Sppech Emotion Recognition

Speech emotion recognition (SER) is the field of technology focused on identifying the emotional state of a speaker from their voice. This goes beyond the words spoken and analyzes how they are spoken.

```
from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly
remount, call drive.mount("/content/drive", force_remount=True).
```

Dataset Used : Ryerson Audio-Visual Database of Emotional Speech and Son(Ravdess)

```
dataset_path = '/content/drive/MyDrive/Ravdess all data'
```

About Dataset : RAVDESS is one of the most common dataset used for this excercise by others. It's well liked because of its quality of speakers, recording and it has 24 actors of different genders. And there's more! You can get it in song format as well. There's something for everyone and their research project. So for convenience, here's 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)

Data Augmentation to introduce irregularities

```
from tqdm.auto import tqdm
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from keras.utils import to_categorical
from keras.models import Sequential
from keras import layers, optimizers, callbacks
from keras.models import Sequential
from keras import layers, callbacks
from keras.layers import Dense, Conv1D, MaxPooling1D, Flatten,
Dropout, BatchNormalization
from keras.callbacks import ReduceLROnPlateau
from sklearn.metrics import confusion_matrix, classification_report

import librosa
import os
import numpy as np
import pandas as pd
import random
import keras
import matplotlib.pyplot as plt
import seaborn as sns

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)
    return np.roll(data, shift_range)

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

def calculate_mfcc(signal):
  # Computes the MFCCs of the audio signal
  mfcc = np.mean(librosa.feature.mfcc(y=signal).T, axis = 0)
  return mfcc

def calculate_zcr(signal):
  # Computes the ZCR of the audio signal
```

```
  return np.mean(librosa.feature.zero_crossing_rate(y = signal).T,
axis = 0)

def calculate_rms(signal):
  # Computes the RMS value of the audio signal
  return np.mean(librosa.feature.rms(y = signal).T, axis = 0)

# Function to extract features from the audio signal
def extract_features(signal):
  res = np.array([])

  # Calculate Zero Crossing Rate
  zcr = calculate_zcr(signal)
  res = np.hstack((res, zcr))

  # Calculate Mel-Frequency Cepstral Coefficients
  mfcc = calculate_mfcc(signal)
  res = np.hstack((res, mfcc))

  # Calculate Root Mean Square Value
  rms = calculate_rms(signal)
  res = np.hstack((res, rms))

  return res

def get_features(audio_file_path):
  #loads the audio file
  signal, sr = librosa.load(audio_file_path, duration = 2.5, offset =
0.6)
  #extract features from the data without augmentation
  res1 = extract_features(signal)
  result = np.array(res1)

  #add noise and then extract the features
  noise_data = noise(signal)
  res2 = extract_features(noise_data)
  result = np.vstack((result, res2))

  #Add Stretch, pitch shift and extract the features
  new_data = stretch(signal)
  data_stretch_pitch = pitch(new_data, sr)
  res3 = extract_features(data_stretch_pitch)
  result = np.vstack((result, res3))

  return result

def prepare_data(X, Y, Y_gender, Y_emotion_gender):
  # walk through the dataset and process each file
  for path, directories, files in tqdm(os.walk(dataset_path)):
    for file in files:
```

```
      audio_file_path = os.path.join(path, file)
      features = get_features(audio_file_path)
      for element in features:
        #Append features to the lists
        X.append(element)
        #Assign emotion labels
        Y.append(emotion_dict[int(file[7:8])-1])
        gender = int(file[-6:-4])
        if (gender%2 == 0):
          #Assign genders
          Y_gender.append("Female")
          emotion_gender = emotion_dict[int(file[7:8])-1] + "_" +
"Female"
        else:
          Y_gender.append("Male")
          emotion_gender = emotion_dict[int(file[7:8])-1] + "_" +
"Male"
        #Assign emotion and gender labels
        Y_emotion_gender.append(emotion_gender)
  return X, Y, Y_gender, Y_emotion_gender

def convert_to_dataframes(X, Y):
  #convert the features to dataframe
  Features = pd.DataFrame(X)
  #Append different labels into the dataframes
  Features['labels'] = Y
  return Features

def segregate_data(df):
  #Segregate into training data
  X = df.iloc[: ,:-1].values
  #segregate into target data
  Y = df['labels'].values
  return X, Y

def perform_one_hot_encoding(Y):
  #perform one hot encoding to the labels
  return encoder.fit_transform(np.array(Y).reshape(-1,1)).toarray()

def split_the_data(X, Y):
  #split the data into train and test
  return train_test_split(X, Y, random_state=10, shuffle=True)

def apply_fit_transform(x_train, x_test):
  # This adjusts the training data so that it has a mean of 0 and
standard deviation of 1, feature-wise.
  x_train = scaler.fit_transform(x_train)
  # This ensures that the testing data is scaled using the same
parameters as the training data.
```

```
  x_test = scaler.transform(x_test)
  return x_train, x_test

def add_third_dimension(x_train, x_test):
  #Adding 3rd dimension to the train data
  x_train = np.expand_dims(x_train, axis=2)
  #adding 3rd dimension to the test data
  x_test = np.expand_dims(x_test, axis=2)
  return x_train, x_test

def create_model_emotional(x_train):
  model=Sequential()

  model.add(Conv1D(128, 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(64, 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(32, kernel_size=5, strides=1, padding='same',
activation='relu'))
  model.add(MaxPooling1D(pool_size=5, strides = 2, padding = 'same'))
  model.add(Dropout(0.4))

  model.add(Flatten())
  model.add(Dense(units=16, activation='relu'))
  model.add(Dropout(0.2))

  model.add(Dense(units=7, activation='softmax'))
  model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy'
, metrics = ['accuracy'])

  model.summary()
  return model

def create_model_gender(x_train):
  model=Sequential()
  model.add(Conv1D(16, 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(Dropout(0.2))

  model.add(Flatten())
  model.add(Dense(units=8, activation='relu'))
  model.add(Dropout(0.4))

  model.add(Dense(units=2, activation='softmax'))
  model.compile(optimizer = 'adam' , loss = 'binary_crossentropy' ,
```

```
metrics = ['accuracy'])

  model.summary()
  return model

def create_model_emotion_gender(x_train):
  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(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(Dropout(0.4))

  model.add(Conv1D(32, 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(Flatten())
  model.add(Dense(units=16, activation='relu'))
  model.add(Dropout(0.4))

  model.add(Dense(units=14, activation='softmax'))
  model.compile(optimizer = 'adam' , loss = 'categorical_crossentropy'
, metrics = ['accuracy'])

  model.summary()
  return model

def fit_model(x_train, y_train, x_test, y_test, lr, model):
  # Initialize ReduceLROnPlateau callback
  rlrp = ReduceLROnPlateau(monitor='loss', factor=0.4, verbose=0,
patience=4, min_lr = lr)
  # Fit the model on the training data
  history=model.fit(x_train, y_train, batch_size=32, epochs=100,
validation_data=(x_test, y_test), callbacks=[rlrp])
  return history

def model_evaluation(model, x_test, y_test):
  print("Accuracy of our model on test data : " ,
model.evaluate(x_test,y_test)[1]*100 , "%")

def create_graphs_for_loss_and_accuracy(history):
```

```
  epochs = [i for i in range(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']

  fig.set_size_inches(14,6)
  ax[0].plot(epochs , train_loss , label = 'Training Loss',marker='o',
linewidth=1)
  ax[0].plot(epochs , test_loss , label = 'Testing Loss',marker='.',
linewidth=1)
  ax[0].set_title('Training & Testing Loss')
  ax[0].legend()
  ax[0].set_xlabel("Epochs")

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

  plt.show()

def predict_model(model, x_test, y_test):
  pred_test = model.predict(x_test)
  y_pred = encoder.inverse_transform(pred_test)
  y_test = encoder.inverse_transform(y_test)
  return y_pred, y_test

def convert_predicted_actual_to_dataframe(y_pred, y_test):
  df = pd.DataFrame(columns=['Predicted Labels', 'Actual Labels'])
