/150
428/428 [==============================] - 45s 106ms/step - loss: 0.7290 -
accuracy: 0.7183 - val_loss: 1.1589 - val_accuracy: 0.5950 - lr: 0.0010
Epoch 65/150
428/428 [==============================] - 45s 105ms/step - loss: 0.7382 -
accuracy: 0.7151 - val_loss: 1.1270 - val_accuracy: 0.5939 - lr: 0.0010
Epoch 66/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7207 -
accuracy: 0.7200 - val_loss: 1.1436 - val_accuracy: 0.6052 - lr: 0.0010
Epoch 67/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7104 -
accuracy: 0.7284 - val_loss: 1.1543 - val_accuracy: 0.6001 - lr: 0.0010
Epoch 68/150
428/428 [==============================] - 46s 107ms/step - loss: 0.7039 -
accuracy: 0.7267 - val_loss: 1.1850 - val_accuracy: 0.5956 - lr: 0.0010
Epoch 69/150
428/428 [==============================] - 45s 106ms/step - loss: 0.6959 -
accuracy: 0.7325 - val_loss: 1.1739 - val_accuracy: 0.6010 - lr: 0.0010
Epoch 70/150
428/428 [==============================] - 45s 105ms/step - loss: 0.7108 -
accuracy: 0.7283 - val_loss: 1.1884 - val_accuracy: 0.6000 - lr: 0.0010
Epoch 71/150
428/428 [==============================] - 45s 104ms/step - loss: 0.6888 -
```

```
accuracy: 0.7331 - val_loss: 1.2028 - val_accuracy: 0.6033 - lr: 0.0010
Epoch 72/150
428/428 [==============================] - 45s 105ms/step - loss: 0.7045 -
accuracy: 0.7318 - val_loss: 1.1904 - val_accuracy: 0.5936 - lr: 0.0010
Epoch 73/150
428/428 [==============================] - 45s 106ms/step - loss: 0.6695 -
accuracy: 0.7419 - val_loss: 1.2655 - val_accuracy: 0.5999 - lr: 0.0010
Epoch 74/150
428/428 [==============================] - 45s 106ms/step - loss: 0.6818 -
accuracy: 0.7359 - val_loss: 1.2182 - val_accuracy: 0.6029 - lr: 0.0010
Epoch 75/150
428/428 [==============================] - ETA: 0s - loss: 0.6801 - accuracy:
0.7385
Epoch 75: ReduceLROnPlateau reducing learning rate to 0.0004000000189989805.
428/428 [==============================] - 45s 104ms/step - loss: 0.6801 -
accuracy: 0.7385 - val_loss: 1.1851 - val_accuracy: 0.5946 - lr: 0.0010
Epoch 76/150
428/428 [==============================] - 45s 104ms/step - loss: 0.5937 -
accuracy: 0.7683 - val_loss: 1.2130 - val_accuracy: 0.6105 - lr: 4.0000e-04
Epoch 77/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5649 -
accuracy: 0.7826 - val_loss: 1.2463 - val_accuracy: 0.6037 - lr: 4.0000e-04
Epoch 78/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5575 -
accuracy: 0.7867 - val_loss: 1.2525 - val_accuracy: 0.6059 - lr: 4.0000e-04
Epoch 79/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5536 -
accuracy: 0.7892 - val_loss: 1.2816 - val_accuracy: 0.6114 - lr: 4.0000e-04
Epoch 80/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5457 -
accuracy: 0.7889 - val_loss: 1.2993 - val_accuracy: 0.6045 - lr: 4.0000e-04
Epoch 81/150
428/428 [==============================] - 45s 105ms/step - loss: 0.5442 -
accuracy: 0.7908 - val_loss: 1.2826 - val_accuracy: 0.6079 - lr: 4.0000e-04
Epoch 82/150
428/428 [==============================] - 45s 105ms/step - loss: 0.5327 -
accuracy: 0.7953 - val_loss: 1.3290 - val_accuracy: 0.6071 - lr: 4.0000e-04
Epoch 83/150
428/428 [==============================] - 45s 105ms/step - loss: 0.5327 -
accuracy: 0.7949 - val_loss: 1.3350 - val_accuracy: 0.6095 - lr: 4.0000e-04
Epoch 84/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5189 -
accuracy: 0.8004 - val_loss: 1.3665 - val_accuracy: 0.6097 - lr: 4.0000e-04
Epoch 85/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5159 -
accuracy: 0.8009 - val_loss: 1.3686 - val_accuracy: 0.6073 - lr: 4.0000e-04
Epoch 86/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5187 -
```

```
accuracy: 0.8031 - val_loss: 1.3671 - val_accuracy: 0.6071 - lr: 4.0000e-04
Epoch 87/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5095 -
accuracy: 0.8061 - val_loss: 1.3403 - val_accuracy: 0.6048 - lr: 4.0000e-04
Epoch 88/150
428/428 [==============================] - 45s 106ms/step - loss: 0.5069 -
accuracy: 0.8067 - val_loss: 1.3534 - val_accuracy: 0.6056 - lr: 4.0000e-04
Epoch 89/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5064 -
accuracy: 0.8059 - val_loss: 1.4083 - val_accuracy: 0.6075 - lr: 4.0000e-04
Epoch 90/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5065 -
accuracy: 0.8051 - val_loss: 1.4062 - val_accuracy: 0.6066 - lr: 4.0000e-04
Epoch 91/150
428/428 [==============================] - 46s 107ms/step - loss: 0.5020 -
accuracy: 0.8097 - val_loss: 1.4025 - val_accuracy: 0.6028 - lr: 4.0000e-04
Epoch 92/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4912 -
accuracy: 0.8127 - val_loss: 1.4420 - val_accuracy: 0.6040 - lr: 4.0000e-04
Epoch 93/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4898 -
accuracy: 0.8146 - val_loss: 1.4089 - val_accuracy: 0.6061 - lr: 4.0000e-04
Epoch 94/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4860 -
accuracy: 0.8147 - val_loss: 1.3990 - val_accuracy: 0.6077 - lr: 4.0000e-04
Epoch 95/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4807 -
accuracy: 0.8167 - val_loss: 1.3911 - val_accuracy: 0.6053 - lr: 4.0000e-04
Epoch 96/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4761 -
accuracy: 0.8169 - val_loss: 1.4931 - val_accuracy: 0.6069 - lr: 4.0000e-04
Epoch 97/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4800 -
accuracy: 0.8194 - val_loss: 1.4390 - val_accuracy: 0.6063 - lr: 4.0000e-04
Epoch 98/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4738 -
accuracy: 0.8185 - val_loss: 1.5033 - val_accuracy: 0.6049 - lr: 4.0000e-04
Epoch 99/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4693 -
accuracy: 0.8220 - val_loss: 1.4589 - val_accuracy: 0.6075 - lr: 4.0000e-04
Epoch 100/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4674 -
accuracy: 0.8208 - val_loss: 1.4574 - val_accuracy: 0.6066 - lr: 4.0000e-04
Epoch 101/150
428/428 [==============================] - 45s 106ms/step - loss: 0.4656 -
accuracy: 0.8232 - val_loss: 1.4676 - val_accuracy: 0.6046 - lr: 4.0000e-04
Epoch 102/150
428/428 [==============================] - 45s 105ms/step - loss: 0.4608 -
```

```
accuracy: 0.8232 - val_loss: 1.4943 - val_accuracy: 0.6062 - lr: 4.0000e-04
Epoch 103/150
428/428 [==============================] - 45s 106ms/step - loss: 0.4607 -
accuracy: 0.8248 - val_loss: 1.4842 - val_accuracy: 0.6084 - lr: 4.0000e-04
Epoch 104/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4543 -
accuracy: 0.8284 - val_loss: 1.4695 - val_accuracy: 0.6044 - lr: 4.0000e-04
Epoch 105/150
428/428 [==============================] - 45s 106ms/step - loss: 0.4471 -
accuracy: 0.8288 - val_loss: 1.5518 - val_accuracy: 0.6052 - lr: 4.0000e-04
Epoch 106/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4595 -
accuracy: 0.8259 - val_loss: 1.4669 - val_accuracy: 0.6096 - lr: 4.0000e-04
Epoch 107/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4437 -
accuracy: 0.8305 - val_loss: 1.5109 - val_accuracy: 0.6015 - lr: 4.0000e-04
Epoch 108/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4437 -
accuracy: 0.8336 - val_loss: 1.5529 - val_accuracy: 0.6032 - lr: 4.0000e-04
Epoch 109/150
428/428 [==============================] - 45s 105ms/step - loss: 0.4363 -
accuracy: 0.8346 - val_loss: 1.5108 - val_accuracy: 0.6033 - lr: 4.0000e-04
Epoch 110/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4342 -
accuracy: 0.8362 - val_loss: 1.5395 - val_accuracy: 0.6002 - lr: 4.0000e-04
Epoch 111/150
428/428 [==============================] - 46s 108ms/step - loss: 0.4274 -
accuracy: 0.8379 - val_loss: 1.5684 - val_accuracy: 0.6044 - lr: 4.0000e-04
Epoch 112/150
428/428 [==============================] - 46s 107ms/step - loss: 0.4364 -
accuracy: 0.8357 - val_loss: 1.5834 - val_accuracy: 0.6021 - lr: 4.0000e-04
Epoch 113/150
428/428 [==============================] - ETA: 0s - loss: 0.4315 - accuracy:
0.8357
Epoch 113: ReduceLROnPlateau reducing learning rate to 0.00016000000759959222.
428/428 [==============================] - 46s 107ms/step - loss: 0.4315 -
accuracy: 0.8357 - val_loss: 1.5445 - val_accuracy: 0.6061 - lr: 4.0000e-04
Epoch 114/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3955 -
accuracy: 0.8506 - val_loss: 1.5956 - val_accuracy: 0.6098 - lr: 1.6000e-04
Epoch 115/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3853 -
accuracy: 0.8544 - val_loss: 1.5945 - val_accuracy: 0.6069 - lr: 1.6000e-04
Epoch 116/150
428/428 [==============================] - 46s 107ms/step - loss: 0.3780 -
accuracy: 0.8570 - val_loss: 1.6369 - val_accuracy: 0.6058 - lr: 1.6000e-04
Epoch 117/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3785 -
```

```
accuracy: 0.8575 - val_loss: 1.6370 - val_accuracy: 0.6061 - lr: 1.6000e-04
Epoch 118/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3773 -
accuracy: 0.8565 - val_loss: 1.6278 - val_accuracy: 0.6057 - lr: 1.6000e-04
Epoch 119/150
428/428 [==============================] - 47s 109ms/step - loss: 0.3718 -
accuracy: 0.8604 - val_loss: 1.6536 - val_accuracy: 0.6041 - lr: 1.6000e-04
Epoch 120/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3694 -
accuracy: 0.8589 - val_loss: 1.6584 - val_accuracy: 0.6073 - lr: 1.6000e-04
Epoch 121/150
428/428 [==============================] - 45s 105ms/step - loss: 0.3667 -
accuracy: 0.8609 - val_loss: 1.6914 - val_accuracy: 0.6059 - lr: 1.6000e-04
Epoch 122/150
428/428 [==============================] - 46s 107ms/step - loss: 0.3704 -
accuracy: 0.8594 - val_loss: 1.6748 - val_accuracy: 0.6089 - lr: 1.6000e-04
Epoch 123/150
428/428 [==============================] - 46s 107ms/step - loss: 0.3613 -
accuracy: 0.8631 - val_loss: 1.7108 - val_accuracy: 0.6061 - lr: 1.6000e-04
Epoch 124/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3571 -
accuracy: 0.8648 - val_loss: 1.7230 - val_accuracy: 0.6057 - lr: 1.6000e-04
Epoch 125/150
428/428 [==============================] - 46s 107ms/step - loss: 0.3655 -
accuracy: 0.8648 - val_loss: 1.6846 - val_accuracy: 0.6083 - lr: 1.6000e-04
Epoch 126/150
428/428 [==============================] - ETA: 0s - loss: 0.3591 - accuracy:
0.8665
Epoch 126: ReduceLROnPlateau reducing learning rate to 6.40000042039901e-05.
428/428 [==============================] - 46s 107ms/step - loss: 0.3591 -
accuracy: 0.8665 - val_loss: 1.7094 - val_accuracy: 0.6072 - lr: 1.6000e-04
Epoch 127/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3492 -
accuracy: 0.8688 - val_loss: 1.7153 - val_accuracy: 0.6086 - lr: 6.4000e-05
Epoch 128/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3397 -
accuracy: 0.8741 - val_loss: 1.7248 - val_accuracy: 0.6096 - lr: 6.4000e-05
Epoch 129/150
428/428 [==============================] - 46s 108ms/step - loss: 0.3348 -
accuracy: 0.8740 - val_loss: 1.7247 - val_accuracy: 0.6073 - lr: 6.4000e-05
Epoch 130/150
428/428 [==============================] - 47s 110ms/step - loss: 0.3408 -
accuracy: 0.8700 - val_loss: 1.7285 - val_accuracy: 0.6075 - lr: 6.4000e-05
Epoch 131/150
428/428 [==============================] - ETA: 0s - loss: 0.3379 - accuracy:
0.8716
Epoch 131: ReduceLROnPlateau reducing learning rate to 2.560000284574926e-05.
428/428 [==============================] - 47s 109ms/step - loss: 0.3379 -
```

```
accuracy: 0.8716 - val_loss: 1.7278 - val_accuracy: 0.6098 - lr: 6.4000e-05
Epoch 132/150
428/428 [==============================] - 47s 110ms/step - loss: 0.3340 -
accuracy: 0.8756 - val_loss: 1.7414 - val_accuracy: 0.6075 - lr: 2.5600e-05
Epoch 133/150
428/428 [==============================] - 47s 109ms/step - loss: 0.3307 -
accuracy: 0.8752 - val_loss: 1.7472 - val_accuracy: 0.6070 - lr: 2.5600e-05
Epoch 134/150
428/428 [==============================] - 46s 109ms/step - loss: 0.3239 -
accuracy: 0.8797 - val_loss: 1.7489 - val_accuracy: 0.6079 - lr: 2.5600e-05
Epoch 135/150
428/428 [==============================] - 47s 110ms/step - loss: 0.3283 -
accuracy: 0.8756 - val_loss: 1.7621 - val_accuracy: 0.6093 - lr: 2.5600e-05
Epoch 136/150
428/428 [==============================] - ETA: 0s - loss: 0.3316 - accuracy:
0.8773
Epoch 136: ReduceLROnPlateau reducing learning rate to 1.0240000847261399e-05.
428/428 [==============================] - 47s 110ms/step - loss: 0.3316 -
accuracy: 0.8773 - val_loss: 1.7552 - val_accuracy: 0.6087 - lr: 2.5600e-05
Epoch 137/150
428/428 [==============================] - 47s 109ms/step - loss: 0.3230 -
accuracy: 0.8792 - val_loss: 1.7591 - val_accuracy: 0.6085 - lr: 1.0240e-05
Epoch 138/150
428/428 [==============================] - 46s 107ms/step - loss: 0.3244 -
accuracy: 0.8785 - val_loss: 1.7632 - val_accuracy: 0.6084 - lr: 1.0240e-05
Epoch 139/150
428/428 [==============================] - ETA: 0s - loss: 0.3268 - accuracy:
0.8778
Epoch 139: ReduceLROnPlateau reducing learning rate to 4.09600033890456e-06.
428/428 [==============================] - 46s 109ms/step - loss: 0.3268 -
accuracy: 0.8778 - val_loss: 1.7678 - val_accuracy: 0.6079 - lr: 1.0240e-05
Epoch 140/150
428/428 [==============================] - 46s 109ms/step - loss: 0.3229 -
accuracy: 0.8806 - val_loss: 1.7668 - val_accuracy: 0.6084 - lr: 4.0960e-06
Epoch 141/150
428/428 [==============================] - ETA: 0s - loss: 0.3240 - accuracy:
0.8808
Epoch 141: ReduceLROnPlateau reducing learning rate to 1.6384001355618238e-06.
428/428 [==============================] - 47s 109ms/step - loss: 0.3240 -
accuracy: 0.8808 - val_loss: 1.7646 - val_accuracy: 0.6071 - lr: 4.0960e-06
Epoch 142/150
428/428 [==============================] - 47s 109ms/step - loss: 0.3247 -
accuracy: 0.8809 - val_loss: 1.7643 - val_accuracy: 0.6067 - lr: 1.6384e-06
Epoch 143/150
428/428 [==============================] - ETA: 0s - loss: 0.3229 - accuracy:
0.8779
Epoch 143: ReduceLROnPlateau reducing learning rate to 6.553600542247295e-07.
428/428 [==============================] - 47s 109ms/step - loss: 0.3229 -
```