  df['Predicted Labels'] = y_pred.flatten()
  df['Actual Labels'] = y_test.flatten()
  return df

def create_confusion_matrix(y_test, y_pred):
  cm = confusion_matrix(y_test, y_pred)
  plt.figure(figsize = (10, 8))
  cm = pd.DataFrame(cm , index = [i for i in encoder.categories_] ,
columns = [i for i in encoder.categories_])
  sns.heatmap(cm, linecolor='white', cmap='Reds', linewidth=1,
annot=True, fmt='')
  plt.title('Confusion Matrix', size=12)
  plt.xlabel('Predicted Labels', size=8)
  plt.ylabel('Actual Labels', size=8)
  plt.show()
```

```
if __name__ == "__main__":
  Y = []
  X = []
  Y_gender = []
  Y_emotion_gender = []
  emotion_dict = {0:'neutral', 1:'neutral', 2:'happy', 3:'sad',
4:'angry', 5:'fearful', 6:'disgust', 7:'surprised'}
  lr = 1e-6

  X, Y, Y_gender, Y_emotion_gender = prepare_data(X, Y, Y_gender,
Y_emotion_gender)

  #convert data according to emotions
  Features = convert_to_dataframes(X, Y)
  Features.shape

  #segregate data according to predict emotions
  X, Y = segregate_data(Features)

  encoder = OneHotEncoder()
  #onehot encoding the data
  Y = perform_one_hot_encoding(Y)

  #Splitting the data
  x_train, x_test, y_train, y_test = split_the_data(X, Y)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  scaler = StandardScaler()
  #applying standard scaler
  x_train, x_test = apply_fit_transform(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  #Adding third dimension
  x_train, x_test = add_third_dimension(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  print("-------------------------------CREATING MODEL FOR PREDICTING
EMOTIONS-----------------------------------------------------------")
  #creating emotion model
  emotion_model = create_model_emotional(x_train)

  history = fit_model(x_train, y_train, x_test, y_test, lr,
emotion_model)

  #evaluate the model
  model_evaluation(emotion_model,x_test,y_test)

  #Create graph for model accuracy and loss
  create_graphs_for_loss_and_accuracy(history)

  #predict the model on the test data
```

```
  y_pred, y_test = predict_model(emotion_model, x_test, y_test)

  df = convert_predicted_actual_to_dataframe(y_pred, y_test)
  display(df.head(10))

  #creating a confusion matrix to see the predcition is more broder
way
  create_confusion_matrix(y_test, y_pred)
  print(classification_report(y_test, y_pred))


  print("-----------------------------Adjusting Data for gender
classification-------------------------------------------------------
-")

  Features.drop('labels', axis=1, inplace = True)
  Features['labels'] = Y_gender
  X, Y = segregate_data(Features)

  Y = perform_one_hot_encoding(Y)

  x_train, x_test, y_train, y_test = split_the_data(X, Y)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  x_train, x_test = apply_fit_transform(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  x_train, x_test = add_third_dimension(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  print("-------------------------------CREATING MODEL FOR PREDICTING
GENDER-----------------------------------------------------------")
  #creating emotion model
  gender_model = create_model_gender(x_train)

  history = fit_model(x_train, y_train, x_test, y_test, lr,
gender_model)

  #evaluate the model
  model_evaluation(gender_model,x_test,y_test)

  #Create graph for model accuracy and loss
  create_graphs_for_loss_and_accuracy(history)

  #predict the model on the test data
  y_pred, y_test = predict_model(gender_model, x_test, y_test)

  df = convert_predicted_actual_to_dataframe(y_pred, y_test)
  display(df.head(10))

  #creating a confusion matrix to see the predcition is more broder
```

```
way
  create_confusion_matrix(y_test, y_pred)
  print(classification_report(y_test, y_pred))

  print("-----------------------------Adjusting Data for gender and
emotion classification
together---------------------------------------------------------")

  Features.drop('labels', axis=1, inplace = True)
  Features['labels'] = Y_emotion_gender
  X, Y = segregate_data(Features)

  Y = perform_one_hot_encoding(Y)

  x_train, x_test, y_train, y_test = split_the_data(X, Y)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  x_train, x_test = apply_fit_transform(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  x_train, x_test = add_third_dimension(x_train, x_test)
  print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)

  print("-------------------------------CREATING MODEL FOR PREDICTING
GENDER AND EMOTION
TOGETHER-----------------------------------------------------------")

  #creating emotion model
  emotion_gender_model = create_model_emotion_gender(x_train)

  history = fit_model(x_train, y_train, x_test, y_test, lr,
emotion_gender_model)

  #evaluate the model
  model_evaluation(emotion_gender_model,x_test,y_test)

  #Create graph for model accuracy and loss
  create_graphs_for_loss_and_accuracy(history)

  #predict the model on the test data
  y_pred, y_test = predict_model(emotion_gender_model, x_test, y_test)

  df = convert_predicted_actual_to_dataframe(y_pred, y_test)
  display(df.head(10))

  #creating a confusion matrix to see the predcition is more broder
way
  create_confusion_matrix(y_test, y_pred)
  print(classification_report(y_test, y_pred))
```

{"model_id":"4da437a124bf49b182309ce3995fef73","version_major":2,"version_minor":0}

```
(3240, 22) (3240, 7) (1080, 22) (1080, 7)
(3240, 22) (3240, 7) (1080, 22) (1080, 7)
(3240, 22, 1) (3240, 7) (1080, 22, 1) (1080, 7)
-------------------------------CREATING MODEL FOR PREDICTING
EMOTIONS-------------------------------------------------------------------
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 22, 128)           768

 max_pooling1d (MaxPooling1  (None, 11, 128)           0
 D)

 conv1d_1 (Conv1D)           (None, 11, 64)            41024

 max_pooling1d_1 (MaxPoolin  (None, 6, 64)             0
 g1D)

 dropout (Dropout)           (None, 6, 64)             0

 conv1d_2 (Conv1D)           (None, 6, 32)             10272

 max_pooling1d_2 (MaxPoolin  (None, 3, 32)             0
 g1D)

 dropout_1 (Dropout)         (None, 3, 32)             0

 flatten (Flatten)           (None, 96)                0

 dense (Dense)               (None, 16)                1552

 dropout_2 (Dropout)         (None, 16)                0

 dense_1 (Dense)             (None, 7)                 119

=================================================================
Total params: 53735 (209.90 KB)
Trainable params: 53735 (209.90 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/100
102/102 [==============================] - 4s 19ms/step - loss: 1.9315
- accuracy: 0.1914 - val_loss: 1.8748 - val_accuracy: 0.2722 - lr:
0.0010
Epoch 2/100
102/102 [==============================] - 2s 15ms/step - loss: 1.8773
```

```
- accuracy: 0.2525 - val_loss: 1.8044 - val_accuracy: 0.2778 - lr:
0.0010
Epoch 3/100
102/102 [==============================] - 1s 10ms/step - loss: 1.8115
- accuracy: 0.2765 - val_loss: 1.7502 - val_accuracy: 0.2917 - lr:
0.0010
Epoch 4/100
102/102 [==============================] - 1s 10ms/step - loss: 1.7576
- accuracy: 0.3062 - val_loss: 1.7020 - val_accuracy: 0.3157 - lr:
0.0010
Epoch 5/100
102/102 [==============================] - 1s 10ms/step - loss: 1.7177
- accuracy: 0.3154 - val_loss: 1.6811 - val_accuracy: 0.3407 - lr:
0.0010
Epoch 6/100
102/102 [==============================] - 1s 11ms/step - loss: 1.6824
- accuracy: 0.3333 - val_loss: 1.6468 - val_accuracy: 0.3407 - lr:
0.0010
Epoch 7/100
102/102 [==============================] - 1s 11ms/step - loss: 1.6576
- accuracy: 0.3512 - val_loss: 1.6218 - val_accuracy: 0.3648 - lr:
0.0010
Epoch 8/100
102/102 [==============================] - 1s 10ms/step - loss: 1.6401
- accuracy: 0.3534 - val_loss: 1.5943 - val_accuracy: 0.3713 - lr:
0.0010
Epoch 9/100
102/102 [==============================] - 1s 10ms/step - loss: 1.6021
- accuracy: 0.3725 - val_loss: 1.6061 - val_accuracy: 0.3630 - lr:
0.0010
Epoch 10/100
102/102 [==============================] - 1s 10ms/step - loss: 1.5939
- accuracy: 0.3799 - val_loss: 1.5699 - val_accuracy: 0.3694 - lr:
0.0010
Epoch 11/100
102/102 [==============================] - 1s 10ms/step - loss: 1.5741