```
accuracy: 0.8779 - val_loss: 1.7640 - val_accuracy: 0.6070 - lr: 1.6384e-06
Epoch 144/150
428/428 [==============================] - 47s 109ms/step - loss: 0.3231 -
accuracy: 0.8790 - val_loss: 1.7643 - val_accuracy: 0.6073 - lr: 6.5536e-07
Epoch 145/150
428/428 [==============================] - ETA: 0s - loss: 0.3230 - accuracy:
0.8780
Epoch 145: ReduceLROnPlateau reducing learning rate to 2.6214402168989184e-07.
428/428 [==============================] - 47s 111ms/step - loss: 0.3230 -
accuracy: 0.8780 - val_loss: 1.7644 - val_accuracy: 0.6073 - lr: 6.5536e-07
Epoch 146/150
428/428 [==============================] - 47s 110ms/step - loss: 0.3220 -
accuracy: 0.8771 - val_loss: 1.7644 - val_accuracy: 0.6072 - lr: 2.6214e-07
Epoch 147/150
428/428 [==============================] - 47s 110ms/step - loss: 0.3221 -
accuracy: 0.8792 - val_loss: 1.7643 - val_accuracy: 0.6073 - lr: 2.6214e-07
Epoch 148/150
428/428 [==============================] - ETA: 0s - loss: 0.3244 - accuracy:
0.8783
Epoch 148: ReduceLROnPlateau reducing learning rate to 1.0485761094969349e-07.
428/428 [==============================] - 47s 111ms/step - loss: 0.3244 -
accuracy: 0.8783 - val_loss: 1.7644 - val_accuracy: 0.6073 - lr: 2.6214e-07
Epoch 149/150
428/428 [==============================] - 47s 111ms/step - loss: 0.3260 -
accuracy: 0.8785 - val_loss: 1.7644 - val_accuracy: 0.6073 - lr: 1.0486e-07
Epoch 150/150
428/428 [==============================] - ETA: 0s - loss: 0.3231 - accuracy:
0.8793
Epoch 150: ReduceLROnPlateau reducing learning rate to 4.1943044948311586e-08.
428/428 [==============================] - 47s 110ms/step - loss: 0.3231 -
accuracy: 0.8793 - val_loss: 1.7643 - val_accuracy: 0.6074 - lr: 1.0486e-07
```

[11]:
```
!mkdir saved_model_Small_CNN_Extended_Dataset
model.save('saved_model_Small_CNN_Extended_Dataset/
 ↪ser_SmallCNN_ExtendedDataset_model')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _update_step_xla while saving (showing 5 of 5).
These functions will not be directly callable after loading.
```

```
INFO:tensorflow:Assets written to:
saved_model_Small_CNN_Extended_Dataset/ser_SmallCNN_ExtendedDataset_model\assets
```

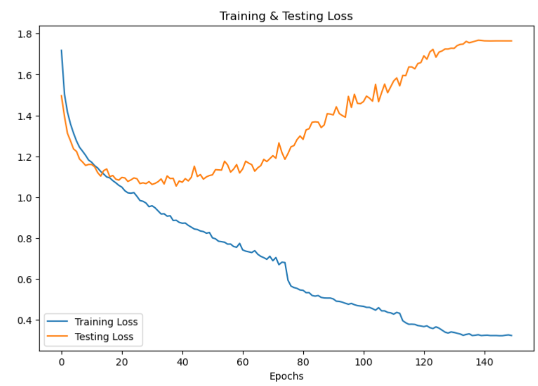

```
INFO:tensorflow:Assets written to:
saved_model_Small_CNN_Extended_Dataset/ser_SmallCNN_ExtendedDataset_model\assets
```

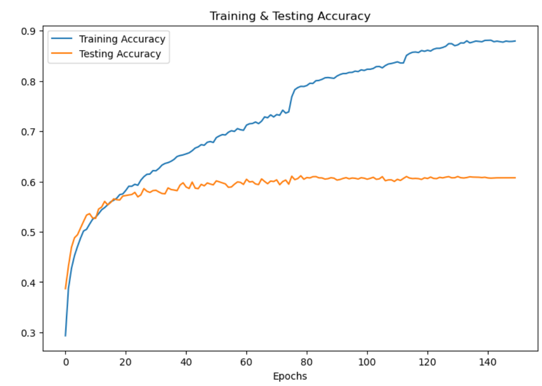

[12]:
```
print("Accuracy of our model on test data : " , model.
 ↪evaluate(X_test,y_test)[1]*100 , "%")
```

```
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
test_acc = history.history['val_accuracy']
test_loss = history.history['val_loss']
epochs = [i for i in range(len(train_loss))]
# epochs = range(50)
print(epochs)

fig.set_size_inches(20,6)
ax[0].plot(epochs , train_loss , label = 'Training Loss')
ax[0].plot(epochs , test_loss , label = 'Testing Loss')
ax[0].set_title('Training & Testing Loss')
ax[0].legend()
ax[0].set_xlabel("Epochs")

ax[1].plot(epochs , train_acc , label = 'Training Accuracy')
ax[1].plot(epochs , test_acc , label = 'Testing Accuracy')
ax[1].set_title('Training & Testing Accuracy')
ax[1].legend()
ax[1].set_xlabel("Epochs")
plt.show()

# predicting on test data.
pred_test = model.predict(X_test)
print(pred_test)

y_pred = encoder.inverse_transform(pred_test)
print(y_pred)
print(y_pred.shape)
y_test = encoder.inverse_transform(y_test)
df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()
df.head(100)
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] , columns = [i␣
↪for i in encoder.categories_])
sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, annot=True,␣
↪fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()
```

```python
print(classification_report(y_test, y_pred))
```

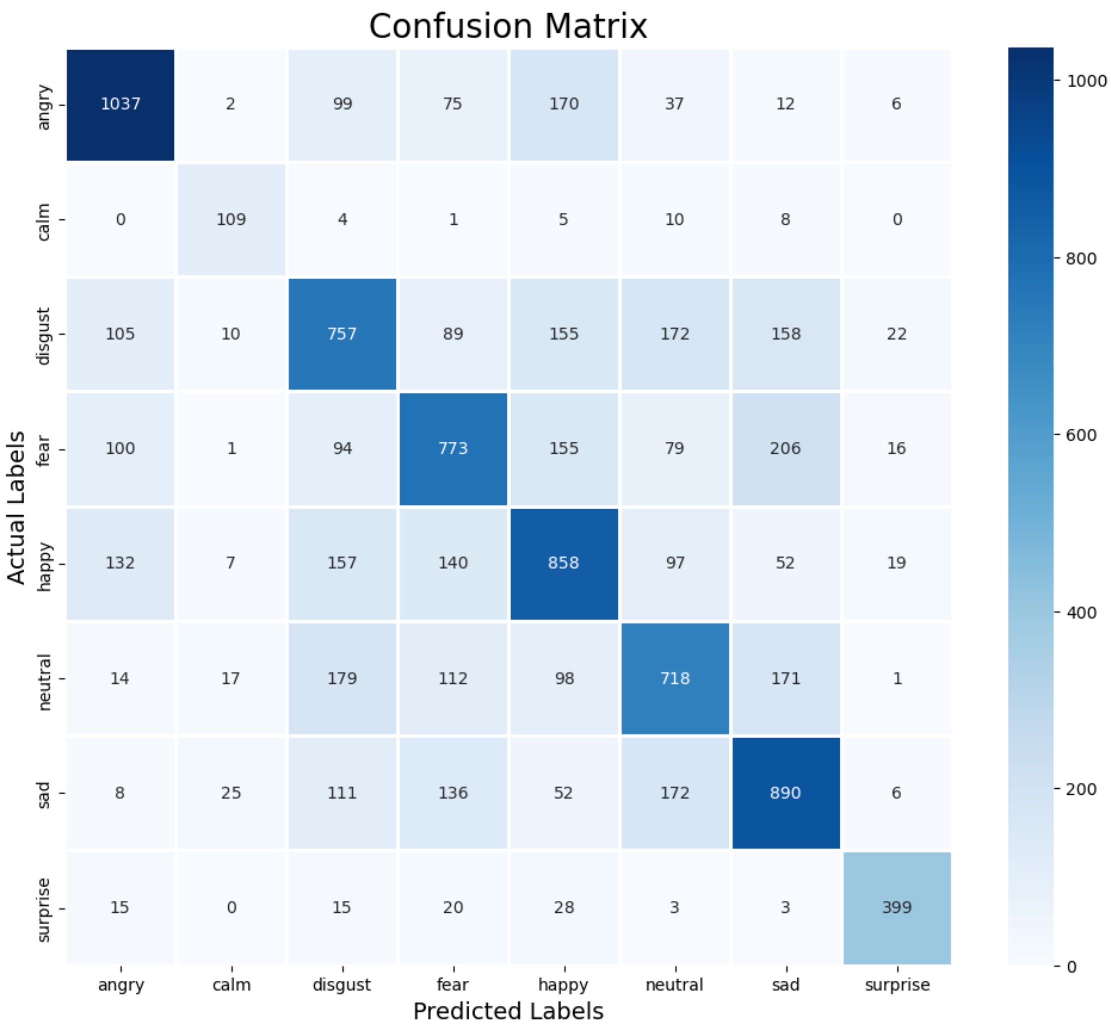
```
286/286 [==============================] - 3s 10ms/step - loss: 1.7643 -
accuracy: 0.6074
Accuracy of our model on test data :  60.74326038360596 %
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148,
149]
```






```
286/286 [==============================] - 3s 10ms/step
[[9.1979477e-05 1.8227789e-09 8.6123221e-02 … 1.6990067e-02
  6.8871325e-01 8.9337964e-06]
 [6.9614122e-18 0.0000000e+00 1.0000000e+00 … 1.4738214e-24
  2.4283428e-26 5.1146866e-34]
 [1.0000000e+00 0.0000000e+00 1.5903262e-16 … 0.0000000e+00
  4.4526717e-36 1.0735920e-31]
 …
 [8.8169783e-01 5.5915426e-21 3.3008533e-03 … 3.6510426e-07
  3.4631900e-10 1.1212846e-12]
 [9.5625952e-02 3.7250134e-12 1.3932598e-01 … 4.3826643e-04
  8.6154432e-06 1.0816229e-05]
 [0.0000000e+00 0.0000000e+00 5.1260548e-19 … 1.9317507e-18
  1.0000000e+00 0.0000000e+00]]
[['sad']
 ['disgust']
 ['angry']
 …
 ['angry']
```

```
 ['happy']
 ['sad']]
(9122, 1)
```



	precision	recall	f1-score	support
angry	0.73	0.72	0.73	1438
calm	0.64	0.80	0.71	137
disgust	0.53	0.52	0.52	1468
fear	0.57	0.54	0.56	1424
happy	0.56	0.59	0.58	1462
neutral	0.56	0.55	0.55	1310
sad	0.59	0.64	0.61	1400
surprise	0.85	0.83	0.84	483
accuracy			0.61	9122

```
   macro avg       0.63      0.65      0.64      9122
weighted avg       0.61      0.61      0.61      9122
```

[9]:

```
model=Sequential()
model.add(Conv1D(512, kernel_size=5, strides=1, padding='same',␣
 ↪activation='tanh', input_shape=(X_train.shape[1], 1)))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(256, kernel_size=5, strides=1, padding='same',␣
 ↪activation='tanh'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same',␣
 ↪activation='tanh'))
model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))
model.add(Dropout(0.2))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same',␣
 ↪activation='tanh'))
model.add(MaxPooling1D(pool_size=3, strides = 2, padding = 'same'))

model.add(Conv1D(128, kernel_size=5, strides=1, padding='same',␣
 ↪activation='tanh'))
model.add(MaxPooling1D(pool_size=2, strides = 2, padding = 'same'))
model.add(Dropout(0.2))

model.add(Flatten())
model.add(Dense(units=1024, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=512, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(units=256, activation='relu'))
model.add(Dropout(0.1))

model.add(Dense(units=256, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=256, activation='relu'))
model.add(Dropout(0.2))

model.add(Dense(units=32, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=8, activation='softmax'))
```

```
model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy' , metrics␣
↪= [tf.keras.metrics.CategoricalCrossentropy(), 'accuracy'])

model.summary()
```

```
Model: "sequential"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 170, 512)          3072

 max_pooling1d (MaxPooling1D  (None, 85, 512)          0
 )

 conv1d_1 (Conv1D)           (None, 85, 256)           655616

 max_pooling1d_1 (MaxPooling  (None, 43, 256)          0
 1D)

 conv1d_2 (Conv1D)           (None, 43, 128)           163968

 max_pooling1d_2 (MaxPooling  (None, 22, 128)          0
 1D)

 dropout (Dropout)           (None, 22, 128)           0

 conv1d_3 (Conv1D)           (None, 22, 128)           82048

 max_pooling1d_3 (MaxPooling  (None, 11, 128)          0
 1D)

 conv1d_4 (Conv1D)           (None, 11, 128)           82048

 max_pooling1d_4 (MaxPooling  (None, 6, 128)           0
 1D)

 dropout_1 (Dropout)         (None, 6, 128)            0

 flatten (Flatten)           (None, 768)               0

 dense (Dense)               (None, 1024)              787456

 dropout_2 (Dropout)         (None, 1024)              0

 dense_1 (Dense)             (None, 512)               524800

 dropout_3 (Dropout)         (None, 512)               0
```

```
 dense_2 (Dense)             (None, 256)               131328

 dropout_4 (Dropout)         (None, 256)               0

 dense_3 (Dense)             (None, 256)               65792

 dropout_5 (Dropout)         (None, 256)               0

 dense_4 (Dense)             (None, 256)               65792

 dropout_6 (Dropout)         (None, 256)               0

 dense_5 (Dense)             (None, 32)                8224

 dropout_7 (Dropout)         (None, 32)                0

 dense_6 (Dense)             (None, 8)                 264

=================================================================
Total params: 2,570,408
Trainable params: 2,570,408
Non-trainable params: 0
_________________________________________________________________
```

[10]:
```
rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=1, patience=2,␣
 ↪min_lr=0.00000000001)
history=model.fit(X_train, y_train, batch_size=64, epochs=150,␣
 ↪validation_data=(X_test, y_test), callbacks=[rlrp])
```

```
Epoch 1/150
428/428 [==============================] - 84s 192ms/step - loss: 1.7998 -
categorical_crossentropy: 1.7998 - accuracy: 0.2644 - val_loss: 1.6516 -
val_categorical_crossentropy: 1.6516 - val_accuracy: 0.3040 - lr: 0.0010
Epoch 2/150
428/428 [==============================] - 82s 191ms/step - loss: 1.6306 -
categorical_crossentropy: 1.6306 - accuracy: 0.3365 - val_loss: 1.5529 -
val_categorical_crossentropy: 1.5529 - val_accuracy: 0.3831 - lr: 0.0010
Epoch 3/150
428/428 [==============================] - 81s 190ms/step - loss: 1.5029 -
categorical_crossentropy: 1.5029 - accuracy: 0.4013 - val_loss: 1.3680 -
val_categorical_crossentropy: 1.3680 - val_accuracy: 0.4418 - lr: 0.0010
Epoch 4/150
428/428 [==============================] - 82s 191ms/step - loss: 1.4359 -
categorical_crossentropy: 1.4359 - accuracy: 0.4275 - val_loss: 1.3446 -
val_categorical_crossentropy: 1.3446 - val_accuracy: 0.4504 - lr: 0.0010
Epoch 5/150
428/428 [==============================] - 81s 190ms/step - loss: 1.3849 -
categorical_crossentropy: 1.3849 - accuracy: 0.4457 - val_loss: 1.3429 -
```

```
val_categorical_crossentropy: 1.3429 - val_accuracy: 0.4549 - lr: 0.0010
Epoch 6/150
428/428 [==============================] - 81s 190ms/step - loss: 1.3871 -
categorical_crossentropy: 1.3871 - accuracy: 0.4496 - val_loss: 1.3432 -
val_categorical_crossentropy: 1.3432 - val_accuracy: 0.4672 - lr: 0.0010
Epoch 7/150
428/428 [==============================] - 81s 190ms/step - loss: 1.3495 -
categorical_crossentropy: 1.3495 - accuracy: 0.4593 - val_loss: 1.2740 -
val_categorical_crossentropy: 1.2740 - val_accuracy: 0.4918 - lr: 0.0010
Epoch 8/150
428/428 [==============================] - 82s 191ms/step - loss: 1.3248 -
categorical_crossentropy: 1.3248 - accuracy: 0.4712 - val_loss: 1.2491 -
val_categorical_crossentropy: 1.2491 - val_accuracy: 0.4923 - lr: 0.0010
Epoch 9/150
428/428 [==============================] - 82s 191ms/step - loss: 1.3296 -
categorical_crossentropy: 1.3296 - accuracy: 0.4689 - val_loss: 1.2658 -
val_categorical_crossentropy: 1.2658 - val_accuracy: 0.4847 - lr: 0.0010
Epoch 10/150
428/428 [==============================] - 82s 190ms/step - loss: 1.3005 -
categorical_crossentropy: 1.3005 - accuracy: 0.4800 - val_loss: 1.2645 -
val_categorical_crossentropy: 1.2645 - val_accuracy: 0.4863 - lr: 0.0010
Epoch 11/150
428/428 [==============================] - 81s 190ms/step - loss: 1.2968 -
categorical_crossentropy: 1.2968 - accuracy: 0.4826 - val_loss: 1.2358 -
val_categorical_crossentropy: 1.2358 - val_accuracy: 0.5020 - lr: 0.0010
Epoch 12/150
428/428 [==============================] - 81s 190ms/step - loss: 1.3046 -
categorical_crossentropy: 1.3046 - accuracy: 0.4839 - val_loss: 1.2487 -
val_categorical_crossentropy: 1.2487 - val_accuracy: 0.5032 - lr: 0.0010
Epoch 13/150
428/428 [==============================] - 81s 190ms/step - loss: 1.2774 -
categorical_crossentropy: 1.2774 - accuracy: 0.4953 - val_loss: 1.2394 -
val_categorical_crossentropy: 1.2394 - val_accuracy: 0.5027 - lr: 0.0010
Epoch 14/150
428/428 [==============================] - 81s 189ms/step - loss: 1.2722 -
categorical_crossentropy: 1.2722 - accuracy: 0.4965 - val_loss: 1.2285 -
val_categorical_crossentropy: 1.2285 - val_accuracy: 0.5008 - lr: 0.0010
Epoch 15/150
428/428 [==============================] - 81s 189ms/step - loss: 1.2719 -
categorical_crossentropy: 1.2719 - accuracy: 0.4989 - val_loss: 1.2126 -
val_categorical_crossentropy: 1.2126 - val_accuracy: 0.5193 - lr: 0.0010
Epoch 16/150
428/428 [==============================] - 81s 189ms/step - loss: 1.2753 -
categorical_crossentropy: 1.2753 - accuracy: 0.4990 - val_loss: 1.2321 -
val_categorical_crossentropy: 1.2321 - val_accuracy: 0.4995 - lr: 0.0010
Epoch 17/150
428/428 [==============================] - 80s 188ms/step - loss: 1.2690 -
categorical_crossentropy: 1.2690 - accuracy: 0.5005 - val_loss: 1.2092 -
```

```
val_categorical_crossentropy: 1.2092 - val_accuracy: 0.5152 - lr: 0.0010
Epoch 18/150
428/428 [==============================] - 82s 191ms/step - loss: 1.2543 -
categorical_crossentropy: 1.2543 - accuracy: 0.5105 - val_loss: 1.1940 -
val_categorical_crossentropy: 1.1940 - val_accuracy: 0.5204 - lr: 0.0010
Epoch 19/150
428/428 [==============================] - 81s 189ms/step - loss: 1.2544 -
categorical_crossentropy: 1.2544 - accuracy: 0.5097 - val_loss: 1.2262 -
val_categorical_crossentropy: 1.2262 - val_accuracy: 0.5145 - lr: 0.0010
Epoch 20/150
428/428 [==============================] - 80s 188ms/step - loss: 1.2441 -
categorical_crossentropy: 1.2441 - accuracy: 0.5126 - val_loss: 1.1996 -
val_categorical_crossentropy: 1.1996 - val_accuracy: 0.5181 - lr: 0.0010
Epoch 21/150
428/428 [==============================] - 80s 188ms/step - loss: 1.2506 -
categorical_crossentropy: 1.2506 - accuracy: 0.5077 - val_loss: 1.2161 -
val_categorical_crossentropy: 1.2161 - val_accuracy: 0.5227 - lr: 0.0010
Epoch 22/150
428/428 [==============================] - 80s 188ms/step - loss: 1.2416 -
categorical_crossentropy: 1.2416 - accuracy: 0.5145 - val_loss: 1.1962 -
val_categorical_crossentropy: 1.1962 - val_accuracy: 0.5125 - lr: 0.0010
Epoch 23/150
428/428 [==============================] - 80s 186ms/step - loss: 1.2371 -
categorical_crossentropy: 1.2371 - accuracy: 0.5153 - val_loss: 1.2083 -
val_categorical_crossentropy: 1.2083 - val_accuracy: 0.5178 - lr: 0.0010
Epoch 24/150
428/428 [==============================] - 80s 186ms/step - loss: 1.2468 -
categorical_crossentropy: 1.2468 - accuracy: 0.5121 - val_loss: 1.2291 -
val_categorical_crossentropy: 1.2291 - val_accuracy: 0.5257 - lr: 0.0010
Epoch 25/150
428/428 [==============================] - ETA: 0s - loss: 1.2481 -
categorical_crossentropy: 1.2481 - accuracy: 0.5125
Epoch 25: ReduceLROnPlateau reducing learning rate to 0.0004000000189989805.
428/428 [==============================] - 80s 186ms/step - loss: 1.2481 -
categorical_crossentropy: 1.2481 - accuracy: 0.5125 - val_loss: 1.2090 -
val_categorical_crossentropy: 1.2090 - val_accuracy: 0.5247 - lr: 0.0010
Epoch 26/150
428/428 [==============================] - 80s 187ms/step - loss: 1.1836 -
categorical_crossentropy: 1.1836 - accuracy: 0.5318 - val_loss: 1.1417 -
val_categorical_crossentropy: 1.1417 - val_accuracy: 0.5478 - lr: 4.0000e-04
Epoch 27/150
428/428 [==============================] - 80s 187ms/step - loss: 1.1587 -
categorical_crossentropy: 1.1587 - accuracy: 0.5433 - val_loss: 1.1453 -
val_categorical_crossentropy: 1.1453 - val_accuracy: 0.5409 - lr: 4.0000e-04
Epoch 28/150
428/428 [==============================] - 80s 186ms/step - loss: 1.1442 -
categorical_crossentropy: 1.1442 - accuracy: 0.5474 - val_loss: 1.1269 -
val_categorical_crossentropy: 1.1269 - val_accuracy: 0.5516 - lr: 4.0000e-04
```

```
Epoch 29/150
428/428 [==============================] - 80s 187ms/step - loss: 1.1323 -
categorical_crossentropy: 1.1323 - accuracy: 0.5543 - val_loss: 1.1245 -
val_categorical_crossentropy: 1.1245 - val_accuracy: 0.5462 - lr: 4.0000e-04
Epoch 30/150
428/428 [==============================] - 80s 188ms/step - loss: 1.1199 -
categorical_crossentropy: 1.1199 - accuracy: 0.5613 - val_loss: 1.1204 -
val_categorical_crossentropy: 1.1204 - val_accuracy: 0.5539 - lr: 4.0000e-04
Epoch 31/150
428/428 [==============================] - 80s 187ms/step - loss: 1.1117 -
categorical_crossentropy: 1.1117 - accuracy: 0.5625 - val_loss: 1.1250 -
val_categorical_crossentropy: 1.1250 - val_accuracy: 0.5509 - lr: 4.0000e-04
Epoch 32/150
428/428 [==============================] - 80s 187ms/step - loss: 1.1221 -
categorical_crossentropy: 1.1221 - accuracy: 0.5570 - val_loss: 1.1277 -
val_categorical_crossentropy: 1.1277 - val_accuracy: 0.5499 - lr: 4.0000e-04
Epoch 33/150
428/428 [==============================] - 80s 188ms/step - loss: 1.1055 -
categorical_crossentropy: 1.1055 - accuracy: 0.5667 - val_loss: 1.1040 -
val_categorical_crossentropy: 1.1040 - val_accuracy: 0.5634 - lr: 4.0000e-04
Epoch 34/150
428/428 [==============================] - 81s 188ms/step - loss: 1.0916 -
categorical_crossentropy: 1.0916 - accuracy: 0.5722 - val_loss: 1.1037 -
val_categorical_crossentropy: 1.1037 - val_accuracy: 0.5596 - lr: 4.0000e-04
Epoch 35/150
428/428 [==============================] - 81s 189ms/step - loss: 1.0883 -
categorical_crossentropy: 1.0883 - accuracy: 0.5732 - val_loss: 1.0981 -
val_categorical_crossentropy: 1.0981 - val_accuracy: 0.5619 - lr: 4.0000e-04
Epoch 36/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0754 -
categorical_crossentropy: 1.0754 - accuracy: 0.5821 - val_loss: 1.0999 -
val_categorical_crossentropy: 1.0999 - val_accuracy: 0.5669 - lr: 4.0000e-04
Epoch 37/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0778 -
categorical_crossentropy: 1.0778 - accuracy: 0.5779 - val_loss: 1.1058 -
val_categorical_crossentropy: 1.1058 - val_accuracy: 0.5648 - lr: 4.0000e-04
Epoch 38/150
428/428 [==============================] - 79s 185ms/step - loss: 1.0682 -
categorical_crossentropy: 1.0682 - accuracy: 0.5821 - val_loss: 1.0921 -
val_categorical_crossentropy: 1.0921 - val_accuracy: 0.5731 - lr: 4.0000e-04
Epoch 39/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0655 -
categorical_crossentropy: 1.0655 - accuracy: 0.5855 - val_loss: 1.0985 -
val_categorical_crossentropy: 1.0985 - val_accuracy: 0.5697 - lr: 4.0000e-04
Epoch 40/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0567 -
categorical_crossentropy: 1.0567 - accuracy: 0.5900 - val_loss: 1.0962 -
val_categorical_crossentropy: 1.0962 - val_accuracy: 0.5677 - lr: 4.0000e-04
```

```
Epoch 41/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0657 -
categorical_crossentropy: 1.0657 - accuracy: 0.5858 - val_loss: 1.0970 -
val_categorical_crossentropy: 1.0970 - val_accuracy: 0.5692 - lr: 4.0000e-04
Epoch 42/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0462 -
categorical_crossentropy: 1.0462 - accuracy: 0.5946 - val_loss: 1.0946 -
val_categorical_crossentropy: 1.0946 - val_accuracy: 0.5813 - lr: 4.0000e-04
Epoch 43/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0380 -
categorical_crossentropy: 1.0380 - accuracy: 0.5961 - val_loss: 1.0956 -
val_categorical_crossentropy: 1.0956 - val_accuracy: 0.5787 - lr: 4.0000e-04
Epoch 44/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0299 -
categorical_crossentropy: 1.0299 - accuracy: 0.6014 - val_loss: 1.1009 -
val_categorical_crossentropy: 1.1009 - val_accuracy: 0.5718 - lr: 4.0000e-04
Epoch 45/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0278 -
categorical_crossentropy: 1.0278 - accuracy: 0.5999 - val_loss: 1.0854 -
val_categorical_crossentropy: 1.0854 - val_accuracy: 0.5782 - lr: 4.0000e-04
Epoch 46/150
428/428 [==============================] - 80s 188ms/step - loss: 1.0190 -
categorical_crossentropy: 1.0190 - accuracy: 0.6083 - val_loss: 1.0815 -
val_categorical_crossentropy: 1.0815 - val_accuracy: 0.5747 - lr: 4.0000e-04
Epoch 47/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0197 -
categorical_crossentropy: 1.0197 - accuracy: 0.6048 - val_loss: 1.0857 -
val_categorical_crossentropy: 1.0857 - val_accuracy: 0.5772 - lr: 4.0000e-04
Epoch 48/150
428/428 [==============================] - 80s 188ms/step - loss: 1.0103 -
categorical_crossentropy: 1.0103 - accuracy: 0.6096 - val_loss: 1.0805 -
val_categorical_crossentropy: 1.0805 - val_accuracy: 0.5845 - lr: 4.0000e-04
Epoch 49/150
428/428 [==============================] - 80s 187ms/step - loss: 1.0075 -
categorical_crossentropy: 1.0075 - accuracy: 0.6148 - val_loss: 1.0729 -
val_categorical_crossentropy: 1.0729 - val_accuracy: 0.5810 - lr: 4.0000e-04
Epoch 50/150
428/428 [==============================] - 79s 184ms/step - loss: 1.0007 -
categorical_crossentropy: 1.0007 - accuracy: 0.6122 - val_loss: 1.0787 -
val_categorical_crossentropy: 1.0787 - val_accuracy: 0.5812 - lr: 4.0000e-04
Epoch 51/150
428/428 [==============================] - 79s 185ms/step - loss: 0.9969 -
categorical_crossentropy: 0.9969 - accuracy: 0.6161 - val_loss: 1.0908 -
val_categorical_crossentropy: 1.0908 - val_accuracy: 0.5772 - lr: 4.0000e-04
Epoch 52/150
428/428 [==============================] - 80s 186ms/step - loss: 1.0023 -
categorical_crossentropy: 1.0023 - accuracy: 0.6144 - val_loss: 1.0834 -
val_categorical_crossentropy: 1.0834 - val_accuracy: 0.5781 - lr: 4.0000e-04
```

```
Epoch 53/150
428/428 [==============================] - ETA: 0s - loss: 1.0083 -
categorical_crossentropy: 1.0083 - accuracy: 0.6130
Epoch 53: ReduceLROnPlateau reducing learning rate to 0.00016000000759959222.
428/428 [==============================] - 79s 185ms/step - loss: 1.0083 -
categorical_crossentropy: 1.0083 - accuracy: 0.6130 - val_loss: 1.0787 -
val_categorical_crossentropy: 1.0787 - val_accuracy: 0.5760 - lr: 4.0000e-04
Epoch 54/150
428/428 [==============================] - 80s 187ms/step - loss: 0.9656 -
categorical_crossentropy: 0.9656 - accuracy: 0.6308 - val_loss: 1.0729 -
val_categorical_crossentropy: 1.0729 - val_accuracy: 0.5876 - lr: 1.6000e-04
Epoch 55/150
428/428 [==============================] - 80s 187ms/step - loss: 0.9475 -
categorical_crossentropy: 0.9475 - accuracy: 0.6331 - val_loss: 1.0697 -
val_categorical_crossentropy: 1.0697 - val_accuracy: 0.5896 - lr: 1.6000e-04
Epoch 56/150
428/428 [==============================] - 80s 187ms/step - loss: 0.9371 -
categorical_crossentropy: 0.9371 - accuracy: 0.6379 - val_loss: 1.0663 -
val_categorical_crossentropy: 1.0663 - val_accuracy: 0.5922 - lr: 1.6000e-04
Epoch 57/150
428/428 [==============================] - 80s 188ms/step - loss: 0.9278 -
categorical_crossentropy: 0.9278 - accuracy: 0.6447 - val_loss: 1.0629 -
val_categorical_crossentropy: 1.0629 - val_accuracy: 0.5936 - lr: 1.6000e-04
Epoch 58/150
428/428 [==============================] - 80s 187ms/step - loss: 0.9216 -
categorical_crossentropy: 0.9216 - accuracy: 0.6465 - val_loss: 1.0751 -
val_categorical_crossentropy: 1.0751 - val_accuracy: 0.5914 - lr: 1.6000e-04
Epoch 59/150
428/428 [==============================] - 80s 188ms/step - loss: 0.9120 -
categorical_crossentropy: 0.9120 - accuracy: 0.6475 - val_loss: 1.0782 -
val_categorical_crossentropy: 1.0782 - val_accuracy: 0.5910 - lr: 1.6000e-04
Epoch 60/150
428/428 [==============================] - 81s 189ms/step - loss: 0.9072 -
categorical_crossentropy: 0.9072 - accuracy: 0.6511 - val_loss: 1.0731 -
val_categorical_crossentropy: 1.0731 - val_accuracy: 0.5960 - lr: 1.6000e-04
Epoch 61/150
428/428 [==============================] - 80s 186ms/step - loss: 0.9048 -
categorical_crossentropy: 0.9048 - accuracy: 0.6560 - val_loss: 1.0587 -
val_categorical_crossentropy: 1.0587 - val_accuracy: 0.5986 - lr: 1.6000e-04
Epoch 62/150
428/428 [==============================] - 80s 186ms/step - loss: 0.8954 -
categorical_crossentropy: 0.8954 - accuracy: 0.6573 - val_loss: 1.0628 -
val_categorical_crossentropy: 1.0628 - val_accuracy: 0.6022 - lr: 1.6000e-04
Epoch 63/150
428/428 [==============================] - 81s 190ms/step - loss: 0.8944 -
categorical_crossentropy: 0.8944 - accuracy: 0.6570 - val_loss: 1.0708 -
val_categorical_crossentropy: 1.0708 - val_accuracy: 0.5978 - lr: 1.6000e-04
Epoch 64/150
```