- accuracy: 0.3895 - val_loss: 1.5339 - val_accuracy: 0.4019 - lr:
0.0010
Epoch 12/100
102/102 [==============================] - 1s 14ms/step - loss: 1.5465
- accuracy: 0.3886 - val_loss: 1.5271 - val_accuracy: 0.3898 - lr:
0.0010
Epoch 13/100
102/102 [==============================] - 2s 15ms/step - loss: 1.5263
- accuracy: 0.4056 - val_loss: 1.5096 - val_accuracy: 0.4167 - lr:
0.0010
Epoch 14/100
102/102 [==============================] - 2s 17ms/step - loss: 1.5164
- accuracy: 0.4086 - val_loss: 1.5100 - val_accuracy: 0.4046 - lr:
```

```
0.0010
Epoch 15/100
102/102 [==============================] - 1s 11ms/step - loss: 1.4916
- accuracy: 0.4247 - val_loss: 1.4660 - val_accuracy: 0.4204 - lr:
0.0010
Epoch 16/100
102/102 [==============================] - 1s 10ms/step - loss: 1.4580
- accuracy: 0.4343 - val_loss: 1.4785 - val_accuracy: 0.4204 - lr:
0.0010
Epoch 17/100
102/102 [==============================] - 1s 10ms/step - loss: 1.4522
- accuracy: 0.4364 - val_loss: 1.4474 - val_accuracy: 0.4315 - lr:
0.0010
Epoch 18/100
102/102 [==============================] - 1s 10ms/step - loss: 1.4324
- accuracy: 0.4534 - val_loss: 1.4239 - val_accuracy: 0.4444 - lr:
0.0010
Epoch 19/100
102/102 [==============================] - 1s 10ms/step - loss: 1.4117
- accuracy: 0.4586 - val_loss: 1.4063 - val_accuracy: 0.4602 - lr:
0.0010
Epoch 20/100
102/102 [==============================] - 1s 11ms/step - loss: 1.3971
- accuracy: 0.4623 - val_loss: 1.4278 - val_accuracy: 0.4370 - lr:
0.0010
Epoch 21/100
102/102 [==============================] - 1s 10ms/step - loss: 1.3678
- accuracy: 0.4713 - val_loss: 1.3903 - val_accuracy: 0.4537 - lr:
0.0010
Epoch 22/100
102/102 [==============================] - 1s 10ms/step - loss: 1.3577
- accuracy: 0.4657 - val_loss: 1.4265 - val_accuracy: 0.4343 - lr:
0.0010
Epoch 23/100
102/102 [==============================] - 1s 11ms/step - loss: 1.3438
- accuracy: 0.4929 - val_loss: 1.3739 - val_accuracy: 0.4704 - lr:
0.0010
Epoch 24/100
102/102 [==============================] - 1s 13ms/step - loss: 1.3128
- accuracy: 0.4948 - val_loss: 1.3643 - val_accuracy: 0.4852 - lr:
0.0010
Epoch 25/100
102/102 [==============================] - 2s 17ms/step - loss: 1.3270
- accuracy: 0.4935 - val_loss: 1.3235 - val_accuracy: 0.4944 - lr:
0.0010
Epoch 26/100
102/102 [==============================] - 2s 17ms/step - loss: 1.2918
- accuracy: 0.5108 - val_loss: 1.3297 - val_accuracy: 0.5009 - lr:
0.0010
```

```
Epoch 27/100
102/102 [==============================] - 1s 10ms/step - loss: 1.2686
- accuracy: 0.5269 - val_loss: 1.3427 - val_accuracy: 0.4731 - lr:
0.0010
Epoch 28/100
102/102 [==============================] - 1s 10ms/step - loss: 1.2481
- accuracy: 0.5219 - val_loss: 1.3202 - val_accuracy: 0.4898 - lr:
0.0010
Epoch 29/100
102/102 [==============================] - 1s 10ms/step - loss: 1.2321
- accuracy: 0.5309 - val_loss: 1.2822 - val_accuracy: 0.5037 - lr:
0.0010
Epoch 30/100
102/102 [==============================] - 1s 10ms/step - loss: 1.2042
- accuracy: 0.5265 - val_loss: 1.2946 - val_accuracy: 0.4963 - lr:
0.0010
Epoch 31/100
102/102 [==============================] - 1s 10ms/step - loss: 1.1969
- accuracy: 0.5512 - val_loss: 1.2699 - val_accuracy: 0.5167 - lr:
0.0010
Epoch 32/100
102/102 [==============================] - 1s 10ms/step - loss: 1.1547
- accuracy: 0.5620 - val_loss: 1.2414 - val_accuracy: 0.5222 - lr:
0.0010
Epoch 33/100
102/102 [==============================] - 1s 10ms/step - loss: 1.1721
- accuracy: 0.5685 - val_loss: 1.2489 - val_accuracy: 0.5315 - lr:
0.0010
Epoch 34/100
102/102 [==============================] - 1s 10ms/step - loss: 1.1615
- accuracy: 0.5602 - val_loss: 1.2291 - val_accuracy: 0.5315 - lr:
0.0010
Epoch 35/100
102/102 [==============================] - 1s 10ms/step - loss: 1.1578
- accuracy: 0.5571 - val_loss: 1.2386 - val_accuracy: 0.5343 - lr:
0.0010
Epoch 36/100
102/102 [==============================] - 1s 13ms/step - loss: 1.1242
- accuracy: 0.5787 - val_loss: 1.3090 - val_accuracy: 0.5231 - lr:
0.0010
Epoch 37/100
102/102 [==============================] - 2s 16ms/step - loss: 1.1498
- accuracy: 0.5685 - val_loss: 1.2066 - val_accuracy: 0.5491 - lr:
0.0010
Epoch 38/100
102/102 [==============================] - 2s 17ms/step - loss: 1.1193
- accuracy: 0.5824 - val_loss: 1.2639 - val_accuracy: 0.5259 - lr:
0.0010
Epoch 39/100
```

```
102/102 [==============================] - 1s 10ms/step - loss: 1.0986
- accuracy: 0.5929 - val_loss: 1.2078 - val_accuracy: 0.5463 - lr:
0.0010
Epoch 40/100
102/102 [==============================] - 1s 11ms/step - loss: 1.0861
- accuracy: 0.5914 - val_loss: 1.1949 - val_accuracy: 0.5565 - lr:
0.0010
Epoch 41/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0933
- accuracy: 0.5920 - val_loss: 1.2042 - val_accuracy: 0.5593 - lr:
0.0010
Epoch 42/100
102/102 [==============================] - 1s 11ms/step - loss: 1.0740
- accuracy: 0.5966 - val_loss: 1.1913 - val_accuracy: 0.5667 - lr:
0.0010
Epoch 43/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0737
- accuracy: 0.6000 - val_loss: 1.2084 - val_accuracy: 0.5602 - lr:
0.0010
Epoch 44/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0508
- accuracy: 0.6068 - val_loss: 1.2085 - val_accuracy: 0.5556 - lr:
0.0010
Epoch 45/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0403
- accuracy: 0.6043 - val_loss: 1.1895 - val_accuracy: 0.5546 - lr:
0.0010
Epoch 46/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0445
- accuracy: 0.6034 - val_loss: 1.1732 - val_accuracy: 0.5620 - lr:
0.0010
Epoch 47/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9993
- accuracy: 0.6340 - val_loss: 1.2076 - val_accuracy: 0.5565 - lr:
0.0010
Epoch 48/100
102/102 [==============================] - 1s 13ms/step - loss: 1.0345
- accuracy: 0.6216 - val_loss: 1.1993 - val_accuracy: 0.5435 - lr:
0.0010
Epoch 49/100
102/102 [==============================] - 2s 16ms/step - loss: 1.0211
- accuracy: 0.6133 - val_loss: 1.1891 - val_accuracy: 0.5648 - lr:
0.0010
Epoch 50/100
102/102 [==============================] - 2s 15ms/step - loss: 1.0094
- accuracy: 0.6299 - val_loss: 1.1669 - val_accuracy: 0.5750 - lr:
0.0010
Epoch 51/100
102/102 [==============================] - 1s 11ms/step - loss: 0.9894
```

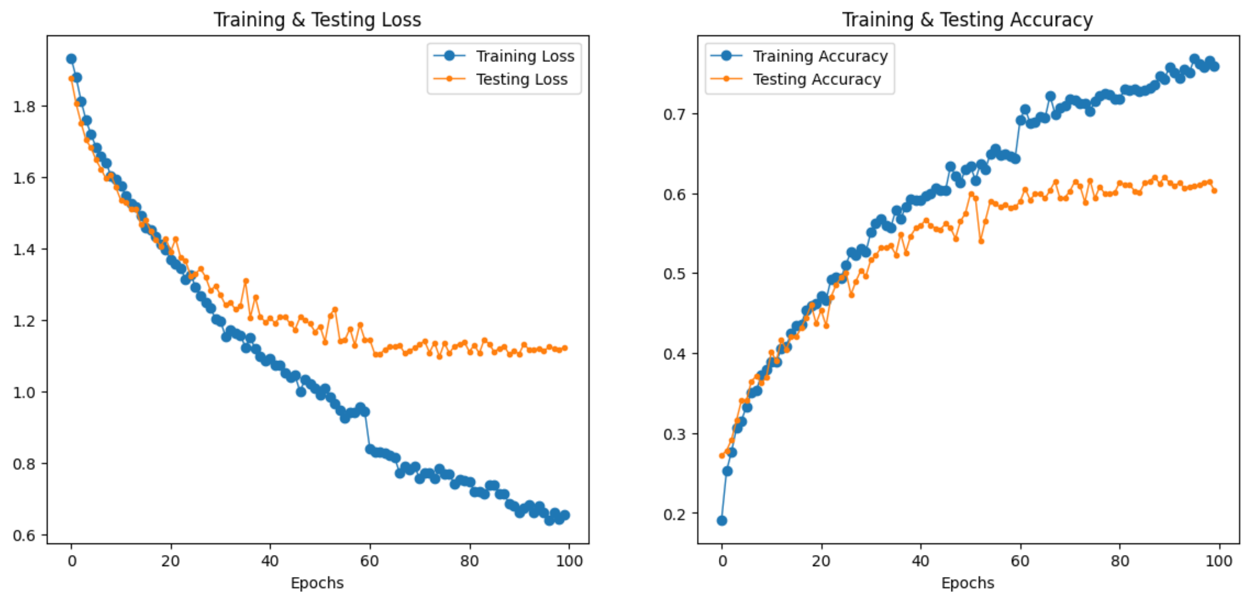
```
- accuracy: 0.6346 - val_loss: 1.1827 - val_accuracy: 0.5991 - lr:
0.0010
Epoch 52/100
102/102 [==============================] - 1s 10ms/step - loss: 1.0092
- accuracy: 0.6160 - val_loss: 1.1388 - val_accuracy: 0.5944 - lr:
0.0010
Epoch 53/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9849
- accuracy: 0.6367 - val_loss: 1.2127 - val_accuracy: 0.5398 - lr:
0.0010
Epoch 54/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9654
- accuracy: 0.6293 - val_loss: 1.2313 - val_accuracy: 0.5657 - lr:
0.0010
Epoch 55/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9492
- accuracy: 0.6497 - val_loss: 1.1424 - val_accuracy: 0.5898 - lr:
0.0010
Epoch 56/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9277
- accuracy: 0.6565 - val_loss: 1.1432 - val_accuracy: 0.5870 - lr:
0.0010
Epoch 57/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9402
- accuracy: 0.6481 - val_loss: 1.1760 - val_accuracy: 0.5824 - lr:
0.0010
Epoch 58/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9402
- accuracy: 0.6491 - val_loss: 1.1296 - val_accuracy: 0.5861 - lr:
0.0010
Epoch 59/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9565
- accuracy: 0.6457 - val_loss: 1.1874 - val_accuracy: 0.5815 - lr:
0.0010
Epoch 60/100
102/102 [==============================] - 1s 10ms/step - loss: 0.9440
- accuracy: 0.6441 - val_loss: 1.1451 - val_accuracy: 0.5824 - lr:
0.0010
Epoch 61/100
102/102 [==============================] - 2s 16ms/step - loss: 0.8417
- accuracy: 0.6923 - val_loss: 1.1448 - val_accuracy: 0.5898 - lr:
4.0000e-04
Epoch 62/100
102/102 [==============================] - 2s 17ms/step - loss: 0.8302
- accuracy: 0.7056 - val_loss: 1.1054 - val_accuracy: 0.6056 - lr:
4.0000e-04
Epoch 63/100
102/102 [==============================] - 1s 13ms/step - loss: 0.8322
- accuracy: 0.6877 - val_loss: 1.1046 - val_accuracy: 0.5917 - lr:
```

```
4.0000e-04
Epoch 64/100
102/102 [==============================] - 1s 11ms/step - loss: 0.8278