```
428/428 [==============================] - 80s 187ms/step - loss: 0.8830 -
categorical_crossentropy: 0.8830 - accuracy: 0.6612 - val_loss: 1.0861 -
val_categorical_crossentropy: 1.0861 - val_accuracy: 0.5970 - lr: 1.6000e-04
Epoch 65/150
428/428 [==============================] - 80s 187ms/step - loss: 0.8832 -
categorical_crossentropy: 0.8832 - accuracy: 0.6605 - val_loss: 1.0743 -
val_categorical_crossentropy: 1.0743 - val_accuracy: 0.5971 - lr: 1.6000e-04
Epoch 66/150
428/428 [==============================] - 80s 188ms/step - loss: 0.8704 -
categorical_crossentropy: 0.8704 - accuracy: 0.6652 - val_loss: 1.0797 -
val_categorical_crossentropy: 1.0797 - val_accuracy: 0.5983 - lr: 1.6000e-04
Epoch 67/150
428/428 [==============================] - 80s 188ms/step - loss: 0.8720 -
categorical_crossentropy: 0.8720 - accuracy: 0.6662 - val_loss: 1.0773 -
val_categorical_crossentropy: 1.0773 - val_accuracy: 0.5983 - lr: 1.6000e-04
Epoch 68/150
428/428 [==============================] - 80s 186ms/step - loss: 0.8667 -
categorical_crossentropy: 0.8667 - accuracy: 0.6701 - val_loss: 1.0747 -
val_categorical_crossentropy: 1.0747 - val_accuracy: 0.5980 - lr: 1.6000e-04
Epoch 69/150
428/428 [==============================] - 80s 186ms/step - loss: 0.8645 -
categorical_crossentropy: 0.8645 - accuracy: 0.6672 - val_loss: 1.0770 -
val_categorical_crossentropy: 1.0770 - val_accuracy: 0.6000 - lr: 1.6000e-04
Epoch 70/150
428/428 [==============================] - 79s 185ms/step - loss: 0.8553 -
categorical_crossentropy: 0.8553 - accuracy: 0.6729 - val_loss: 1.0797 -
val_categorical_crossentropy: 1.0797 - val_accuracy: 0.6055 - lr: 1.6000e-04
Epoch 71/150
428/428 [==============================] - 79s 186ms/step - loss: 0.8559 -
categorical_crossentropy: 0.8559 - accuracy: 0.6751 - val_loss: 1.0709 -
val_categorical_crossentropy: 1.0709 - val_accuracy: 0.6044 - lr: 1.6000e-04
Epoch 72/150
428/428 [==============================] - ETA: 0s - loss: 0.8554 -
categorical_crossentropy: 0.8554 - accuracy: 0.6732
Epoch 72: ReduceLROnPlateau reducing learning rate to 6.40000042039901e-05.
428/428 [==============================] - 81s 190ms/step - loss: 0.8554 -
categorical_crossentropy: 0.8554 - accuracy: 0.6732 - val_loss: 1.0633 -
val_categorical_crossentropy: 1.0633 - val_accuracy: 0.6028 - lr: 1.6000e-04
Epoch 73/150
428/428 [==============================] - 81s 188ms/step - loss: 0.8380 -
categorical_crossentropy: 0.8380 - accuracy: 0.6805 - val_loss: 1.0783 -
val_categorical_crossentropy: 1.0783 - val_accuracy: 0.6073 - lr: 6.4000e-05
Epoch 74/150
428/428 [==============================] - 81s 189ms/step - loss: 0.8351 -
categorical_crossentropy: 0.8351 - accuracy: 0.6811 - val_loss: 1.0778 -
val_categorical_crossentropy: 1.0778 - val_accuracy: 0.6047 - lr: 6.4000e-05
Epoch 75/150
428/428 [==============================] - 81s 189ms/step - loss: 0.8292 -
```

```
categorical_crossentropy: 0.8292 - accuracy: 0.6832 - val_loss: 1.0760 -
val_categorical_crossentropy: 1.0760 - val_accuracy: 0.6073 - lr: 6.4000e-05
Epoch 76/150
428/428 [==============================] - 81s 190ms/step - loss: 0.8146 -
categorical_crossentropy: 0.8146 - accuracy: 0.6898 - val_loss: 1.0980 -
val_categorical_crossentropy: 1.0980 - val_accuracy: 0.6064 - lr: 6.4000e-05
Epoch 77/150
428/428 [==============================] - 81s 189ms/step - loss: 0.8202 -
categorical_crossentropy: 0.8202 - accuracy: 0.6868 - val_loss: 1.0900 -
val_categorical_crossentropy: 1.0900 - val_accuracy: 0.6103 - lr: 6.4000e-05
Epoch 78/150
428/428 [==============================] - 81s 190ms/step - loss: 0.8070 -
categorical_crossentropy: 0.8070 - accuracy: 0.6887 - val_loss: 1.0954 -
val_categorical_crossentropy: 1.0954 - val_accuracy: 0.6081 - lr: 6.4000e-05
Epoch 79/150
428/428 [==============================] - 81s 190ms/step - loss: 0.8117 -
categorical_crossentropy: 0.8117 - accuracy: 0.6908 - val_loss: 1.0915 -
val_categorical_crossentropy: 1.0915 - val_accuracy: 0.6025 - lr: 6.4000e-05
Epoch 80/150
428/428 [==============================] - ETA: 0s - loss: 0.8098 -
categorical_crossentropy: 0.8098 - accuracy: 0.6917
Epoch 80: ReduceLROnPlateau reducing learning rate to 2.560000284574926e-05.
428/428 [==============================] - 81s 190ms/step - loss: 0.8098 -
categorical_crossentropy: 0.8098 - accuracy: 0.6917 - val_loss: 1.0902 -
val_categorical_crossentropy: 1.0902 - val_accuracy: 0.6068 - lr: 6.4000e-05
Epoch 81/150
428/428 [==============================] - 82s 191ms/step - loss: 0.8013 -
categorical_crossentropy: 0.8013 - accuracy: 0.6956 - val_loss: 1.0913 -
val_categorical_crossentropy: 1.0913 - val_accuracy: 0.6073 - lr: 2.5600e-05
Epoch 82/150
428/428 [==============================] - 82s 192ms/step - loss: 0.7950 -
categorical_crossentropy: 0.7950 - accuracy: 0.6954 - val_loss: 1.0967 -
val_categorical_crossentropy: 1.0967 - val_accuracy: 0.6046 - lr: 2.5600e-05
Epoch 83/150
428/428 [==============================] - 81s 190ms/step - loss: 0.7979 -
categorical_crossentropy: 0.7979 - accuracy: 0.6966 - val_loss: 1.0966 -
val_categorical_crossentropy: 1.0966 - val_accuracy: 0.6071 - lr: 2.5600e-05
Epoch 84/150
428/428 [==============================] - 81s 190ms/step - loss: 0.7924 -
categorical_crossentropy: 0.7924 - accuracy: 0.6966 - val_loss: 1.1053 -
val_categorical_crossentropy: 1.1053 - val_accuracy: 0.6093 - lr: 2.5600e-05
Epoch 85/150
428/428 [==============================] - 81s 190ms/step - loss: 0.7924 -
categorical_crossentropy: 0.7924 - accuracy: 0.6977 - val_loss: 1.1038 -
val_categorical_crossentropy: 1.1038 - val_accuracy: 0.6075 - lr: 2.5600e-05
Epoch 86/150
428/428 [==============================] - 87s 203ms/step - loss: 0.7882 -
categorical_crossentropy: 0.7882 - accuracy: 0.6985 - val_loss: 1.1041 -
```

```
val_categorical_crossentropy: 1.1041 - val_accuracy: 0.6106 - lr: 2.5600e-05
Epoch 87/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7864 -
categorical_crossentropy: 0.7864 - accuracy: 0.7004 - val_loss: 1.0974 -
val_categorical_crossentropy: 1.0974 - val_accuracy: 0.6084 - lr: 2.5600e-05
Epoch 88/150
428/428 [==============================] - 79s 186ms/step - loss: 0.7857 -
categorical_crossentropy: 0.7857 - accuracy: 0.7002 - val_loss: 1.1063 -
val_categorical_crossentropy: 1.1063 - val_accuracy: 0.6082 - lr: 2.5600e-05
Epoch 89/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7835 -
categorical_crossentropy: 0.7835 - accuracy: 0.7012 - val_loss: 1.1035 -
val_categorical_crossentropy: 1.1035 - val_accuracy: 0.6061 - lr: 2.5600e-05
Epoch 90/150
428/428 [==============================] - 81s 190ms/step - loss: 0.7855 -
categorical_crossentropy: 0.7855 - accuracy: 0.7017 - val_loss: 1.1017 -
val_categorical_crossentropy: 1.1017 - val_accuracy: 0.6085 - lr: 2.5600e-05
Epoch 91/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7833 -
categorical_crossentropy: 0.7833 - accuracy: 0.7014 - val_loss: 1.1070 -
val_categorical_crossentropy: 1.1070 - val_accuracy: 0.6104 - lr: 2.5600e-05
Epoch 92/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7798 -
categorical_crossentropy: 0.7798 - accuracy: 0.7030 - val_loss: 1.1057 -
val_categorical_crossentropy: 1.1057 - val_accuracy: 0.6106 - lr: 2.5600e-05
Epoch 93/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7825 -
categorical_crossentropy: 0.7825 - accuracy: 0.7024 - val_loss: 1.1001 -
val_categorical_crossentropy: 1.1001 - val_accuracy: 0.6108 - lr: 2.5600e-05
Epoch 94/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7747 -
categorical_crossentropy: 0.7747 - accuracy: 0.7044 - val_loss: 1.1043 -
val_categorical_crossentropy: 1.1043 - val_accuracy: 0.6095 - lr: 2.5600e-05
Epoch 95/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7805 -
categorical_crossentropy: 0.7805 - accuracy: 0.7043 - val_loss: 1.1096 -
val_categorical_crossentropy: 1.1096 - val_accuracy: 0.6085 - lr: 2.5600e-05
Epoch 96/150
428/428 [==============================] - ETA: 0s - loss: 0.7808 -
categorical_crossentropy: 0.7808 - accuracy: 0.7004
Epoch 96: ReduceLROnPlateau reducing learning rate to 1.0240000847261399e-05.
428/428 [==============================] - 80s 187ms/step - loss: 0.7808 -
categorical_crossentropy: 0.7808 - accuracy: 0.7004 - val_loss: 1.1043 -
val_categorical_crossentropy: 1.1043 - val_accuracy: 0.6116 - lr: 2.5600e-05
Epoch 97/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7749 -
categorical_crossentropy: 0.7749 - accuracy: 0.7045 - val_loss: 1.1047 -
val_categorical_crossentropy: 1.1047 - val_accuracy: 0.6115 - lr: 1.0240e-05
```

```
Epoch 98/150
428/428 [==============================] - ETA: 0s - loss: 0.7796 -
categorical_crossentropy: 0.7796 - accuracy: 0.7018
Epoch 98: ReduceLROnPlateau reducing learning rate to 4.09600033890456e-06.
428/428 [==============================] - 80s 187ms/step - loss: 0.7796 -
categorical_crossentropy: 0.7796 - accuracy: 0.7018 - val_loss: 1.1041 -
val_categorical_crossentropy: 1.1041 - val_accuracy: 0.6105 - lr: 1.0240e-05
Epoch 99/150
428/428 [==============================] - 79s 186ms/step - loss: 0.7634 -
categorical_crossentropy: 0.7634 - accuracy: 0.7079 - val_loss: 1.1045 -
val_categorical_crossentropy: 1.1045 - val_accuracy: 0.6108 - lr: 4.0960e-06
Epoch 100/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7694 -
categorical_crossentropy: 0.7694 - accuracy: 0.7061 - val_loss: 1.1047 -
val_categorical_crossentropy: 1.1047 - val_accuracy: 0.6103 - lr: 4.0960e-06
Epoch 101/150
428/428 [==============================] - ETA: 0s - loss: 0.7714 -
categorical_crossentropy: 0.7714 - accuracy: 0.7071
Epoch 101: ReduceLROnPlateau reducing learning rate to 1.6384001355618238e-06.
428/428 [==============================] - 81s 189ms/step - loss: 0.7714 -
categorical_crossentropy: 0.7714 - accuracy: 0.7071 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6107 - lr: 4.0960e-06
Epoch 102/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7718 -
categorical_crossentropy: 0.7718 - accuracy: 0.7055 - val_loss: 1.1044 -
val_categorical_crossentropy: 1.1044 - val_accuracy: 0.6111 - lr: 1.6384e-06
Epoch 103/150
428/428 [==============================] - ETA: 0s - loss: 0.7722 -
categorical_crossentropy: 0.7722 - accuracy: 0.7070
Epoch 103: ReduceLROnPlateau reducing learning rate to 6.553600542247295e-07.
428/428 [==============================] - 79s 185ms/step - loss: 0.7722 -
categorical_crossentropy: 0.7722 - accuracy: 0.7070 - val_loss: 1.1046 -
val_categorical_crossentropy: 1.1046 - val_accuracy: 0.6109 - lr: 1.6384e-06
Epoch 104/150
428/428 [==============================] - 79s 186ms/step - loss: 0.7677 -
categorical_crossentropy: 0.7677 - accuracy: 0.7074 - val_loss: 1.1047 -
val_categorical_crossentropy: 1.1047 - val_accuracy: 0.6108 - lr: 6.5536e-07
Epoch 105/150
428/428 [==============================] - ETA: 0s - loss: 0.7654 -
categorical_crossentropy: 0.7654 - accuracy: 0.7076
Epoch 105: ReduceLROnPlateau reducing learning rate to 2.6214402168989184e-07.
428/428 [==============================] - 80s 187ms/step - loss: 0.7654 -
categorical_crossentropy: 0.7654 - accuracy: 0.7076 - val_loss: 1.1051 -
val_categorical_crossentropy: 1.1051 - val_accuracy: 0.6107 - lr: 6.5536e-07
Epoch 106/150
428/428 [==============================] - 80s 188ms/step - loss: 0.7726 -
categorical_crossentropy: 0.7726 - accuracy: 0.7036 - val_loss: 1.1051 -
val_categorical_crossentropy: 1.1051 - val_accuracy: 0.6102 - lr: 2.6214e-07
```

```
Epoch 107/150
428/428 [==============================] - ETA: 0s - loss: 0.7702 -
categorical_crossentropy: 0.7702 - accuracy: 0.7076
Epoch 107: ReduceLROnPlateau reducing learning rate to 1.0485761094969349e-07.
428/428 [==============================] - 80s 187ms/step - loss: 0.7702 -
categorical_crossentropy: 0.7702 - accuracy: 0.7076 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.6214e-07
Epoch 108/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7758 -
categorical_crossentropy: 0.7758 - accuracy: 0.7040 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6103 - lr: 1.0486e-07
Epoch 109/150
428/428 [==============================] - ETA: 0s - loss: 0.7679 -
categorical_crossentropy: 0.7679 - accuracy: 0.7073
Epoch 109: ReduceLROnPlateau reducing learning rate to 4.1943044948311586e-08.
428/428 [==============================] - 80s 188ms/step - loss: 0.7679 -
categorical_crossentropy: 0.7679 - accuracy: 0.7073 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0486e-07
Epoch 110/150
428/428 [==============================] - 81s 189ms/step - loss: 0.7742 -
categorical_crossentropy: 0.7742 - accuracy: 0.7081 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 4.1943e-08
Epoch 111/150
428/428 [==============================] - ETA: 0s - loss: 0.7742 -
categorical_crossentropy: 0.7742 - accuracy: 0.7053
Epoch 111: ReduceLROnPlateau reducing learning rate to 1.677721854775882e-08.
428/428 [==============================] - 81s 190ms/step - loss: 0.7742 -
categorical_crossentropy: 0.7742 - accuracy: 0.7053 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 4.1943e-08
Epoch 112/150
428/428 [==============================] - 82s 191ms/step - loss: 0.7654 -
categorical_crossentropy: 0.7654 - accuracy: 0.7085 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.6777e-08
Epoch 113/150
428/428 [==============================] - ETA: 0s - loss: 0.7700 -
categorical_crossentropy: 0.7700 - accuracy: 0.7045
Epoch 113: ReduceLROnPlateau reducing learning rate to 6.710887134886434e-09.
428/428 [==============================] - 80s 187ms/step - loss: 0.7700 -
categorical_crossentropy: 0.7700 - accuracy: 0.7045 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.6777e-08
Epoch 114/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7808 -
categorical_crossentropy: 0.7808 - accuracy: 0.7049 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 6.7109e-09
Epoch 115/150
428/428 [==============================] - ETA: 0s - loss: 0.7744 -
categorical_crossentropy: 0.7744 - accuracy: 0.7036
Epoch 115: ReduceLROnPlateau reducing learning rate to 2.6843547829003003e-09.
```

```
428/428 [==============================] - 80s 187ms/step - loss: 0.7744 -
categorical_crossentropy: 0.7744 - accuracy: 0.7036 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 6.7109e-09
Epoch 116/150
428/428 [==============================] - 80s 188ms/step - loss: 0.7624 -
categorical_crossentropy: 0.7624 - accuracy: 0.7092 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.6844e-09
Epoch 117/150
428/428 [==============================] - 81s 188ms/step - loss: 0.7704 -
categorical_crossentropy: 0.7704 - accuracy: 0.7055 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.6844e-09
Epoch 118/150
428/428 [==============================] - ETA: 0s - loss: 0.7690 -
categorical_crossentropy: 0.7690 - accuracy: 0.7074
Epoch 118: ReduceLROnPlateau reducing learning rate to 1.0737418953965518e-09.
428/428 [==============================] - 80s 188ms/step - loss: 0.7690 -
categorical_crossentropy: 0.7690 - accuracy: 0.7074 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.6844e-09
Epoch 119/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7702 -
categorical_crossentropy: 0.7702 - accuracy: 0.7082 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0737e-09
Epoch 120/150
428/428 [==============================] - ETA: 0s - loss: 0.7745 -
categorical_crossentropy: 0.7745 - accuracy: 0.7012
Epoch 120: ReduceLROnPlateau reducing learning rate to 4.294967492768365e-10.
428/428 [==============================] - 81s 188ms/step - loss: 0.7745 -
categorical_crossentropy: 0.7745 - accuracy: 0.7012 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0737e-09
Epoch 121/150
428/428 [==============================] - 79s 185ms/step - loss: 0.7712 -
categorical_crossentropy: 0.7712 - accuracy: 0.7082 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 4.2950e-10
Epoch 122/150
428/428 [==============================] - ETA: 0s - loss: 0.7745 -
categorical_crossentropy: 0.7745 - accuracy: 0.7048
Epoch 122: ReduceLROnPlateau reducing learning rate to 1.7179869749028854e-10.
428/428 [==============================] - 79s 186ms/step - loss: 0.7745 -
categorical_crossentropy: 0.7745 - accuracy: 0.7048 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 4.2950e-10
Epoch 123/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7732 -
categorical_crossentropy: 0.7732 - accuracy: 0.7067 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.7180e-10
Epoch 124/150
428/428 [==============================] - ETA: 0s - loss: 0.7693 -
categorical_crossentropy: 0.7693 - accuracy: 0.7053
Epoch 124: ReduceLROnPlateau reducing learning rate to 6.871948010633844e-11.
```

```
428/428 [==============================] - 80s 187ms/step - loss: 0.7693 -
categorical_crossentropy: 0.7693 - accuracy: 0.7053 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.7180e-10
Epoch 125/150
428/428 [==============================] - 81s 188ms/step - loss: 0.7730 -
categorical_crossentropy: 0.7730 - accuracy: 0.7032 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 6.8719e-11
Epoch 126/150
428/428 [==============================] - ETA: 0s - loss: 0.7765 -
categorical_crossentropy: 0.7765 - accuracy: 0.7031
Epoch 126: ReduceLROnPlateau reducing learning rate to 2.748779259764689e-11.
428/428 [==============================] - 81s 188ms/step - loss: 0.7765 -
categorical_crossentropy: 0.7765 - accuracy: 0.7031 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 6.8719e-11
Epoch 127/150
428/428 [==============================] - 81s 189ms/step - loss: 0.7682 -
categorical_crossentropy: 0.7682 - accuracy: 0.7074 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.7488e-11
Epoch 128/150
428/428 [==============================] - ETA: 0s - loss: 0.7671 -
categorical_crossentropy: 0.7671 - accuracy: 0.7079
Epoch 128: ReduceLROnPlateau reducing learning rate to 1.0995117316614512e-11.
428/428 [==============================] - 80s 188ms/step - loss: 0.7671 -
categorical_crossentropy: 0.7671 - accuracy: 0.7079 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 2.7488e-11
Epoch 129/150
428/428 [==============================] - 80s 188ms/step - loss: 0.7741 -
categorical_crossentropy: 0.7741 - accuracy: 0.7052 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0995e-11
Epoch 130/150
428/428 [==============================] - ETA: 0s - loss: 0.7727 -
categorical_crossentropy: 0.7727 - accuracy: 0.7026
Epoch 130: ReduceLROnPlateau reducing learning rate to 1e-11.
428/428 [==============================] - 80s 188ms/step - loss: 0.7727 -
categorical_crossentropy: 0.7727 - accuracy: 0.7026 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0995e-11
Epoch 131/150
428/428 [==============================] - 81s 189ms/step - loss: 0.7740 -
categorical_crossentropy: 0.7740 - accuracy: 0.7056 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 132/150
428/428 [==============================] - 79s 185ms/step - loss: 0.7747 -
categorical_crossentropy: 0.7747 - accuracy: 0.7055 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 133/150
428/428 [==============================] - 79s 185ms/step - loss: 0.7754 -
categorical_crossentropy: 0.7754 - accuracy: 0.7051 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
```

```
Epoch 134/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7717 -
categorical_crossentropy: 0.7717 - accuracy: 0.7078 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 135/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7687 -
categorical_crossentropy: 0.7687 - accuracy: 0.7040 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 136/150
428/428 [==============================] - 79s 185ms/step - loss: 0.7720 -
categorical_crossentropy: 0.7720 - accuracy: 0.7071 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 137/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7704 -
categorical_crossentropy: 0.7704 - accuracy: 0.7072 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 138/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7678 -
categorical_crossentropy: 0.7678 - accuracy: 0.7067 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 139/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7700 -
categorical_crossentropy: 0.7700 - accuracy: 0.7063 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 140/150
428/428 [==============================] - 81s 188ms/step - loss: 0.7690 -
categorical_crossentropy: 0.7690 - accuracy: 0.7070 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 141/150
428/428 [==============================] - 81s 188ms/step - loss: 0.7719 -
categorical_crossentropy: 0.7719 - accuracy: 0.7068 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 142/150
428/428 [==============================] - 80s 188ms/step - loss: 0.7732 -
categorical_crossentropy: 0.7732 - accuracy: 0.7043 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 143/150
428/428 [==============================] - 80s 187ms/step - loss: 0.7691 -
categorical_crossentropy: 0.7691 - accuracy: 0.7038 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 144/150
428/428 [==============================] - 79s 186ms/step - loss: 0.7734 -
categorical_crossentropy: 0.7734 - accuracy: 0.7049 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 145/150
428/428 [==============================] - 79s 185ms/step - loss: 0.7756 -
categorical_crossentropy: 0.7756 - accuracy: 0.7021 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
```

```
Epoch 146/150
428/428 [==============================] - 79s 186ms/step - loss: 0.7703 -
categorical_crossentropy: 0.7703 - accuracy: 0.7040 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 147/150
428/428 [==============================] - 80s 186ms/step - loss: 0.7756 -
categorical_crossentropy: 0.7756 - accuracy: 0.7068 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 148/150
428/428 [==============================] - 80s 188ms/step - loss: 0.7720 -
categorical_crossentropy: 0.7720 - accuracy: 0.7042 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 149/150
428/428 [==============================] - 84s 197ms/step - loss: 0.7765 -
categorical_crossentropy: 0.7765 - accuracy: 0.7031 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
Epoch 150/150
428/428 [==============================] - 84s 196ms/step - loss: 0.7684 -
categorical_crossentropy: 0.7684 - accuracy: 0.7068 - val_loss: 1.1052 -
val_categorical_crossentropy: 1.1052 - val_accuracy: 0.6102 - lr: 1.0000e-11
```

[11]:
```
!mkdir saved_model_Long_CNN
model.save('saved_model_Long_CNN/ser_LongCNN_model')
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op,
_jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing
5 of 6). These functions will not be directly callable after loading.
```

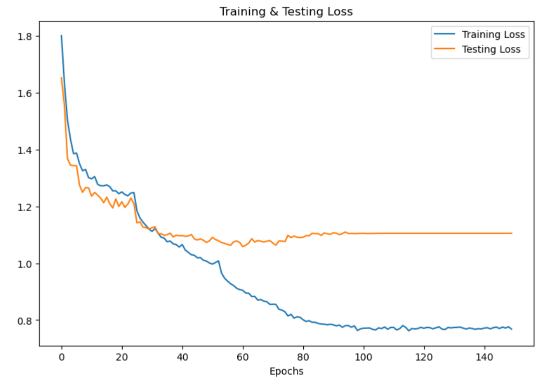
```
INFO:tensorflow:Assets written to: saved_model_Long_CNN/ser_LongCNN_model\assets
```

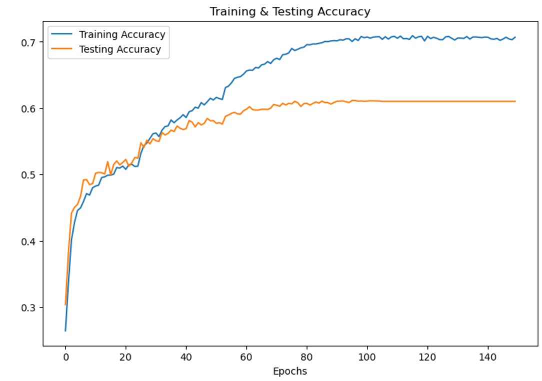

```
INFO:tensorflow:Assets written to: saved_model_Long_CNN/ser_LongCNN_model\assets
```


[13]:
```
print("Accuracy of our model on test data : " , model.
 ↪evaluate(X_test,y_test)[1]*100 , "%")



fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
test_acc = history.history['val_accuracy']
test_loss = history.history['val_loss']
epochs = [i for i in range(len(train_loss))]
# epochs = range(50)
print(epochs)

fig.set_size_inches(20,6)
ax[0].plot(epochs , train_loss , label = 'Training Loss')
ax[0].plot(epochs , test_loss , label = 'Testing Loss')
```

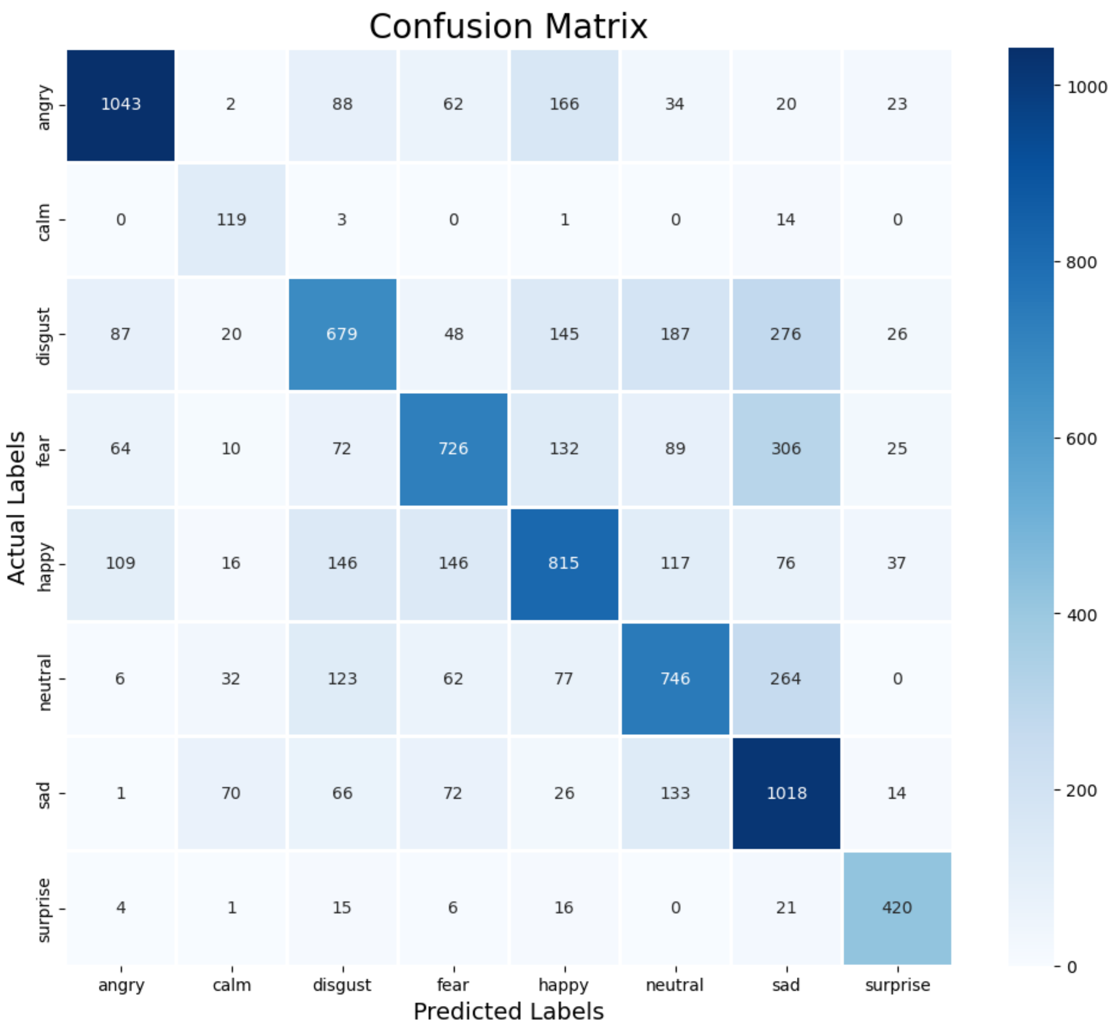
```
ax[0].set_title('Training & Testing Loss')
ax[0].legend()
ax[0].set_xlabel("Epochs")

ax[1].plot(epochs , train_acc , label = 'Training Accuracy')
ax[1].plot(epochs , test_acc , label = 'Testing Accuracy')
ax[1].set_title('Training & Testing Accuracy')
ax[1].legend()
ax[1].set_xlabel("Epochs")
plt.show()