- accuracy: 0.6895 - val_loss: 1.1162 - val_accuracy: 0.5991 - lr:
4.0000e-04
Epoch 65/100
102/102 [==============================] - 1s 10ms/step - loss: 0.8231
- accuracy: 0.6963 - val_loss: 1.1270 - val_accuracy: 0.6000 - lr:
4.0000e-04
Epoch 66/100
102/102 [==============================] - 1s 10ms/step - loss: 0.8158
- accuracy: 0.6938 - val_loss: 1.1265 - val_accuracy: 0.5935 - lr:
4.0000e-04
Epoch 67/100
102/102 [==============================] - 1s 11ms/step - loss: 0.7722
- accuracy: 0.7222 - val_loss: 1.1301 - val_accuracy: 0.6037 - lr:
4.0000e-04
Epoch 68/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7908
- accuracy: 0.6991 - val_loss: 1.1069 - val_accuracy: 0.6148 - lr:
4.0000e-04
Epoch 69/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7832
- accuracy: 0.7062 - val_loss: 1.1138 - val_accuracy: 0.5935 - lr:
4.0000e-04
Epoch 70/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7901
- accuracy: 0.7096 - val_loss: 1.1217 - val_accuracy: 0.5944 - lr:
4.0000e-04
Epoch 71/100
102/102 [==============================] - 1s 11ms/step - loss: 0.7580
- accuracy: 0.7185 - val_loss: 1.1314 - val_accuracy: 0.6028 - lr:
4.0000e-04
Epoch 72/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7730
- accuracy: 0.7167 - val_loss: 1.1410 - val_accuracy: 0.6148 - lr:
4.0000e-04
Epoch 73/100
102/102 [==============================] - 2s 16ms/step - loss: 0.7735
- accuracy: 0.7117 - val_loss: 1.1086 - val_accuracy: 0.6093 - lr:
4.0000e-04
Epoch 74/100
102/102 [==============================] - 2s 17ms/step - loss: 0.7570
- accuracy: 0.7123 - val_loss: 1.1365 - val_accuracy: 0.5889 - lr:
4.0000e-04
Epoch 75/100
102/102 [==============================] - 1s 14ms/step - loss: 0.7842
- accuracy: 0.7028 - val_loss: 1.0973 - val_accuracy: 0.6167 - lr:
4.0000e-04
```

```
Epoch 76/100
102/102 [==============================] - 1s 11ms/step - loss: 0.7683
- accuracy: 0.7151 - val_loss: 1.1353 - val_accuracy: 0.5935 - lr:
4.0000e-04
Epoch 77/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7698
- accuracy: 0.7216 - val_loss: 1.1065 - val_accuracy: 0.6083 - lr:
4.0000e-04
Epoch 78/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7404
- accuracy: 0.7250 - val_loss: 1.1272 - val_accuracy: 0.5991 - lr:
4.0000e-04
Epoch 79/100
102/102 [==============================] - 1s 11ms/step - loss: 0.7557
- accuracy: 0.7228 - val_loss: 1.1327 - val_accuracy: 0.6000 - lr:
4.0000e-04
Epoch 80/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7503
- accuracy: 0.7182 - val_loss: 1.1383 - val_accuracy: 0.6009 - lr:
4.0000e-04
Epoch 81/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7486
- accuracy: 0.7182 - val_loss: 1.1112 - val_accuracy: 0.6139 - lr:
4.0000e-04
Epoch 82/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7205
- accuracy: 0.7309 - val_loss: 1.1299 - val_accuracy: 0.6102 - lr:
4.0000e-04
Epoch 83/100
102/102 [==============================] - 1s 10ms/step - loss: 0.7216
- accuracy: 0.7293 - val_loss: 1.1065 - val_accuracy: 0.6102 - lr:
4.0000e-04
Epoch 84/100
102/102 [==============================] - 1s 11ms/step - loss: 0.7148
- accuracy: 0.7296 - val_loss: 1.1451 - val_accuracy: 0.6028 - lr:
4.0000e-04
Epoch 85/100
102/102 [==============================] - 2s 15ms/step - loss: 0.7386
- accuracy: 0.7281 - val_loss: 1.1325 - val_accuracy: 0.6009 - lr:
4.0000e-04
Epoch 86/100
102/102 [==============================] - 2s 16ms/step - loss: 0.7374
- accuracy: 0.7287 - val_loss: 1.1105 - val_accuracy: 0.6130 - lr:
4.0000e-04
Epoch 87/100
102/102 [==============================] - 2s 16ms/step - loss: 0.7150
- accuracy: 0.7315 - val_loss: 1.1204 - val_accuracy: 0.6148 - lr:
4.0000e-04
Epoch 88/100
```

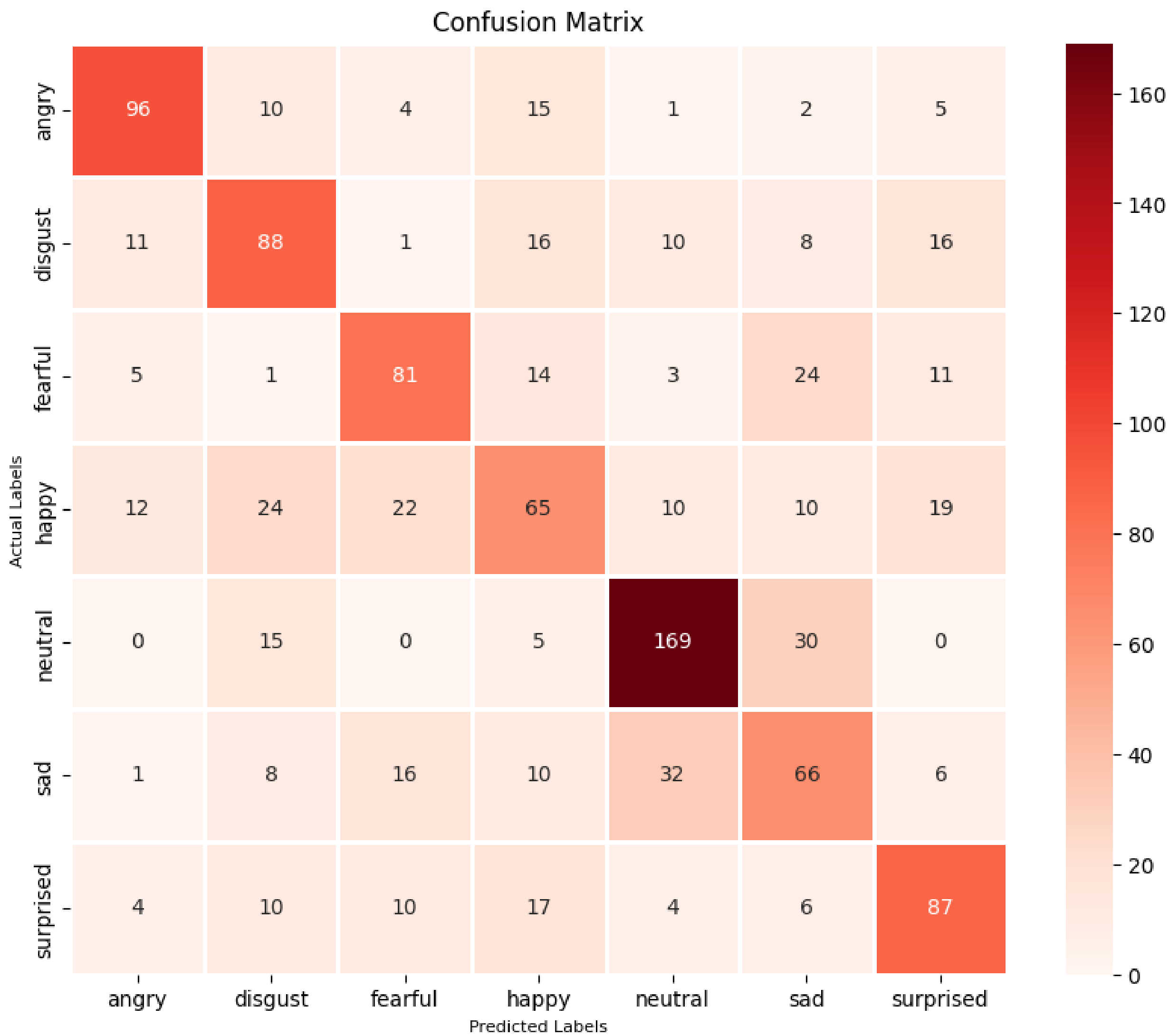
```
102/102 [==============================] - 1s 11ms/step - loss: 0.7148
- accuracy: 0.7355 - val_loss: 1.1270 - val_accuracy: 0.6204 - lr:
4.0000e-04
Epoch 89/100
102/102 [==============================] - 1s 13ms/step - loss: 0.6860
- accuracy: 0.7463 - val_loss: 1.1039 - val_accuracy: 0.6120 - lr:
1.6000e-04
Epoch 90/100
102/102 [==============================] - 1s 13ms/step - loss: 0.6814
- accuracy: 0.7429 - val_loss: 1.1143 - val_accuracy: 0.6204 - lr:
1.6000e-04
Epoch 91/100
102/102 [==============================] - 1s 13ms/step - loss: 0.6611
- accuracy: 0.7583 - val_loss: 1.1045 - val_accuracy: 0.6130 - lr:
1.6000e-04
Epoch 92/100
102/102 [==============================] - 2s 15ms/step - loss: 0.6755
- accuracy: 0.7506 - val_loss: 1.1335 - val_accuracy: 0.6093 - lr:
1.6000e-04
Epoch 93/100
102/102 [==============================] - 1s 14ms/step - loss: 0.6847
- accuracy: 0.7435 - val_loss: 1.1156 - val_accuracy: 0.6130 - lr:
1.6000e-04
Epoch 94/100
102/102 [==============================] - 1s 14ms/step - loss: 0.6626
- accuracy: 0.7546 - val_loss: 1.1180 - val_accuracy: 0.6065 - lr:
1.6000e-04
Epoch 95/100
102/102 [==============================] - 1s 15ms/step - loss: 0.6803
- accuracy: 0.7509 - val_loss: 1.1196 - val_accuracy: 0.6083 - lr:
1.6000e-04
Epoch 96/100
102/102 [==============================] - 2s 17ms/step - loss: 0.6623
- accuracy: 0.7691 - val_loss: 1.1124 - val_accuracy: 0.6093 - lr:
6.4000e-05
Epoch 97/100
102/102 [==============================] - 2s 17ms/step - loss: 0.6414
- accuracy: 0.7617 - val_loss: 1.1252 - val_accuracy: 0.6102 - lr:
6.4000e-05
Epoch 98/100
102/102 [==============================] - 1s 10ms/step - loss: 0.6621
- accuracy: 0.7583 - val_loss: 1.1200 - val_accuracy: 0.6139 - lr:
6.4000e-05
Epoch 99/100
102/102 [==============================] - 1s 10ms/step - loss: 0.6436
- accuracy: 0.7654 - val_loss: 1.1152 - val_accuracy: 0.6148 - lr:
6.4000e-05
Epoch 100/100
102/102 [==============================] - 1s 10ms/step - loss: 0.6553
```

```
- accuracy: 0.7586 - val_loss: 1.1228 - val_accuracy: 0.6037 - lr:
6.4000e-05
34/34 [==============================] - 0s 5ms/step - loss: 1.1228 -
accuracy: 0.6037
Accuracy of our model on test data :  60.370367765426636 %
```




```
34/34 [==============================] - 0s 5ms/step
```


{"summary":"{\n \"name\": \" print(classification_report(y_test, y_pred))\",\n \"rows\": 10,\n \"fields\": [\n   {\n   \"column\": \"Predicted Labels\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 6,\n   \"samples\": [\n   \"disgust\",\n   \"happy\",\n   \"angry\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Actual Labels\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 4,\n   \"samples\": [\n   \"neutral\",\n   \"angry\",\n   \"happy\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}




```
               precision    recall  f1-score   support

       angry       0.74      0.72      0.73       133
     disgust       0.56      0.59      0.58       150
     fearful       0.60      0.58      0.59       139
       happy       0.46      0.40      0.43       162
     neutral       0.74      0.77      0.75       219
         sad       0.45      0.47      0.46       139
   surprised       0.60      0.63      0.62       138

    accuracy                           0.60      1080
   macro avg       0.59      0.60      0.59      1080
weighted avg       0.60      0.60      0.60      1080

-----------------------------Adjusting Data for gender 
classification--------------------------------------------------------
```

```
-
(3240, 22) (3240, 2) (1080, 22) (1080, 2)
(3240, 22) (3240, 2) (1080, 22) (1080, 2)
(3240, 22, 1) (3240, 2) (1080, 22, 1) (1080, 2)
-------------------------------CREATING MODEL FOR PREDICTING
GENDER-----------------------------------------------------------------
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_3 (Conv1D)           (None, 22, 16)            96

 max_pooling1d_3 (MaxPoolin  (None, 11, 16)            0
 g1D)

 dropout_3 (Dropout)         (None, 11, 16)            0

 flatten_1 (Flatten)         (None, 176)               0

 dense_2 (Dense)             (None, 8)                 1416

 dropout_4 (Dropout)         (None, 8)                 0

 dense_3 (Dense)             (None, 2)                 18

=================================================================
Total params: 1530 (5.98 KB)
Trainable params: 1530 (5.98 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/100