# predicting on test data.
pred_test = model.predict(X_test)
print(pred_test)

y_pred = encoder.inverse_transform(pred_test)
print(y_pred)
print(y_pred.shape)
y_test = encoder.inverse_transform(y_test)
df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()
df.head(100)
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] , columns = [i␣
↪for i in encoder.categories_])
sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, annot=True,␣
↪fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()
print(classification_report(y_test, y_pred))
```

```
286/286 [==============================] - 6s 19ms/step - loss: 1.1052 -
categorical_crossentropy: 1.1052 - accuracy: 0.6102
Accuracy of our model on test data :  110.52205562591553 %
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148,
149]
```






```
286/286 [==============================] - 5s 19ms/step
[[1.07656240e-06 5.48158168e-19 1.04791433e-01 … 1.82567257e-02
  7.01780379e-01 9.02292102e-20]
 [2.81857697e-11 0.00000000e+00 1.00000000e+00 … 2.57464883e-11
  3.87721834e-11 2.48926856e-12]
 [1.00000000e+00 0.00000000e+00 0.00000000e+00 … 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 …
 [1.20629795e-01 3.77792686e-14 1.22446589e-01 … 9.00925603e-03
  4.58990922e-04 6.78196386e-07]
 [8.30115750e-02 3.17032728e-12 1.17087021e-01 … 4.62225974e-02
  3.31930071e-03 3.19312198e-06]
 [0.00000000e+00 0.00000000e+00 5.94362444e-28 … 6.74500218e-38
  1.00000000e+00 0.00000000e+00]]
[['sad']
 ['disgust']
 ['angry']
 …
 ['happy']
 ['happy']
 ['sad']]
(9122, 1)
```



	precision	recall	f1-score	support
angry	0.79	0.73	0.76	1438
calm	0.44	0.87	0.58	137
disgust	0.57	0.46	0.51	1468
fear	0.65	0.51	0.57	1424
happy	0.59	0.56	0.57	1462
neutral	0.57	0.57	0.57	1310
sad	0.51	0.73	0.60	1400
surprise	0.77	0.87	0.82	483
accuracy			0.61	9122
macro avg	0.61	0.66	0.62	9122
weighted avg	0.62	0.61	0.61	9122

[9]:
```
model=Sequential()
#
model.add(LSTM(units=256, activation='tanh',  input_shape=(X_train.shape[1],␣
 ↪1), return_sequences=True))
model.add(keras.layers.Dropout(rate=0.3))
#
model.add(LSTM(units=128, activation='tanh', return_sequences=True))
model.add(keras.layers.Dropout(rate=0.3))

model.add(LSTM(units=64, activation='tanh'))
model.add(keras.layers.Dropout(rate=0.3))

model.add(Flatten())
model.add(Dense(units=128, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=64, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=32, activation='relu'))
model.add(Dropout(0.3))

model.add(Dense(units=8, activation='softmax'))
model.compile(optimizer = tf.keras.optimizers.Adam(learning_rate=1e-4) , loss =␣
 ↪'categorical_crossentropy' ,
              metrics = ['accuracy'])

model.summary()
```

```
Model: "sequential_1"

_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 170, 256)          264192

 dropout_8 (Dropout)         (None, 170, 256)          0

 lstm_1 (LSTM)               (None, 170, 128)          197120

 dropout_9 (Dropout)         (None, 170, 128)          0

 lstm_2 (LSTM)               (None, 64)                49408

 dropout_10 (Dropout)        (None, 64)                0

 flatten_1 (Flatten)         (None, 64)                0

 dense_7 (Dense)             (None, 128)               8320
```

```
 dropout_11 (Dropout)        (None, 128)               0

 dense_8 (Dense)             (None, 64)                8256

 dropout_12 (Dropout)        (None, 64)                0

 dense_9 (Dense)             (None, 32)                2080

 dropout_13 (Dropout)        (None, 32)                0

 dense_10 (Dense)            (None, 8)                 264

=================================================================
Total params: 529,640
Trainable params: 529,640
Non-trainable params: 0
_________________________________________________________________
```

[11]:

```python
rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=1, patience=2,␣
 ↪min_lr=0.0000000001)
history=model.fit(X_train, y_train, batch_size=64, epochs=150,␣
 ↪validation_data=(X_test, y_test), callbacks=[rlrp])
```

```
Epoch 1/150
428/428 [==============================] - 242s 566ms/step - loss: 1.9645 -
accuracy: 0.2155 - val_loss: 1.8618 - val_accuracy: 0.2574 - lr: 1.0000e-04
Epoch 2/150
428/428 [==============================] - 231s 539ms/step - loss: 1.8696 -
accuracy: 0.2419 - val_loss: 1.8187 - val_accuracy: 0.2718 - lr: 1.0000e-04
Epoch 3/150
428/428 [==============================] - 241s 563ms/step - loss: 1.8346 -
accuracy: 0.2543 - val_loss: 1.7672 - val_accuracy: 0.2827 - lr: 1.0000e-04
Epoch 4/150
428/428 [==============================] - 247s 578ms/step - loss: 1.8166 -
accuracy: 0.2592 - val_loss: 1.7610 - val_accuracy: 0.2834 - lr: 1.0000e-04
Epoch 5/150
428/428 [==============================] - 232s 543ms/step - loss: 1.7996 -
accuracy: 0.2682 - val_loss: 1.7610 - val_accuracy: 0.2931 - lr: 1.0000e-04
Epoch 6/150
428/428 [==============================] - 235s 550ms/step - loss: 1.7829 -
accuracy: 0.2781 - val_loss: 1.7465 - val_accuracy: 0.2930 - lr: 1.0000e-04
Epoch 7/150
428/428 [==============================] - 234s 546ms/step - loss: 1.7763 -
accuracy: 0.2782 - val_loss: 1.7237 - val_accuracy: 0.2996 - lr: 1.0000e-04
Epoch 8/150
428/428 [==============================] - 236s 551ms/step - loss: 1.7669 -
accuracy: 0.2834 - val_loss: 1.7185 - val_accuracy: 0.3052 - lr: 1.0000e-04
```

```
Epoch 9/150
428/428 [==============================] - 232s 543ms/step - loss: 1.7613 -
accuracy: 0.2870 - val_loss: 1.7218 - val_accuracy: 0.2960 - lr: 1.0000e-04
Epoch 10/150
428/428 [==============================] - 233s 544ms/step - loss: 1.7466 -
accuracy: 0.2927 - val_loss: 1.7158 - val_accuracy: 0.3003 - lr: 1.0000e-04
Epoch 11/150
428/428 [==============================] - 237s 554ms/step - loss: 1.7341 -
accuracy: 0.2963 - val_loss: 1.7038 - val_accuracy: 0.3065 - lr: 1.0000e-04
Epoch 12/150
428/428 [==============================] - 242s 566ms/step - loss: 1.7158 -
accuracy: 0.3035 - val_loss: 1.6792 - val_accuracy: 0.3190 - lr: 1.0000e-04
Epoch 13/150
428/428 [==============================] - 232s 543ms/step - loss: 1.7053 -
accuracy: 0.3146 - val_loss: 1.6572 - val_accuracy: 0.3239 - lr: 1.0000e-04
Epoch 14/150
428/428 [==============================] - 238s 557ms/step - loss: 1.6954 -
accuracy: 0.3168 - val_loss: 1.6446 - val_accuracy: 0.3364 - lr: 1.0000e-04
Epoch 15/150
428/428 [==============================] - 232s 543ms/step - loss: 1.6828 -
accuracy: 0.3241 - val_loss: 1.6346 - val_accuracy: 0.3408 - lr: 1.0000e-04
Epoch 16/150
428/428 [==============================] - 244s 571ms/step - loss: 1.6730 -
accuracy: 0.3331 - val_loss: 1.6435 - val_accuracy: 0.3410 - lr: 1.0000e-04
Epoch 17/150
428/428 [==============================] - 237s 554ms/step - loss: 1.6616 -
accuracy: 0.3335 - val_loss: 1.6485 - val_accuracy: 0.3508 - lr: 1.0000e-04
Epoch 18/150
428/428 [==============================] - 242s 564ms/step - loss: 1.6546 -
accuracy: 0.3400 - val_loss: 1.6197 - val_accuracy: 0.3379 - lr: 1.0000e-04
Epoch 19/150
428/428 [==============================] - 252s 590ms/step - loss: 1.6376 -
accuracy: 0.3438 - val_loss: 1.5917 - val_accuracy: 0.3508 - lr: 1.0000e-04
Epoch 20/150
428/428 [==============================] - 255s 595ms/step - loss: 1.6294 -
accuracy: 0.3478 - val_loss: 1.5869 - val_accuracy: 0.3664 - lr: 1.0000e-04
Epoch 21/150
428/428 [==============================] - 252s 589ms/step - loss: 1.6172 -
accuracy: 0.3520 - val_loss: 1.5932 - val_accuracy: 0.3456 - lr: 1.0000e-04
Epoch 22/150
428/428 [==============================] - 255s 597ms/step - loss: 1.6049 -
accuracy: 0.3522 - val_loss: 1.5609 - val_accuracy: 0.3687 - lr: 1.0000e-04
Epoch 23/150
428/428 [==============================] - 244s 571ms/step - loss: 1.5955 -
accuracy: 0.3592 - val_loss: 1.5430 - val_accuracy: 0.3667 - lr: 1.0000e-04
Epoch 24/150
428/428 [==============================] - 252s 590ms/step - loss: 1.5831 -
accuracy: 0.3615 - val_loss: 1.5236 - val_accuracy: 0.3809 - lr: 1.0000e-04
```

```
Epoch 25/150
428/428 [==============================] - 258s 604ms/step - loss: 1.5766 -
accuracy: 0.3654 - val_loss: 1.5321 - val_accuracy: 0.3756 - lr: 1.0000e-04
Epoch 26/150
428/428 [==============================] - 262s 612ms/step - loss: 1.5688 -
accuracy: 0.3697 - val_loss: 1.5182 - val_accuracy: 0.3824 - lr: 1.0000e-04
Epoch 27/150
428/428 [==============================] - 243s 568ms/step - loss: 1.5664 -
accuracy: 0.3656 - val_loss: 1.5184 - val_accuracy: 0.3721 - lr: 1.0000e-04
Epoch 28/150
428/428 [==============================] - 253s 592ms/step - loss: 1.5460 -
accuracy: 0.3753 - val_loss: 1.4994 - val_accuracy: 0.3879 - lr: 1.0000e-04
Epoch 29/150
428/428 [==============================] - 256s 599ms/step - loss: 1.5420 -
accuracy: 0.3768 - val_loss: 1.5250 - val_accuracy: 0.3917 - lr: 1.0000e-04
Epoch 30/150
428/428 [==============================] - 258s 603ms/step - loss: 1.5423 -
accuracy: 0.3789 - val_loss: 1.4724 - val_accuracy: 0.4021 - lr: 1.0000e-04
Epoch 31/150
428/428 [==============================] - 290s 679ms/step - loss: 1.5358 -
accuracy: 0.3785 - val_loss: 1.4724 - val_accuracy: 0.4043 - lr: 1.0000e-04
Epoch 32/150
428/428 [==============================] - 244s 569ms/step - loss: 1.5271 -
accuracy: 0.3784 - val_loss: 1.4687 - val_accuracy: 0.4121 - lr: 1.0000e-04
Epoch 33/150
428/428 [==============================] - 232s 543ms/step - loss: 1.5221 -
accuracy: 0.3873 - val_loss: 1.4705 - val_accuracy: 0.4067 - lr: 1.0000e-04
Epoch 34/150
428/428 [==============================] - 230s 537ms/step - loss: 1.5104 -
accuracy: 0.3867 - val_loss: 1.4588 - val_accuracy: 0.4221 - lr: 1.0000e-04
Epoch 35/150
428/428 [==============================] - 232s 542ms/step - loss: 1.5106 -
accuracy: 0.3884 - val_loss: 1.4491 - val_accuracy: 0.4172 - lr: 1.0000e-04
Epoch 36/150
428/428 [==============================] - 231s 540ms/step - loss: 1.4987 -
accuracy: 0.3944 - val_loss: 1.4401 - val_accuracy: 0.4171 - lr: 1.0000e-04
Epoch 37/150
428/428 [==============================] - 231s 539ms/step - loss: 1.4895 -
accuracy: 0.4008 - val_loss: 1.4617 - val_accuracy: 0.4255 - lr: 1.0000e-04
Epoch 38/150
428/428 [==============================] - 233s 543ms/step - loss: 1.4850 -
accuracy: 0.4041 - val_loss: 1.4508 - val_accuracy: 0.4019 - lr: 1.0000e-04
Epoch 39/150
428/428 [==============================] - 230s 538ms/step - loss: 1.4875 -
accuracy: 0.4015 - val_loss: 1.4238 - val_accuracy: 0.4251 - lr: 1.0000e-04
Epoch 40/150
428/428 [==============================] - 232s 541ms/step - loss: 1.4760 -
accuracy: 0.4100 - val_loss: 1.4114 - val_accuracy: 0.4365 - lr: 1.0000e-04
```

```
Epoch 41/150
428/428 [==============================] - 232s 542ms/step - loss: 1.4683 -
accuracy: 0.4127 - val_loss: 1.4141 - val_accuracy: 0.4287 - lr: 1.0000e-04
Epoch 42/150
428/428 [==============================] - 234s 548ms/step - loss: 1.4669 -
accuracy: 0.4132 - val_loss: 1.4066 - val_accuracy: 0.4456 - lr: 1.0000e-04
Epoch 43/150
428/428 [==============================] - 239s 560ms/step - loss: 1.4711 -
accuracy: 0.4124 - val_loss: 1.4147 - val_accuracy: 0.4264 - lr: 1.0000e-04
Epoch 44/150
428/428 [==============================] - 244s 569ms/step - loss: 1.4583 -
accuracy: 0.4164 - val_loss: 1.3919 - val_accuracy: 0.4484 - lr: 1.0000e-04
Epoch 45/150
428/428 [==============================] - 230s 537ms/step - loss: 1.4510 -
accuracy: 0.4217 - val_loss: 1.3956 - val_accuracy: 0.4399 - lr: 1.0000e-04
Epoch 46/150
428/428 [==============================] - 231s 539ms/step - loss: 1.4491 -
accuracy: 0.4211 - val_loss: 1.4303 - val_accuracy: 0.4229 - lr: 1.0000e-04
Epoch 47/150
428/428 [==============================] - 235s 548ms/step - loss: 1.4427 -
accuracy: 0.4237 - val_loss: 1.3988 - val_accuracy: 0.4409 - lr: 1.0000e-04
Epoch 48/150
428/428 [==============================] - 230s 538ms/step - loss: 1.4454 -
accuracy: 0.4269 - val_loss: 1.3953 - val_accuracy: 0.4469 - lr: 1.0000e-04
Epoch 49/150
428/428 [==============================] - 231s 540ms/step - loss: 1.4334 -
accuracy: 0.4302 - val_loss: 1.4074 - val_accuracy: 0.4372 - lr: 1.0000e-04
Epoch 50/150
428/428 [==============================] - 230s 538ms/step - loss: 1.4424 -
accuracy: 0.4266 - val_loss: 1.3908 - val_accuracy: 0.4426 - lr: 1.0000e-04
Epoch 51/150
428/428 [==============================] - 231s 540ms/step - loss: 1.4257 -
accuracy: 0.4378 - val_loss: 1.3864 - val_accuracy: 0.4544 - lr: 1.0000e-04
Epoch 52/150
428/428 [==============================] - 235s 548ms/step - loss: 1.4259 -
accuracy: 0.4363 - val_loss: 1.3885 - val_accuracy: 0.4494 - lr: 1.0000e-04
Epoch 53/150
428/428 [==============================] - 231s 541ms/step - loss: 1.4185 -
accuracy: 0.4382 - val_loss: 1.3551 - val_accuracy: 0.4613 - lr: 1.0000e-04
Epoch 54/150
428/428 [==============================] - 231s 541ms/step - loss: 1.4169 -
accuracy: 0.4441 - val_loss: 1.3667 - val_accuracy: 0.4536 - lr: 1.0000e-04
Epoch 55/150
428/428 [==============================] - 232s 542ms/step - loss: 1.4089 -
accuracy: 0.4442 - val_loss: 1.3559 - val_accuracy: 0.4654 - lr: 1.0000e-04
Epoch 56/150
428/428 [==============================] - 234s 546ms/step - loss: 1.4109 -
accuracy: 0.4442 - val_loss: 1.3650 - val_accuracy: 0.4552 - lr: 1.0000e-04
```

```
Epoch 57/150
428/428 [==============================] - 232s 542ms/step - loss: 1.4009 -
accuracy: 0.4454 - val_loss: 1.3461 - val_accuracy: 0.4605 - lr: 1.0000e-04
Epoch 58/150
428/428 [==============================] - 233s 545ms/step - loss: 1.3960 -
accuracy: 0.4521 - val_loss: 1.3403 - val_accuracy: 0.4670 - lr: 1.0000e-04
Epoch 59/150
428/428 [==============================] - 234s 547ms/step - loss: 1.3966 -
accuracy: 0.4483 - val_loss: 1.3531 - val_accuracy: 0.4608 - lr: 1.0000e-04
Epoch 60/150
428/428 [==============================] - 241s 563ms/step - loss: 1.3901 -
accuracy: 0.4495 - val_loss: 1.3913 - val_accuracy: 0.4497 - lr: 1.0000e-04
Epoch 61/150
428/428 [==============================] - 235s 549ms/step - loss: 1.3904 -
accuracy: 0.4531 - val_loss: 1.3484 - val_accuracy: 0.4620 - lr: 1.0000e-04
Epoch 62/150
428/428 [==============================] - 233s 543ms/step - loss: 1.3850 -
accuracy: 0.4514 - val_loss: 1.3324 - val_accuracy: 0.4754 - lr: 1.0000e-04
Epoch 63/150
428/428 [==============================] - 235s 549ms/step - loss: 1.3822 -
accuracy: 0.4537 - val_loss: 1.3218 - val_accuracy: 0.4783 - lr: 1.0000e-04
Epoch 64/150
428/428 [==============================] - 235s 549ms/step - loss: 1.3847 -
accuracy: 0.4559 - val_loss: 1.3492 - val_accuracy: 0.4639 - lr: 1.0000e-04
Epoch 65/150
428/428 [==============================] - 237s 554ms/step - loss: 1.3815 -
accuracy: 0.4527 - val_loss: 1.3686 - val_accuracy: 0.4522 - lr: 1.0000e-04
Epoch 66/150
428/428 [==============================] - 237s 553ms/step - loss: 1.3801 -
accuracy: 0.4536 - val_loss: 1.3405 - val_accuracy: 0.4617 - lr: 1.0000e-04
Epoch 67/150
428/428 [==============================] - 237s 554ms/step - loss: 1.3643 -
accuracy: 0.4604 - val_loss: 1.3335 - val_accuracy: 0.4655 - lr: 1.0000e-04
Epoch 68/150
428/428 [==============================] - 233s 545ms/step - loss: 1.3701 -
accuracy: 0.4591 - val_loss: 1.3155 - val_accuracy: 0.4717 - lr: 1.0000e-04
Epoch 69/150
428/428 [==============================] - ETA: 0s - loss: 1.4135 - accuracy:
0.4399
Epoch 69: ReduceLROnPlateau reducing learning rate to 3.9999998989515007e-05.
428/428 [==============================] - 235s 548ms/step - loss: 1.4135 -
accuracy: 0.4399 - val_loss: 1.3498 - val_accuracy: 0.4568 - lr: 1.0000e-04
Epoch 70/150
428/428 [==============================] - 234s 546ms/step - loss: 1.3695 -
accuracy: 0.4555 - val_loss: 1.3223 - val_accuracy: 0.4727 - lr: 4.0000e-05
Epoch 71/150
428/428 [==============================] - 234s 548ms/step - loss: 1.3509 -
accuracy: 0.4667 - val_loss: 1.3109 - val_accuracy: 0.4766 - lr: 4.0000e-05
```

```
Epoch 72/150
428/428 [==============================] - 232s 542ms/step - loss: 1.3410 -
accuracy: 0.4695 - val_loss: 1.3070 - val_accuracy: 0.4764 - lr: 4.0000e-05
Epoch 73/150
428/428 [==============================] - 232s 543ms/step - loss: 1.3399 -
accuracy: 0.4678 - val_loss: 1.3021 - val_accuracy: 0.4795 - lr: 4.0000e-05
Epoch 74/150
428/428 [==============================] - 230s 538ms/step - loss: 1.3340 -
accuracy: 0.4714 - val_loss: 1.2947 - val_accuracy: 0.4841 - lr: 4.0000e-05
Epoch 75/150
428/428 [==============================] - 239s 558ms/step - loss: 1.3274 -
accuracy: 0.4763 - val_loss: 1.2981 - val_accuracy: 0.4783 - lr: 4.0000e-05
Epoch 76/150
428/428 [==============================] - 234s 548ms/step - loss: 1.3275 -
accuracy: 0.4738 - val_loss: 1.2921 - val_accuracy: 0.4821 - lr: 4.0000e-05
Epoch 77/150
428/428 [==============================] - 231s 539ms/step - loss: 1.3254 -
accuracy: 0.4754 - val_loss: 1.2972 - val_accuracy: 0.4821 - lr: 4.0000e-05
Epoch 78/150
428/428 [==============================] - 230s 537ms/step - loss: 1.3231 -
accuracy: 0.4774 - val_loss: 1.2900 - val_accuracy: 0.4887 - lr: 4.0000e-05
Epoch 79/150
428/428 [==============================] - 231s 539ms/step - loss: 1.3197 -
accuracy: 0.4788 - val_loss: 1.2889 - val_accuracy: 0.4863 - lr: 4.0000e-05
Epoch 80/150
428/428 [==============================] - 228s 534ms/step - loss: 1.3232 -
accuracy: 0.4769 - val_loss: 1.3044 - val_accuracy: 0.4786 - lr: 4.0000e-05
Epoch 81/150
428/428 [==============================] - 228s 534ms/step - loss: 1.3193 -
accuracy: 0.4790 - val_loss: 1.2928 - val_accuracy: 0.4871 - lr: 4.0000e-05
Epoch 82/150
428/428 [==============================] - 229s 534ms/step - loss: 1.3141 -
accuracy: 0.4777 - val_loss: 1.3038 - val_accuracy: 0.4769 - lr: 4.0000e-05
Epoch 83/150
428/428 [==============================] - 229s 536ms/step - loss: 1.3122 -
accuracy: 0.4815 - val_loss: 1.2877 - val_accuracy: 0.4841 - lr: 4.0000e-05
Epoch 84/150
428/428 [==============================] - 230s 537ms/step - loss: 1.3155 -
accuracy: 0.4793 - val_loss: 1.2829 - val_accuracy: 0.4853 - lr: 4.0000e-05
Epoch 85/150
428/428 [==============================] - 227s 530ms/step - loss: 1.3092 -
accuracy: 0.4801 - val_loss: 1.2917 - val_accuracy: 0.4839 - lr: 4.0000e-05
Epoch 86/150
428/428 [==============================] - 228s 533ms/step - loss: 1.3111 -
accuracy: 0.4793 - val_loss: 1.2861 - val_accuracy: 0.4827 - lr: 4.0000e-05
Epoch 87/150
428/428 [==============================] - 230s 537ms/step - loss: 1.3061 -
accuracy: 0.4838 - val_loss: 1.2902 - val_accuracy: 0.4861 - lr: 4.0000e-05
```

```
Epoch 88/150
428/428 [==============================] - 229s 536ms/step - loss: 1.3150 -
accuracy: 0.4831 - val_loss: 1.2874 - val_accuracy: 0.4842 - lr: 4.0000e-05
Epoch 89/150
428/428 [==============================] - ETA: 0s - loss: 1.3068 - accuracy:
0.4822
Epoch 89: ReduceLROnPlateau reducing learning rate to 1.5999999595806004e-05.
428/428 [==============================] - 230s 538ms/step - loss: 1.3068 -
accuracy: 0.4822 - val_loss: 1.2851 - val_accuracy: 0.4894 - lr: 4.0000e-05
Epoch 90/150
428/428 [==============================] - 228s 532ms/step - loss: 1.2997 -
accuracy: 0.4861 - val_loss: 1.2747 - val_accuracy: 0.4887 - lr: 1.6000e-05
Epoch 91/150
428/428 [==============================] - 232s 542ms/step - loss: 1.2899 -
accuracy: 0.4868 - val_loss: 1.2752 - val_accuracy: 0.4898 - lr: 1.6000e-05
Epoch 92/150
428/428 [==============================] - 229s 535ms/step - loss: 1.2906 -
accuracy: 0.4873 - val_loss: 1.2749 - val_accuracy: 0.4867 - lr: 1.6000e-05
Epoch 93/150
428/428 [==============================] - 227s 531ms/step - loss: 1.2880 -
accuracy: 0.4881 - val_loss: 1.2768 - val_accuracy: 0.4888 - lr: 1.6000e-05
Epoch 94/150
428/428 [==============================] - 226s 529ms/step - loss: 1.2860 -
accuracy: 0.4865 - val_loss: 1.2826 - val_accuracy: 0.4874 - lr: 1.6000e-05
Epoch 95/150
428/428 [==============================] - 228s 533ms/step - loss: 1.2859 -
accuracy: 0.4906 - val_loss: 1.2792 - val_accuracy: 0.4867 - lr: 1.6000e-05
Epoch 96/150
428/428 [==============================] - ETA: 0s - loss: 1.2869 - accuracy:
0.4917
Epoch 96: ReduceLROnPlateau reducing learning rate to 6.399999983841554e-06.
428/428 [==============================] - 228s 532ms/step - loss: 1.2869 -
accuracy: 0.4917 - val_loss: 1.2766 - val_accuracy: 0.4911 - lr: 1.6000e-05
Epoch 97/150
428/428 [==============================] - 226s 528ms/step - loss: 1.2789 -
accuracy: 0.4902 - val_loss: 1.2728 - val_accuracy: 0.4895 - lr: 6.4000e-06
Epoch 98/150
428/428 [==============================] - 230s 538ms/step - loss: 1.2822 -
accuracy: 0.4923 - val_loss: 1.2703 - val_accuracy: 0.4924 - lr: 6.4000e-06
Epoch 99/150
428/428 [==============================] - 230s 537ms/step - loss: 1.2785 -
accuracy: 0.4928 - val_loss: 1.2702 - val_accuracy: 0.4900 - lr: 6.4000e-06
Epoch 100/150
428/428 [==============================] - 227s 531ms/step - loss: 1.2771 -
accuracy: 0.4912 - val_loss: 1.2692 - val_accuracy: 0.4920 - lr: 6.4000e-06
Epoch 101/150
428/428 [==============================] - 227s 531ms/step - loss: 1.2774 -
accuracy: 0.4932 - val_loss: 1.2720 - val_accuracy: 0.4910 - lr: 6.4000e-06
```

```
Epoch 102/150
428/428 [==============================] - ETA: 0s - loss: 1.2796 - accuracy:
0.4892
Epoch 102: ReduceLROnPlateau reducing learning rate to 2.5600000299164097e-06.
428/428 [==============================] - 231s 540ms/step - loss: 1.2796 -
accuracy: 0.4892 - val_loss: 1.2699 - val_accuracy: 0.4909 - lr: 6.4000e-06
Epoch 103/150
428/428 [==============================] - 229s 535ms/step - loss: 1.2744 -
accuracy: 0.4931 - val_loss: 1.2700 - val_accuracy: 0.4910 - lr: 2.5600e-06
Epoch 104/150
428/428 [==============================] - 236s 552ms/step - loss: 1.2773 -
accuracy: 0.4904 - val_loss: 1.2718 - val_accuracy: 0.4914 - lr: 2.5600e-06
Epoch 105/150
428/428 [==============================] - 248s 581ms/step - loss: 1.2741 -
accuracy: 0.4932 - val_loss: 1.2706 - val_accuracy: 0.4909 - lr: 2.5600e-06
Epoch 106/150
428/428 [==============================] - 238s 557ms/step - loss: 1.2739 -
accuracy: 0.4924 - val_loss: 1.2702 - val_accuracy: 0.4922 - lr: 2.5600e-06
Epoch 107/150
428/428 [==============================] - 232s 542ms/step - loss: 1.2744 -
accuracy: 0.4913 - val_loss: 1.2739 - val_accuracy: 0.4887 - lr: 2.5600e-06
Epoch 108/150
428/428 [==============================] - ETA: 0s - loss: 1.2762 - accuracy:
0.4963
Epoch 108: ReduceLROnPlateau reducing learning rate to 1.0239999937766699e-06.
428/428 [==============================] - 236s 550ms/step - loss: 1.2762 -
accuracy: 0.4963 - val_loss: 1.2705 - val_accuracy: 0.4916 - lr: 2.5600e-06
Epoch 109/150
428/428 [==============================] - 244s 571ms/step - loss: 1.2745 -
accuracy: 0.4931 - val_loss: 1.2699 - val_accuracy: 0.4919 - lr: 1.0240e-06
Epoch 110/150
428/428 [==============================] - 229s 536ms/step - loss: 1.2737 -
accuracy: 0.4940 - val_loss: 1.2699 - val_accuracy: 0.4917 - lr: 1.0240e-06
Epoch 111/150
428/428 [==============================] - 229s 535ms/step - loss: 1.2733 -
accuracy: 0.4922 - val_loss: 1.2695 - val_accuracy: 0.4910 - lr: 1.0240e-06
Epoch 112/150
428/428 [==============================] - 230s 538ms/step - loss: 1.2742 -
accuracy: 0.4920 - val_loss: 1.2693 - val_accuracy: 0.4932 - lr: 1.0240e-06
Epoch 113/150
428/428 [==============================] - ETA: 0s - loss: 1.2734 - accuracy:
0.4918
Epoch 113: ReduceLROnPlateau reducing learning rate to 4.095999884157209e-07.
428/428 [==============================] - 229s 534ms/step - loss: 1.2734 -
accuracy: 0.4918 - val_loss: 1.2694 - val_accuracy: 0.4917 - lr: 1.0240e-06
Epoch 114/150
428/428 [==============================] - 237s 555ms/step - loss: 1.2730 -
accuracy: 0.4915 - val_loss: 1.2694 - val_accuracy: 0.4912 - lr: 4.0960e-07
```