102/102 [==============================] - 1s 5ms/step - loss: 0.6370
- accuracy: 0.6846 - val_loss: 0.4976 - val_accuracy: 0.8148 - lr:
0.0010
Epoch 2/100
102/102 [==============================] - 0s 3ms/step - loss: 0.4941
- accuracy: 0.7935 - val_loss: 0.4006 - val_accuracy: 0.8556 - lr:
0.0010
Epoch 3/100
102/102 [==============================] - 0s 3ms/step - loss: 0.4403
- accuracy: 0.8198 - val_loss: 0.3627 - val_accuracy: 0.8722 - lr:
0.0010
Epoch 4/100
102/102 [==============================] - 0s 4ms/step - loss: 0.4124
- accuracy: 0.8414 - val_loss: 0.3335 - val_accuracy: 0.8861 - lr:
0.0010
Epoch 5/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3738
- accuracy: 0.8556 - val_loss: 0.3016 - val_accuracy: 0.8972 - lr:
```

```
0.0010
Epoch 6/100
102/102 [==============================] - 0s 3ms/step - loss: 0.3550
- accuracy: 0.8636 - val_loss: 0.2987 - val_accuracy: 0.8991 - lr:
0.0010
Epoch 7/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3423
- accuracy: 0.8728 - val_loss: 0.2686 - val_accuracy: 0.9083 - lr:
0.0010
Epoch 8/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3341
- accuracy: 0.8784 - val_loss: 0.2493 - val_accuracy: 0.9093 - lr:
0.0010
Epoch 9/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3224
- accuracy: 0.8728 - val_loss: 0.2444 - val_accuracy: 0.9194 - lr:
0.0010
Epoch 10/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3050
- accuracy: 0.8849 - val_loss: 0.2339 - val_accuracy: 0.9250 - lr:
0.0010
Epoch 11/100
102/102 [==============================] - 0s 4ms/step - loss: 0.3084
- accuracy: 0.8836 - val_loss: 0.2214 - val_accuracy: 0.9222 - lr:
0.0010
Epoch 12/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2957
- accuracy: 0.8901 - val_loss: 0.2154 - val_accuracy: 0.9222 - lr:
0.0010
Epoch 13/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2953
- accuracy: 0.8840 - val_loss: 0.2079 - val_accuracy: 0.9269 - lr:
0.0010
Epoch 14/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2853
- accuracy: 0.8954 - val_loss: 0.1974 - val_accuracy: 0.9306 - lr:
0.0010
Epoch 15/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2681
- accuracy: 0.8969 - val_loss: 0.1931 - val_accuracy: 0.9315 - lr:
0.0010
Epoch 16/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2788
- accuracy: 0.8914 - val_loss: 0.1897 - val_accuracy: 0.9324 - lr:
0.0010
Epoch 17/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2664
- accuracy: 0.8988 - val_loss: 0.1967 - val_accuracy: 0.9333 - lr:
0.0010
```

```
Epoch 18/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2521
- accuracy: 0.9006 - val_loss: 0.1782 - val_accuracy: 0.9333 - lr:
0.0010
Epoch 19/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2628
- accuracy: 0.9068 - val_loss: 0.1812 - val_accuracy: 0.9417 - lr:
0.0010
Epoch 20/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2581
- accuracy: 0.9040 - val_loss: 0.1663 - val_accuracy: 0.9417 - lr:
0.0010
Epoch 21/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2551
- accuracy: 0.9040 - val_loss: 0.1651 - val_accuracy: 0.9444 - lr:
0.0010
Epoch 22/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2512
- accuracy: 0.9099 - val_loss: 0.1629 - val_accuracy: 0.9398 - lr:
0.0010
Epoch 23/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2499
- accuracy: 0.9102 - val_loss: 0.1605 - val_accuracy: 0.9444 - lr:
0.0010
Epoch 24/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2432
- accuracy: 0.9080 - val_loss: 0.1613 - val_accuracy: 0.9435 - lr:
0.0010
Epoch 25/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2460
- accuracy: 0.9083 - val_loss: 0.1521 - val_accuracy: 0.9481 - lr:
0.0010
Epoch 26/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2488
- accuracy: 0.9080 - val_loss: 0.1550 - val_accuracy: 0.9491 - lr:
0.0010
Epoch 27/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2423
- accuracy: 0.9114 - val_loss: 0.1541 - val_accuracy: 0.9491 - lr:
0.0010
Epoch 28/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2392
- accuracy: 0.9136 - val_loss: 0.1515 - val_accuracy: 0.9528 - lr:
0.0010
Epoch 29/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2412
- accuracy: 0.9093 - val_loss: 0.1486 - val_accuracy: 0.9491 - lr:
0.0010
Epoch 30/100
```

```
102/102 [==============================] - 0s 4ms/step - loss: 0.2275
- accuracy: 0.9160 - val_loss: 0.1503 - val_accuracy: 0.9509 - lr:
0.0010
Epoch 31/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2290
- accuracy: 0.9179 - val_loss: 0.1388 - val_accuracy: 0.9546 - lr:
0.0010
Epoch 32/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2307
- accuracy: 0.9139 - val_loss: 0.1431 - val_accuracy: 0.9583 - lr:
0.0010
Epoch 33/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2412
- accuracy: 0.9123 - val_loss: 0.1439 - val_accuracy: 0.9556 - lr:
0.0010
Epoch 34/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2250
- accuracy: 0.9133 - val_loss: 0.1395 - val_accuracy: 0.9556 - lr:
0.0010
Epoch 35/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2252
- accuracy: 0.9160 - val_loss: 0.1456 - val_accuracy: 0.9528 - lr:
0.0010
Epoch 36/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2374
- accuracy: 0.9086 - val_loss: 0.1387 - val_accuracy: 0.9565 - lr:
0.0010
Epoch 37/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2155
- accuracy: 0.9216 - val_loss: 0.1357 - val_accuracy: 0.9537 - lr:
0.0010
Epoch 38/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2184
- accuracy: 0.9176 - val_loss: 0.1313 - val_accuracy: 0.9611 - lr:
0.0010
Epoch 39/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2256
- accuracy: 0.9182 - val_loss: 0.1275 - val_accuracy: 0.9602 - lr:
0.0010
Epoch 40/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2193
- accuracy: 0.9262 - val_loss: 0.1295 - val_accuracy: 0.9583 - lr:
0.0010
Epoch 41/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2161
- accuracy: 0.9167 - val_loss: 0.1283 - val_accuracy: 0.9583 - lr:
0.0010
Epoch 42/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2155
```

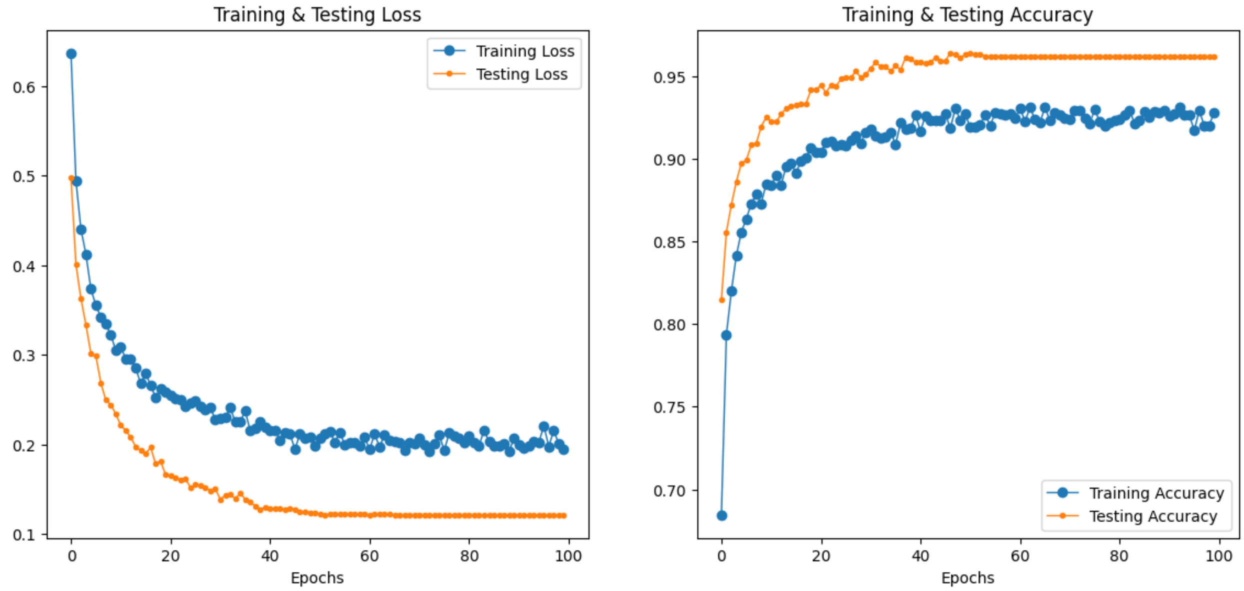
```
- accuracy: 0.9259 - val_loss: 0.1279 - val_accuracy: 0.9574 - lr:
4.0000e-04
Epoch 43/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2049
- accuracy: 0.9235 - val_loss: 0.1280 - val_accuracy: 0.9583 - lr:
4.0000e-04
Epoch 44/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2135
- accuracy: 0.9228 - val_loss: 0.1278 - val_accuracy: 0.9611 - lr:
4.0000e-04
Epoch 45/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2119
- accuracy: 0.9228 - val_loss: 0.1286 - val_accuracy: 0.9593 - lr:
4.0000e-04
Epoch 46/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1942
- accuracy: 0.9272 - val_loss: 0.1276 - val_accuracy: 0.9593 - lr:
4.0000e-04
Epoch 47/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2121
- accuracy: 0.9182 - val_loss: 0.1244 - val_accuracy: 0.9639 - lr:
4.0000e-04
Epoch 48/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2068
- accuracy: 0.9306 - val_loss: 0.1251 - val_accuracy: 0.9630 - lr:
4.0000e-04
Epoch 49/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2081
- accuracy: 0.9231 - val_loss: 0.1241 - val_accuracy: 0.9611 - lr:
4.0000e-04
Epoch 50/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1985
- accuracy: 0.9269 - val_loss: 0.1234 - val_accuracy: 0.9630 - lr:
4.0000e-04
Epoch 51/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2064
- accuracy: 0.9188 - val_loss: 0.1218 - val_accuracy: 0.9639 - lr:
1.6000e-04
Epoch 52/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2114
- accuracy: 0.9194 - val_loss: 0.1214 - val_accuracy: 0.9630 - lr:
1.6000e-04
Epoch 53/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2143
- accuracy: 0.9207 - val_loss: 0.1221 - val_accuracy: 0.9630 - lr:
1.6000e-04
Epoch 54/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2024
- accuracy: 0.9265 - val_loss: 0.1220 - val_accuracy: 0.9620 - lr:
```

```
1.6000e-04
Epoch 55/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2134
- accuracy: 0.9201 - val_loss: 0.1225 - val_accuracy: 0.9620 - lr:
6.4000e-05
Epoch 56/100
102/102 [==============================] - 1s 5ms/step - loss: 0.1999
- accuracy: 0.9275 - val_loss: 0.1222 - val_accuracy: 0.9620 - lr:
6.4000e-05
Epoch 57/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2018