```
Epoch 115/150
428/428 [==============================] - 239s 559ms/step - loss: 1.2722 -
accuracy: 0.4931 - val_loss: 1.2693 - val_accuracy: 0.4918 - lr: 4.0960e-07
Epoch 116/150
428/428 [==============================] - 232s 541ms/step - loss: 1.2726 -
accuracy: 0.4934 - val_loss: 1.2697 - val_accuracy: 0.4920 - lr: 4.0960e-07
Epoch 117/150
428/428 [==============================] - ETA: 0s - loss: 1.2776 - accuracy:
0.4888
Epoch 117: ReduceLROnPlateau reducing learning rate to 1.6383999081881485e-07.
428/428 [==============================] - 237s 554ms/step - loss: 1.2776 -
accuracy: 0.4888 - val_loss: 1.2693 - val_accuracy: 0.4930 - lr: 4.0960e-07
Epoch 118/150
428/428 [==============================] - 232s 543ms/step - loss: 1.2771 -
accuracy: 0.4912 - val_loss: 1.2696 - val_accuracy: 0.4923 - lr: 1.6384e-07
Epoch 119/150
428/428 [==============================] - ETA: 0s - loss: 1.2730 - accuracy:
0.4906
Epoch 119: ReduceLROnPlateau reducing learning rate to 6.553599405378919e-08.
428/428 [==============================] - 232s 542ms/step - loss: 1.2730 -
accuracy: 0.4906 - val_loss: 1.2696 - val_accuracy: 0.4919 - lr: 1.6384e-07
Epoch 120/150
428/428 [==============================] - 229s 534ms/step - loss: 1.2703 -
accuracy: 0.4960 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 6.5536e-08
Epoch 121/150
428/428 [==============================] - 236s 551ms/step - loss: 1.2736 -
accuracy: 0.4940 - val_loss: 1.2694 - val_accuracy: 0.4921 - lr: 6.5536e-08
Epoch 122/150
428/428 [==============================] - ETA: 0s - loss: 1.2706 - accuracy:
0.4932
Epoch 122: ReduceLROnPlateau reducing learning rate to 2.6214397053081487e-08.
428/428 [==============================] - 229s 536ms/step - loss: 1.2706 -
accuracy: 0.4932 - val_loss: 1.2695 - val_accuracy: 0.4919 - lr: 6.5536e-08
Epoch 123/150
428/428 [==============================] - 230s 537ms/step - loss: 1.2713 -
accuracy: 0.4923 - val_loss: 1.2695 - val_accuracy: 0.4921 - lr: 2.6214e-08
Epoch 124/150
428/428 [==============================] - ETA: 0s - loss: 1.2710 - accuracy:
0.4949
Epoch 124: ReduceLROnPlateau reducing learning rate to 1.0485759105449689e-08.
428/428 [==============================] - 232s 542ms/step - loss: 1.2710 -
accuracy: 0.4949 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 2.6214e-08
Epoch 125/150
428/428 [==============================] - 232s 542ms/step - loss: 1.2708 -
accuracy: 0.4950 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0486e-08
Epoch 126/150
428/428 [==============================] - ETA: 0s - loss: 1.2721 - accuracy:
0.4927
```

```
Epoch 126: ReduceLROnPlateau reducing learning rate to 4.194303571125602e-09.
428/428 [==============================] - 231s 539ms/step - loss: 1.2721 -
accuracy: 0.4927 - val_loss: 1.2695 - val_accuracy: 0.4923 - lr: 1.0486e-08
Epoch 127/150
428/428 [==============================] - 236s 552ms/step - loss: 1.2691 -
accuracy: 0.4968 - val_loss: 1.2695 - val_accuracy: 0.4923 - lr: 4.1943e-09
Epoch 128/150
428/428 [==============================] - 234s 547ms/step - loss: 1.2747 -
accuracy: 0.4926 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 4.1943e-09
Epoch 129/150
428/428 [==============================] - ETA: 0s - loss: 1.2729 - accuracy:
0.4929
Epoch 129: ReduceLROnPlateau reducing learning rate to 1.6777214284502408e-09.
428/428 [==============================] - 234s 548ms/step - loss: 1.2729 -
accuracy: 0.4929 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 4.1943e-09
Epoch 130/150
428/428 [==============================] - 233s 545ms/step - loss: 1.2717 -
accuracy: 0.4925 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.6777e-09
Epoch 131/150
428/428 [==============================] - ETA: 0s - loss: 1.2709 - accuracy:
0.4932
Epoch 131: ReduceLROnPlateau reducing learning rate to 6.710885624983121e-10.
428/428 [==============================] - 235s 549ms/step - loss: 1.2709 -
accuracy: 0.4932 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.6777e-09
Epoch 132/150
428/428 [==============================] - 240s 562ms/step - loss: 1.2693 -
accuracy: 0.4957 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 6.7109e-10
Epoch 133/150
428/428 [==============================] - ETA: 0s - loss: 1.2703 - accuracy:
0.4954
Epoch 133: ReduceLROnPlateau reducing learning rate to 2.684354294402169e-10.
428/428 [==============================] - 244s 571ms/step - loss: 1.2703 -
accuracy: 0.4954 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 6.7109e-10
Epoch 134/150
428/428 [==============================] - 235s 548ms/step - loss: 1.2720 -
accuracy: 0.4917 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 2.6844e-10
Epoch 135/150
428/428 [==============================] - ETA: 0s - loss: 1.2706 - accuracy:
0.4948
Epoch 135: ReduceLROnPlateau reducing learning rate to 1.0737417621697888e-10.
428/428 [==============================] - 234s 548ms/step - loss: 1.2706 -
accuracy: 0.4948 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 2.6844e-10
Epoch 136/150
428/428 [==============================] - 234s 547ms/step - loss: 1.2717 -
accuracy: 0.4938 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0737e-10
Epoch 137/150
428/428 [==============================] - ETA: 0s - loss: 1.2765 - accuracy:
0.4926
```

```
Epoch 137: ReduceLROnPlateau reducing learning rate to 1e-10.
428/428 [==============================] - 233s 544ms/step - loss: 1.2765 -
accuracy: 0.4926 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0737e-10
Epoch 138/150
428/428 [==============================] - 234s 547ms/step - loss: 1.2710 -
accuracy: 0.4959 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 139/150
428/428 [==============================] - 235s 550ms/step - loss: 1.2739 -
accuracy: 0.4913 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 140/150
428/428 [==============================] - 236s 552ms/step - loss: 1.2692 -
accuracy: 0.4957 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 141/150
428/428 [==============================] - 238s 557ms/step - loss: 1.2705 -
accuracy: 0.4943 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 142/150
428/428 [==============================] - 233s 545ms/step - loss: 1.2696 -
accuracy: 0.4950 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 143/150
428/428 [==============================] - 237s 554ms/step - loss: 1.2742 -
accuracy: 0.4929 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 144/150
428/428 [==============================] - 235s 550ms/step - loss: 1.2730 -
accuracy: 0.4946 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 145/150
428/428 [==============================] - 240s 561ms/step - loss: 1.2728 -
accuracy: 0.4939 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 146/150
428/428 [==============================] - 236s 551ms/step - loss: 1.2756 -
accuracy: 0.4913 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 147/150
428/428 [==============================] - 233s 545ms/step - loss: 1.2751 -
accuracy: 0.4938 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 148/150
428/428 [==============================] - 237s 554ms/step - loss: 1.2699 -
accuracy: 0.4940 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 149/150
428/428 [==============================] - 232s 541ms/step - loss: 1.2726 -
accuracy: 0.4923 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
Epoch 150/150
428/428 [==============================] - 234s 547ms/step - loss: 1.2758 -
accuracy: 0.4916 - val_loss: 1.2695 - val_accuracy: 0.4922 - lr: 1.0000e-10
```

[13]:
```
!mkdir saved_model_LSTM
model.save('saved_model_LSTM/ser_LSTM_model')
```

```
WARNING:absl:Found untraced functions such as _update_step_xla,
lstm_cell_layer_call_fn, lstm_cell_layer_call_and_return_conditional_losses,
```

```
lstm_cell_1_layer_call_fn, lstm_cell_1_layer_call_and_return_conditional_losses
while saving (showing 5 of 7). These functions will not be directly callable
after loading.
```

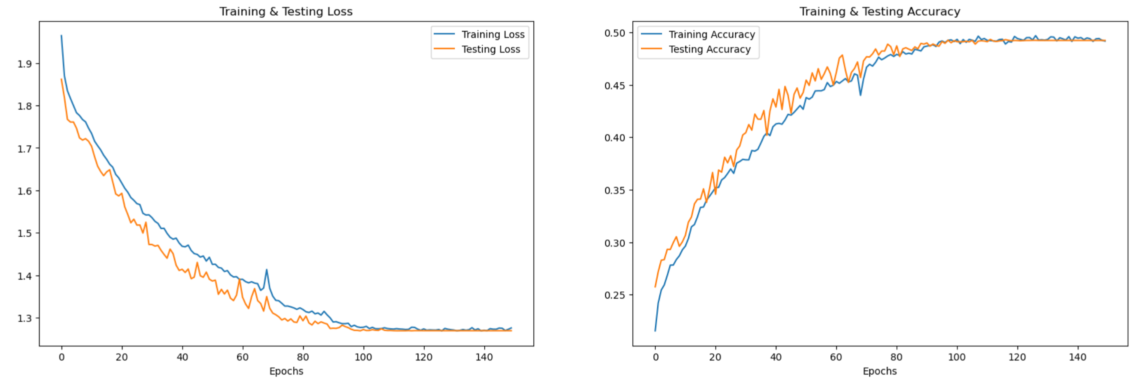
```
INFO:tensorflow:Assets written to: saved_model_LSTM/ser_LSTM_model\assets
```

```
INFO:tensorflow:Assets written to: saved_model_LSTM/ser_LSTM_model\assets
```

[14]:

```
print("Accuracy of our model on test data : " , model.
 ↪evaluate(X_test,y_test)[1]*100 , "%")

epochs = [i for i in range(len(train_loss))]
# epochs = range(50)
print(epochs)
fig , ax = plt.subplots(1,2)
train_acc = history.history['accuracy']
train_loss = history.history['loss']
test_acc = history.history['val_accuracy']
test_loss = history.history['val_loss']

fig.set_size_inches(20,6)
ax[0].plot(epochs , train_loss , label = 'Training Loss')
ax[0].plot(epochs , test_loss , label = 'Testing Loss')
ax[0].set_title('Training & Testing Loss')
ax[0].legend()
ax[0].set_xlabel("Epochs")

ax[1].plot(epochs , train_acc , label = 'Training Accuracy')
ax[1].plot(epochs , test_acc , label = 'Testing Accuracy')
ax[1].set_title('Training & Testing Accuracy')
ax[1].legend()
ax[1].set_xlabel("Epochs")
plt.show()
```

```
286/286 [==============================] - 32s 113ms/step - loss: 1.2695 -
accuracy: 0.4922
Accuracy of our model on test data :  49.22166168689728 %
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21,
22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41,
42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100,
101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116,
117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132,
133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148,
149]
```



[15]:
```
# predicting on test data.
pred_test = model.predict(X_test)
print(pred_test)
```

```
286/286 [==============================] - 32s 108ms/step
[[2.30481452e-03 3.23933782e-05 1.70706213e-01 … 6.46013170e-02
  5.21407247e-01 5.48360276e-06]
 [1.90624863e-01 1.60514930e-04 4.11512971e-01 … 4.22188714e-02
  1.22685535e-02 1.79257281e-02]
 [9.70186710e-01 2.23937874e-20 4.99473070e-04 … 2.63756084e-09
  9.56180468e-09 3.94728158e-05]
 …
 [5.54905951e-01 9.68173097e-08 1.01042375e-01 … 1.79741776e-03
  1.18006603e-03 4.94719297e-03]
 [1.03912890e-01 3.72209121e-04 2.86538690e-01 … 2.00211346e-01
  5.43016680e-02 1.31174398e-03]
 [2.65575367e-10 6.06588174e-12 7.58137845e-04 … 9.76915355e-04
  9.94731784e-01 2.32304874e-17]]
```

[16]:
```
pred_test.shape
```

[16]:
```
(9122, 8)
```

[17]:
```
y_pred = encoder.inverse_transform(pred_test)
print(y_pred)
print(y_pred.shape)
```

```
[['sad']
 ['disgust']
 ['angry']
 …
 ['angry']
 ['disgust']
 ['sad']]
(9122, 1)
```

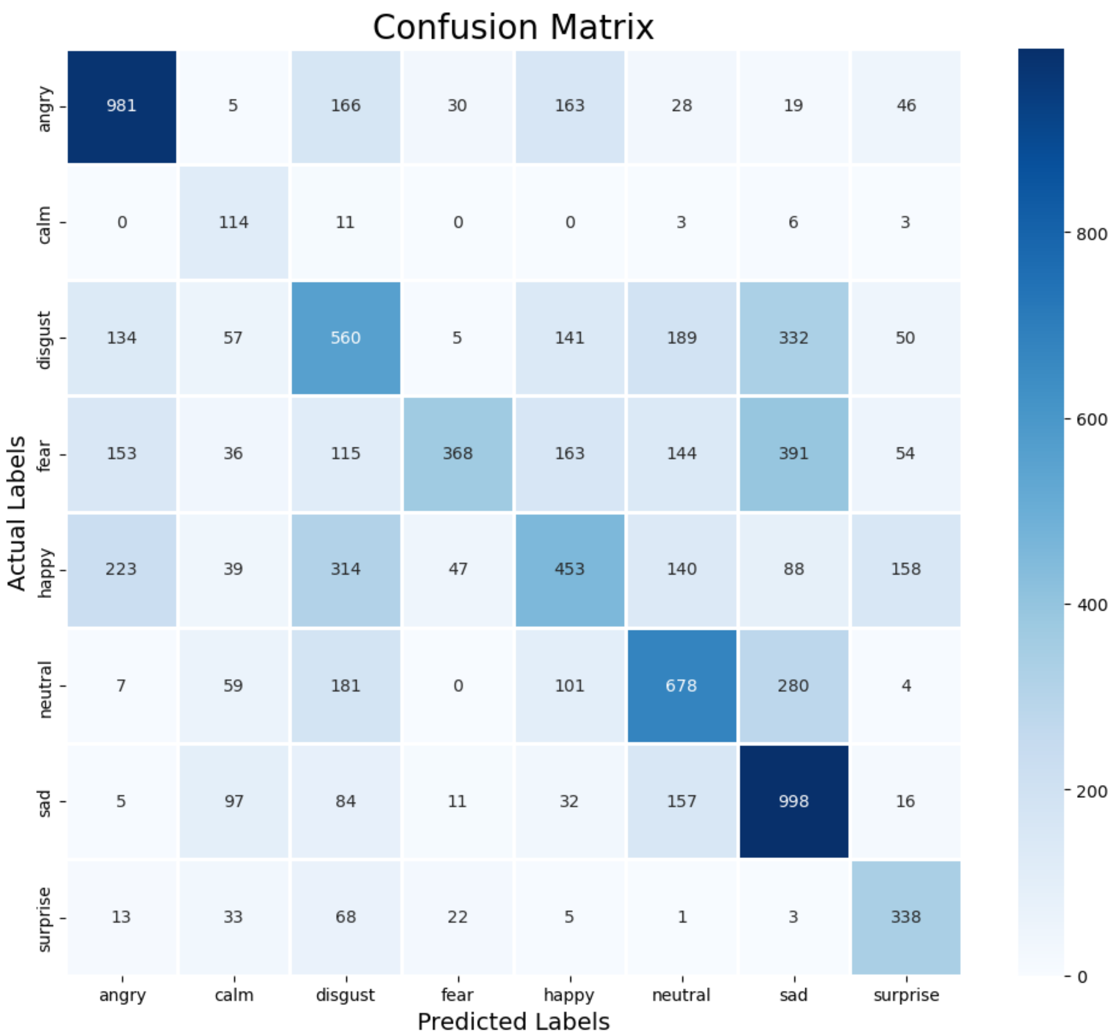
[18]:
```
y_test = encoder.inverse_transform(y_test)
```

[19]:
```
df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
df['Predicted Labels'] = y_pred.flatten()
df['Actual Labels'] = y_test.flatten()
```

[20]:
```
df.head(100)
```

[20]:
```
   Predicted Labels Actual Labels
0               sad       disgust
1           disgust       disgust
2             angry         angry
3           disgust       disgust
4           neutral          fear
..              …             …
95             calm          calm
96             fear          fear
97          disgust         angry
98              sad          fear
99            angry         angry

[100 rows x 2 columns]
```

[21]:
```
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize = (12, 10))
cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] , columns = [i␣
 ↪for i in encoder.categories_])
sns.heatmap(cm, linecolor='white', cmap='Blues', linewidth=1, annot=True,␣
 ↪fmt='')
plt.title('Confusion Matrix', size=20)
plt.xlabel('Predicted Labels', size=14)
plt.ylabel('Actual Labels', size=14)
plt.show()
```



[22]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

       angry       0.65      0.68      0.66      1438
        calm       0.26      0.83      0.40       137
     disgust       0.37      0.38      0.38      1468
        fear       0.76      0.26      0.39      1424
       happy       0.43      0.31      0.36      1462
     neutral       0.51      0.52      0.51      1310
         sad       0.47      0.71      0.57      1400
    surprise       0.51      0.70      0.59       483

    accuracy                           0.49      9122
   macro avg       0.49      0.55      0.48      9122
```

```
weighted avg       0.53      0.49      0.48      9122
```