- accuracy: 0.9272 - val_loss: 0.1220 - val_accuracy: 0.9620 - lr:
6.4000e-05
Epoch 58/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2021
- accuracy: 0.9262 - val_loss: 0.1218 - val_accuracy: 0.9620 - lr:
6.4000e-05
Epoch 59/100
102/102 [==============================] - 1s 5ms/step - loss: 0.1982
- accuracy: 0.9269 - val_loss: 0.1218 - val_accuracy: 0.9620 - lr:
2.5600e-05
Epoch 60/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2079
- accuracy: 0.9247 - val_loss: 0.1218 - val_accuracy: 0.9620 - lr:
2.5600e-05
Epoch 61/100
102/102 [==============================] - 1s 5ms/step - loss: 0.1944
- accuracy: 0.9306 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
2.5600e-05
Epoch 62/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2118
- accuracy: 0.9225 - val_loss: 0.1217 - val_accuracy: 0.9620 - lr:
2.5600e-05
Epoch 63/100
102/102 [==============================] - 1s 6ms/step - loss: 0.1970
- accuracy: 0.9309 - val_loss: 0.1217 - val_accuracy: 0.9620 - lr:
1.0240e-05
Epoch 64/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2111
- accuracy: 0.9238 - val_loss: 0.1217 - val_accuracy: 0.9620 - lr:
1.0240e-05
Epoch 65/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2050
- accuracy: 0.9216 - val_loss: 0.1217 - val_accuracy: 0.9620 - lr:
1.0240e-05
Epoch 66/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2035
- accuracy: 0.9312 - val_loss: 0.1216 - val_accuracy: 0.9620 - lr:
1.0240e-05
```

```
Epoch 67/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2026
- accuracy: 0.9228 - val_loss: 0.1216 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 68/100
102/102 [==============================] - 0s 3ms/step - loss: 0.1931
- accuracy: 0.9278 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 69/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2026
- accuracy: 0.9265 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 70/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2005
- accuracy: 0.9247 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 71/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2075
- accuracy: 0.9241 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 72/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1997
- accuracy: 0.9290 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
4.0960e-06
Epoch 73/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1928
- accuracy: 0.9290 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.6384e-06
Epoch 74/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2012
- accuracy: 0.9244 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.6384e-06
Epoch 75/100
102/102 [==============================] - 0s 3ms/step - loss: 0.2101
- accuracy: 0.9210 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.6384e-06
Epoch 76/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1933
- accuracy: 0.9299 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.6384e-06
Epoch 77/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2127
- accuracy: 0.9222 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.6384e-06
Epoch 78/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2092
- accuracy: 0.9201 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 79/100
```

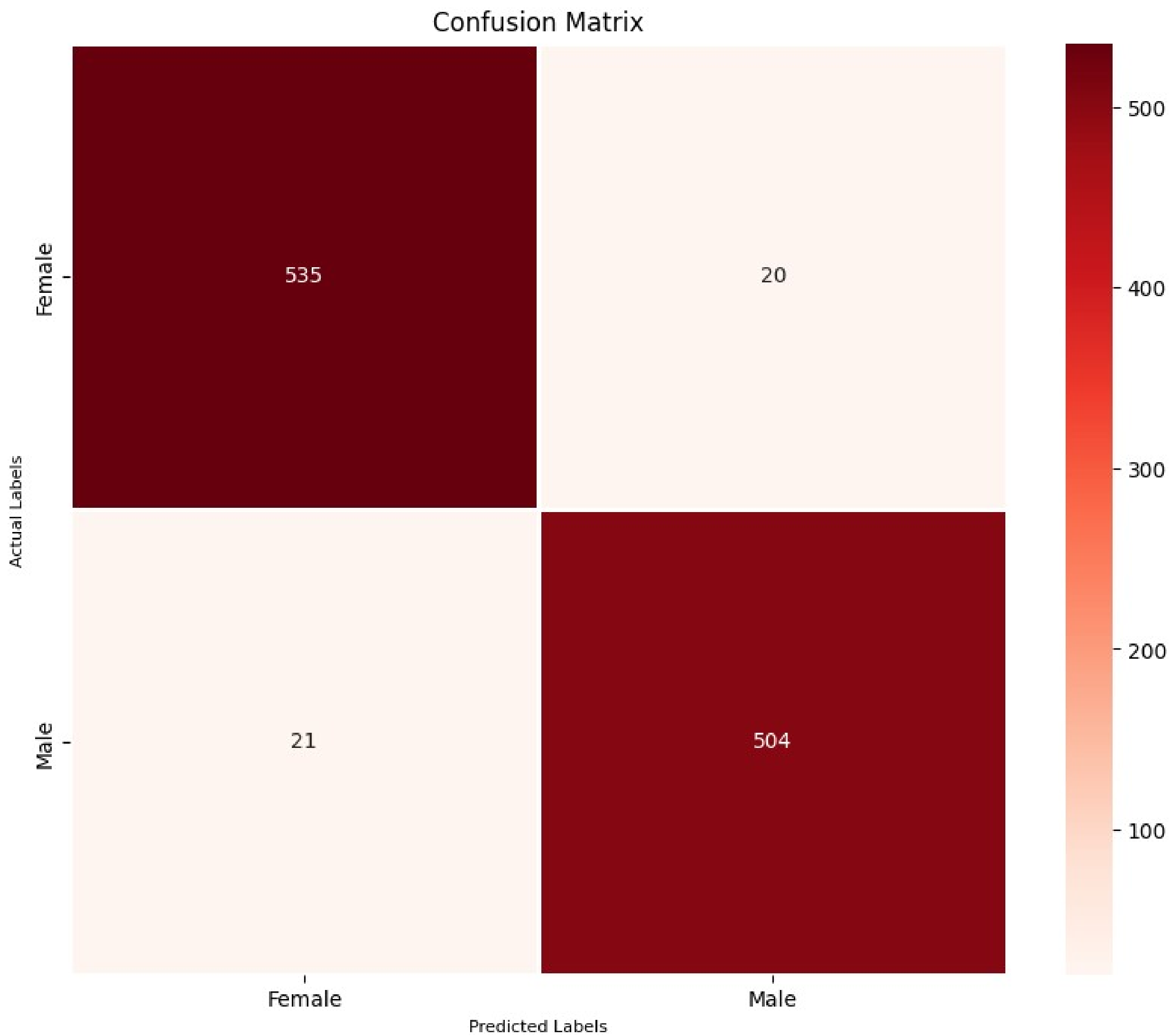
```
102/102 [==============================] - 0s 4ms/step - loss: 0.2067
- accuracy: 0.9216 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 80/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2026
- accuracy: 0.9231 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 81/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2095
- accuracy: 0.9241 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 82/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2024
- accuracy: 0.9265 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 83/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1988
- accuracy: 0.9293 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 84/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2160
- accuracy: 0.9213 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 85/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2028
- accuracy: 0.9228 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 86/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1985
- accuracy: 0.9287 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 87/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1980
- accuracy: 0.9250 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 88/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2003
- accuracy: 0.9284 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 89/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1918
- accuracy: 0.9278 - val_loss: 0.1215 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 90/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2066
- accuracy: 0.9290 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 91/100
102/102 [==============================] - 0s 5ms/step - loss: 0.1999
```

```
- accuracy: 0.9256 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 92/100
102/102 [==============================] - 1s 5ms/step - loss: 0.1954
- accuracy: 0.9269 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 93/100
102/102 [==============================] - 1s 6ms/step - loss: 0.1988
- accuracy: 0.9309 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 94/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2032
- accuracy: 0.9265 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 95/100
102/102 [==============================] - 1s 5ms/step - loss: 0.2021
- accuracy: 0.9265 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 96/100
102/102 [==============================] - 1s 6ms/step - loss: 0.2208
- accuracy: 0.9170 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 97/100
102/102 [==============================] - 1s 6ms/step - loss: 0.1977
- accuracy: 0.9290 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 98/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2159
- accuracy: 0.9201 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 99/100
102/102 [==============================] - 0s 4ms/step - loss: 0.2003
- accuracy: 0.9201 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
Epoch 100/100
102/102 [==============================] - 0s 4ms/step - loss: 0.1950
- accuracy: 0.9281 - val_loss: 0.1214 - val_accuracy: 0.9620 - lr:
1.0000e-06
34/34 [==============================] - 0s 2ms/step - loss: 0.1214 -
accuracy: 0.9620
Accuracy of our model on test data :  96.20370268821716 %
```




```
34/34 [==============================] - 0s 2ms/step
```

{"summary":"{\n \"name\": \" print(classification_report(y_test, y_pred))\",\n \"rows\": 10,\n \"fields\": [\n   {\n   \"column\": \"Predicted Labels\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Female\",\n   \"Male\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Actual Labels\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 2,\n   \"samples\": [\n   \"Female\",\n   \"Male\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}




```
              precision    recall  f1-score   support

      Female       0.96      0.96      0.96       555
        Male       0.96      0.96      0.96       525

    accuracy                           0.96      1080
   macro avg       0.96      0.96      0.96      1080
weighted avg       0.96      0.96      0.96      1080

-----------------------------Adjusting Data for gender and emotion
classification
together-------------------------------------------------------------
(3240, 22) (3240, 14) (1080, 22) (1080, 14)
(3240, 22) (3240, 14) (1080, 22) (1080, 14)
(3240, 22, 1) (3240, 14) (1080, 22, 1) (1080, 14)
-------------------------------CREATING MODEL FOR PREDICTING GENDER
```

```
AND EMOTION
TOGETHER--------------------------------------------------------------------
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d_4 (Conv1D)           (None, 22, 256)           1536

 max_pooling1d_4 (MaxPoolin  (None, 11, 256)           0
 g1D)

 conv1d_5 (Conv1D)           (None, 11, 128)           163968

 max_pooling1d_5 (MaxPoolin  (None, 6, 128)            0
 g1D)

 dropout_5 (Dropout)         (None, 6, 128)            0

 conv1d_6 (Conv1D)           (None, 6, 64)             41024

 max_pooling1d_6 (MaxPoolin  (None, 3, 64)             0
 g1D)

 dropout_6 (Dropout)         (None, 3, 64)             0

 conv1d_7 (Conv1D)           (None, 3, 32)             10272

 max_pooling1d_7 (MaxPoolin  (None, 2, 32)             0
 g1D)

 dropout_7 (Dropout)         (None, 2, 32)             0

 flatten_2 (Flatten)         (None, 64)                0

 dense_4 (Dense)             (None, 16)                1040

 dropout_8 (Dropout)         (None, 16)                0

 dense_5 (Dense)             (None, 14)                238

=================================================================
Total params: 218078 (851.87 KB)
Trainable params: 218078 (851.87 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/100
102/102 [==============================] - 5s 32ms/step - loss: 2.6111
- accuracy: 0.0978 - val_loss: 2.5295 - val_accuracy: 0.1398 - lr:
0.0010
```

```
Epoch 2/100
102/102 [==============================] - 3s 33ms/step - loss: 2.5004
- accuracy: 0.1432 - val_loss: 2.3371 - val_accuracy: 0.2241 - lr:
0.0010
Epoch 3/100
102/102 [==============================] - 4s 44ms/step - loss: 2.3435
- accuracy: 0.1914 - val_loss: 2.2193 - val_accuracy: 0.2454 - lr:
0.0010
Epoch 4/100
102/102 [==============================] - 3s 31ms/step - loss: 2.2879
- accuracy: 0.2145 - val_loss: 2.1549 - val_accuracy: 0.2537 - lr:
0.0010
Epoch 5/100
102/102 [==============================] - 3s 31ms/step - loss: 2.2148
- accuracy: 0.2117 - val_loss: 2.0517 - val_accuracy: 0.2676 - lr:
0.0010
Epoch 6/100
102/102 [==============================] - 3s 31ms/step - loss: 2.1412
- accuracy: 0.2247 - val_loss: 2.0151 - val_accuracy: 0.2630 - lr:
0.0010
Epoch 7/100
102/102 [==============================] - 4s 44ms/step - loss: 2.0823
- accuracy: 0.2327 - val_loss: 1.9306 - val_accuracy: 0.2944 - lr:
0.0010
Epoch 8/100
102/102 [==============================] - 3s 32ms/step - loss: 2.0324
- accuracy: 0.2497 - val_loss: 1.8702 - val_accuracy: 0.3000 - lr:
0.0010
Epoch 9/100
102/102 [==============================] - 3s 31ms/step - loss: 1.9575
- accuracy: 0.2633 - val_loss: 1.8177 - val_accuracy: 0.3278 - lr:
0.0010
Epoch 10/100
102/102 [==============================] - 3s 30ms/step - loss: 1.9236
- accuracy: 0.2873 - val_loss: 1.8222 - val_accuracy: 0.3296 - lr:
0.0010
Epoch 11/100
102/102 [==============================] - 4s 39ms/step - loss: 1.8913
- accuracy: 0.2969 - val_loss: 1.7668 - val_accuracy: 0.3352 - lr:
0.0010
Epoch 12/100
102/102 [==============================] - 4s 37ms/step - loss: 1.8612
- accuracy: 0.3145 - val_loss: 1.7848 - val_accuracy: 0.3491 - lr:
0.0010
Epoch 13/100
102/102 [==============================] - 3s 31ms/step - loss: 1.8242
- accuracy: 0.3182 - val_loss: 1.7078 - val_accuracy: 0.3528 - lr:
0.0010
Epoch 14/100
```

```
102/102 [==============================] - 3s 31ms/step - loss: 1.8079
- accuracy: 0.3160 - val_loss: 1.6631 - val_accuracy: 0.3861 - lr:
0.0010
Epoch 15/100
102/102 [==============================] - 4s 36ms/step - loss: 1.7490
- accuracy: 0.3333 - val_loss: 1.6231 - val_accuracy: 0.3796 - lr:
0.0010
Epoch 16/100
102/102 [==============================] - 4s 41ms/step - loss: 1.7439
- accuracy: 0.3327 - val_loss: 1.6233 - val_accuracy: 0.3796 - lr:
0.0010
Epoch 17/100
102/102 [==============================] - 3s 31ms/step - loss: 1.7137
- accuracy: 0.3522 - val_loss: 1.5837 - val_accuracy: 0.3759 - lr:
0.0010
Epoch 18/100
102/102 [==============================] - 3s 31ms/step - loss: 1.7297
- accuracy: 0.3642 - val_loss: 1.5863 - val_accuracy: 0.3833 - lr:
0.0010
Epoch 19/100
102/102 [==============================] - 3s 32ms/step - loss: 1.6729
- accuracy: 0.3701 - val_loss: 1.6045 - val_accuracy: 0.3889 - lr:
0.0010
Epoch 20/100
102/102 [==============================] - 5s 45ms/step - loss: 1.6698
- accuracy: 0.3710 - val_loss: 1.5769 - val_accuracy: 0.4111 - lr:
0.0010
Epoch 21/100
102/102 [==============================] - 3s 31ms/step - loss: 1.6435
- accuracy: 0.3682 - val_loss: 1.5542 - val_accuracy: 0.3954 - lr:
0.0010
Epoch 22/100
102/102 [==============================] - 3s 31ms/step - loss: 1.6582
- accuracy: 0.3809 - val_loss: 1.5498 - val_accuracy: 0.4028 - lr:
0.0010
Epoch 23/100
102/102 [==============================] - 3s 31ms/step - loss: 1.6357
- accuracy: 0.3762 - val_loss: 1.5696 - val_accuracy: 0.4241 - lr:
0.0010
Epoch 24/100
102/102 [==============================] - 4s 44ms/step - loss: 1.6136
- accuracy: 0.3799 - val_loss: 1.5431 - val_accuracy: 0.3981 - lr:
0.0010
Epoch 25/100
102/102 [==============================] - 3s 31ms/step - loss: 1.6036
- accuracy: 0.4003 - val_loss: 1.5151 - val_accuracy: 0.4167 - lr:
0.0010
Epoch 26/100
102/102 [==============================] - 3s 31ms/step - loss: 1.6234
```

```
- accuracy: 0.3849 - val_loss: 1.5548 - val_accuracy: 0.4194 - lr:
0.0010
Epoch 27/100
102/102 [==============================] - 3s 30ms/step - loss: 1.6253
- accuracy: 0.3793 - val_loss: 1.5370 - val_accuracy: 0.4120 - lr:
0.0010
Epoch 28/100
102/102 [==============================] - 4s 38ms/step - loss: 1.5899
- accuracy: 0.3997 - val_loss: 1.5565 - val_accuracy: 0.4361 - lr:
0.0010
Epoch 29/100
102/102 [==============================] - 4s 38ms/step - loss: 1.5790
- accuracy: 0.4105 - val_loss: 1.5158 - val_accuracy: 0.4324 - lr:
0.0010
Epoch 30/100
102/102 [==============================] - 3s 31ms/step - loss: 1.5665
- accuracy: 0.4090 - val_loss: 1.5166 - val_accuracy: 0.4204 - lr:
0.0010
Epoch 31/100
102/102 [==============================] - 3s 31ms/step - loss: 1.5534
- accuracy: 0.4108 - val_loss: 1.5337 - val_accuracy: 0.4120 - lr:
0.0010
Epoch 32/100
102/102 [==============================] - 4s 35ms/step - loss: 1.5289
- accuracy: 0.4151 - val_loss: 1.4713 - val_accuracy: 0.4509 - lr:
0.0010
Epoch 33/100
102/102 [==============================] - 4s 42ms/step - loss: 1.4938
- accuracy: 0.4367 - val_loss: 1.5462 - val_accuracy: 0.4278 - lr:
0.0010
Epoch 34/100
102/102 [==============================] - 3s 31ms/step - loss: 1.4957
- accuracy: 0.4306 - val_loss: 1.4350 - val_accuracy: 0.4528 - lr:
0.0010
Epoch 35/100
102/102 [==============================] - 3s 31ms/step - loss: 1.5446
- accuracy: 0.4241 - val_loss: 1.4423 - val_accuracy: 0.4657 - lr:
0.0010
Epoch 36/100
102/102 [==============================] - 3s 32ms/step - loss: 1.4947
- accuracy: 0.4340 - val_loss: 1.4597 - val_accuracy: 0.4519 - lr:
0.0010
Epoch 37/100
102/102 [==============================] - 4s 43ms/step - loss: 1.4832
- accuracy: 0.4293 - val_loss: 1.4290 - val_accuracy: 0.4704 - lr:
0.0010
Epoch 38/100
102/102 [==============================] - 3s 31ms/step - loss: 1.4762
- accuracy: 0.4503 - val_loss: 1.4567 - val_accuracy: 0.4417 - lr:
```

```
0.0010
Epoch 39/100
102/102 [==============================] - 3s 30ms/step - loss: 1.4614
- accuracy: 0.4478 - val_loss: 1.4890 - val_accuracy: 0.4620 - lr:
0.0010
Epoch 40/100
102/102 [==============================] - 3s 31ms/step - loss: 1.4394
- accuracy: 0.4497 - val_loss: 1.4258 - val_accuracy: 0.4778 - lr:
0.0010
Epoch 41/100
102/102 [==============================] - 4s 42ms/step - loss: 1.4232
- accuracy: 0.4654 - val_loss: 1.3729 - val_accuracy: 0.4889 - lr:
0.0010
Epoch 42/100
102/102 [==============================] - 4s 34ms/step - loss: 1.4049
- accuracy: 0.4552 - val_loss: 1.4260 - val_accuracy: 0.4676 - lr:
0.0010
Epoch 43/100
102/102 [==============================] - 3s 31ms/step - loss: 1.4050
- accuracy: 0.4602 - val_loss: 1.3933 - val_accuracy: 0.4778 - lr:
0.0010
Epoch 44/100
102/102 [==============================] - 3s 31ms/step - loss: 1.4010
- accuracy: 0.4772 - val_loss: 1.4364 - val_accuracy: 0.4630 - lr:
0.0010
Epoch 45/100
102/102 [==============================] - 4s 37ms/step - loss: 1.3680
- accuracy: 0.4784 - val_loss: 1.3809 - val_accuracy: 0.4769 - lr:
0.0010
Epoch 46/100
102/102 [==============================] - 4s 39ms/step - loss: 1.3820
- accuracy: 0.4793 - val_loss: 1.3884 - val_accuracy: 0.4907 - lr:
0.0010
Epoch 47/100
102/102 [==============================] - 3s 30ms/step - loss: 1.3861
- accuracy: 0.4769 - val_loss: 1.3682 - val_accuracy: 0.4972 - lr:
0.0010
Epoch 48/100
102/102 [==============================] - 3s 31ms/step - loss: 1.3369
- accuracy: 0.4775 - val_loss: 1.3697 - val_accuracy: 0.5102 - lr:
0.0010
Epoch 49/100
102/102 [==============================] - 3s 34ms/step - loss: 1.3774
- accuracy: 0.4799 - val_loss: 1.3668 - val_accuracy: 0.5000 - lr:
0.0010
Epoch 50/100
102/102 [==============================] - 4s 43ms/step - loss: 1.3219
- accuracy: 0.4969 - val_loss: 1.3236 - val_accuracy: 0.5028 - lr:
0.0010
```

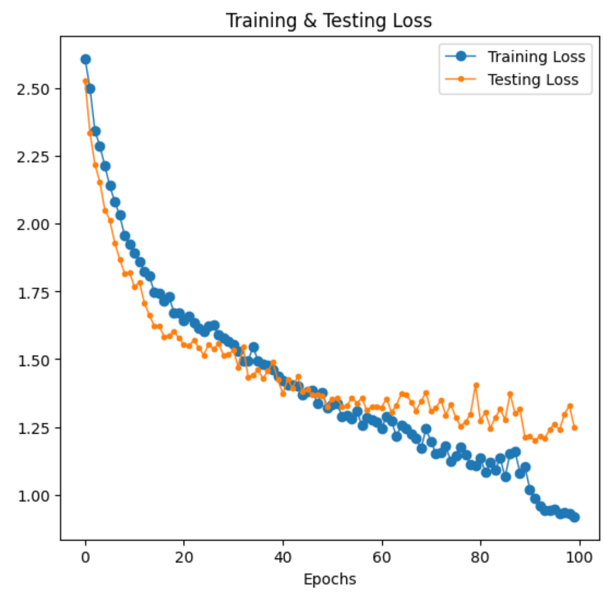
```
Epoch 51/100
102/102 [==============================] - 3s 31ms/step - loss: 1.3310
- accuracy: 0.4991 - val_loss: 1.3538 - val_accuracy: 0.5093 - lr:
0.0010
Epoch 52/100
102/102 [==============================] - 3s 30ms/step - loss: 1.3357
- accuracy: 0.4981 - val_loss: 1.3556 - val_accuracy: 0.5102 - lr:
0.0010
Epoch 53/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2877
- accuracy: 0.5077 - val_loss: 1.3251 - val_accuracy: 0.5074 - lr:
0.0010
Epoch 54/100
102/102 [==============================] - 4s 44ms/step - loss: 1.2928
- accuracy: 0.5012 - val_loss: 1.3279 - val_accuracy: 0.5074 - lr:
0.0010
Epoch 55/100
102/102 [==============================] - 3s 32ms/step - loss: 1.2789
- accuracy: 0.5191 - val_loss: 1.3585 - val_accuracy: 0.5037 - lr:
0.0010
Epoch 56/100
102/102 [==============================] - 3s 31ms/step - loss: 1.3095
- accuracy: 0.5071 - val_loss: 1.3382 - val_accuracy: 0.5074 - lr:
0.0010
Epoch 57/100
102/102 [==============================] - 3s 30ms/step - loss: 1.2549
- accuracy: 0.5265 - val_loss: 1.3558 - val_accuracy: 0.4926 - lr:
0.0010
Epoch 58/100
102/102 [==============================] - 4s 39ms/step - loss: 1.2850
- accuracy: 0.5170 - val_loss: 1.3119 - val_accuracy: 0.5028 - lr:
0.0010
Epoch 59/100
102/102 [==============================] - 4s 37ms/step - loss: 1.2781
- accuracy: 0.5238 - val_loss: 1.3244 - val_accuracy: 0.4963 - lr:
0.0010
Epoch 60/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2680
- accuracy: 0.5309 - val_loss: 1.3249 - val_accuracy: 0.5056 - lr:
0.0010
Epoch 61/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2441
- accuracy: 0.5284 - val_loss: 1.3224 - val_accuracy: 0.5037 - lr:
0.0010
Epoch 62/100
102/102 [==============================] - 4s 36ms/step - loss: 1.2875
- accuracy: 0.5191 - val_loss: 1.3541 - val_accuracy: 0.4963 - lr:
0.0010
Epoch 63/100
```

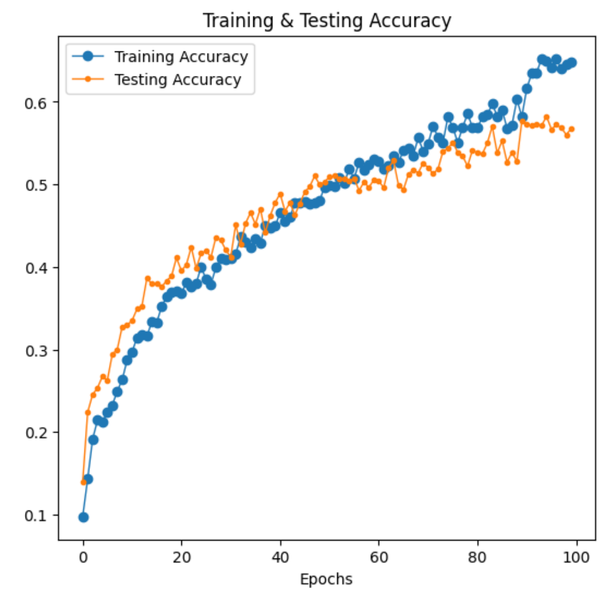
```
102/102 [==============================] - 4s 41ms/step - loss: 1.2739
- accuracy: 0.5225 - val_loss: 1.3044 - val_accuracy: 0.5204 - lr:
0.0010
Epoch 64/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2142
- accuracy: 0.5349 - val_loss: 1.3286 - val_accuracy: 0.5287 - lr:
0.0010
Epoch 65/100
102/102 [==============================] - 3s 30ms/step - loss: 1.2569
- accuracy: 0.5265 - val_loss: 1.3743 - val_accuracy: 0.4991 - lr:
0.0010
Epoch 66/100
102/102 [==============================] - 3s 32ms/step - loss: 1.2454
- accuracy: 0.5407 - val_loss: 1.3677 - val_accuracy: 0.4935 - lr:
0.0010
Epoch 67/100
102/102 [==============================] - 5s 44ms/step - loss: 1.2252
- accuracy: 0.5432 - val_loss: 1.3420 - val_accuracy: 0.5120 - lr:
0.0010
Epoch 68/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2064
- accuracy: 0.5352 - val_loss: 1.3096 - val_accuracy: 0.5176 - lr:
0.0010
Epoch 69/100
102/102 [==============================] - 3s 30ms/step - loss: 1.1712
- accuracy: 0.5565 - val_loss: 1.3462 - val_accuracy: 0.5139 - lr:
0.0010
Epoch 70/100
102/102 [==============================] - 3s 31ms/step - loss: 1.2426
- accuracy: 0.5404 - val_loss: 1.3768 - val_accuracy: 0.5259 - lr:
0.0010
Epoch 71/100
102/102 [==============================] - 4s 42ms/step - loss: 1.1959
- accuracy: 0.5494 - val_loss: 1.3065 - val_accuracy: 0.5204 - lr:
0.0010
Epoch 72/100
102/102 [==============================] - 3s 34ms/step - loss: 1.1510
- accuracy: 0.5707 - val_loss: 1.3205 - val_accuracy: 0.5130 - lr:
0.0010
Epoch 73/100
102/102 [==============================] - 3s 31ms/step - loss: 1.1562
- accuracy: 0.5574 - val_loss: 1.3491 - val_accuracy: 0.5185 - lr:
0.0010
Epoch 74/100
102/102 [==============================] - 3s 30ms/step - loss: 1.1782
- accuracy: 0.5500 - val_loss: 1.2936 - val_accuracy: 0.5398 - lr:
0.0010
Epoch 75/100
102/102 [==============================] - 4s 37ms/step - loss: 1.1226
```

```
- accuracy: 0.5815 - val_loss: 1.3339 - val_accuracy: 0.5435 - lr:
0.0010
Epoch 76/100
102/102 [==============================] - 4s 39ms/step - loss: 1.1420
- accuracy: 0.5694 - val_loss: 1.2856 - val_accuracy: 0.5509 - lr:
0.0010
Epoch 77/100
102/102 [==============================] - 3s 30ms/step - loss: 1.1767
- accuracy: 0.5506 - val_loss: 1.2522 - val_accuracy: 0.5389 - lr:
0.0010
Epoch 78/100
102/102 [==============================] - 3s 29ms/step - loss: 1.1476
- accuracy: 0.5691 - val_loss: 1.2690 - val_accuracy: 0.5343 - lr:
0.0010
Epoch 79/100
102/102 [==============================] - 3s 31ms/step - loss: 1.1102
- accuracy: 0.5855 - val_loss: 1.2977 - val_accuracy: 0.5222 - lr:
0.0010
Epoch 80/100
102/102 [==============================] - 5s 44ms/step - loss: 1.1065
- accuracy: 0.5691 - val_loss: 1.4042 - val_accuracy: 0.5417 - lr:
0.0010
Epoch 81/100
102/102 [==============================] - 3s 30ms/step - loss: 1.1352
- accuracy: 0.5688 - val_loss: 1.2735 - val_accuracy: 0.5380 - lr:
0.0010
Epoch 82/100
102/102 [==============================] - 3s 30ms/step - loss: 1.0813
- accuracy: 0.5815 - val_loss: 1.3046 - val_accuracy: 0.5370 - lr:
0.0010
Epoch 83/100
102/102 [==============================] - 3s 31ms/step - loss: 1.1171
- accuracy: 0.5843 - val_loss: 1.2456 - val_accuracy: 0.5500 - lr:
0.0010
Epoch 84/100
102/102 [==============================] - 4s 40ms/step - loss: 1.0918
- accuracy: 0.5975 - val_loss: 1.2832 - val_accuracy: 0.5704 - lr:
0.0010
Epoch 85/100
102/102 [==============================] - 4s 37ms/step - loss: 1.1344
- accuracy: 0.5827 - val_loss: 1.3157 - val_accuracy: 0.5380 - lr:
0.0010
Epoch 86/100
102/102 [==============================] - 3s 31ms/step - loss: 1.0648
- accuracy: 0.5901 - val_loss: 1.2763 - val_accuracy: 0.5528 - lr:
0.0010
Epoch 87/100
102/102 [==============================] - 3s 31ms/step - loss: 1.1527
- accuracy: 0.5682 - val_loss: 1.3723 - val_accuracy: 0.5269 - lr:
```

```
0.0010
Epoch 88/100
102/102 [==============================] - 4s 37ms/step - loss: 1.1605
- accuracy: 0.5719 - val_loss: 1.3002 - val_accuracy: 0.5389 - lr:
0.0010
Epoch 89/100
102/102 [==============================] - 4s 40ms/step - loss: 1.0799
- accuracy: 0.6028 - val_loss: 1.3154 - val_accuracy: 0.5278 - lr:
0.0010
Epoch 90/100
102/102 [==============================] - 3s 31ms/step - loss: 1.1039
- accuracy: 0.5821 - val_loss: 1.2106 - val_accuracy: 0.5769 - lr:
0.0010
Epoch 91/100
102/102 [==============================] - 3s 30ms/step - loss: 1.0198
- accuracy: 0.6160 - val_loss: 1.2176 - val_accuracy: 0.5722 - lr:
4.0000e-04
Epoch 92/100
102/102 [==============================] - 3s 32ms/step - loss: 0.9846
- accuracy: 0.6346 - val_loss: 1.1980 - val_accuracy: 0.5713 - lr:
4.0000e-04
Epoch 93/100
102/102 [==============================] - 4s 44ms/step - loss: 0.9560
- accuracy: 0.6352 - val_loss: 1.2154 - val_accuracy: 0.5731 - lr:
4.0000e-04
Epoch 94/100
102/102 [==============================] - 3s 31ms/step - loss: 0.9426
- accuracy: 0.6525 - val_loss: 1.2069 - val_accuracy: 0.5713 - lr:
4.0000e-04
Epoch 95/100
102/102 [==============================] - 3s 31ms/step - loss: 0.9428
- accuracy: 0.6491 - val_loss: 1.2395 - val_accuracy: 0.5824 - lr:
4.0000e-04
Epoch 96/100
102/102 [==============================] - 3s 31ms/step - loss: 0.9453
- accuracy: 0.6414 - val_loss: 1.2608 - val_accuracy: 0.5667 - lr:
4.0000e-04
Epoch 97/100
102/102 [==============================] - 4s 42ms/step - loss: 0.9300
- accuracy: 0.6515 - val_loss: 1.2391 - val_accuracy: 0.5722 - lr:
4.0000e-04
Epoch 98/100
102/102 [==============================] - 4s 35ms/step - loss: 0.9344
- accuracy: 0.6404 - val_loss: 1.2961 - val_accuracy: 0.5685 - lr:
4.0000e-04
Epoch 99/100
102/102 [==============================] - 3s 31ms/step - loss: 0.9285
- accuracy: 0.6457 - val_loss: 1.3300 - val_accuracy: 0.5602 - lr:
4.0000e-04
```

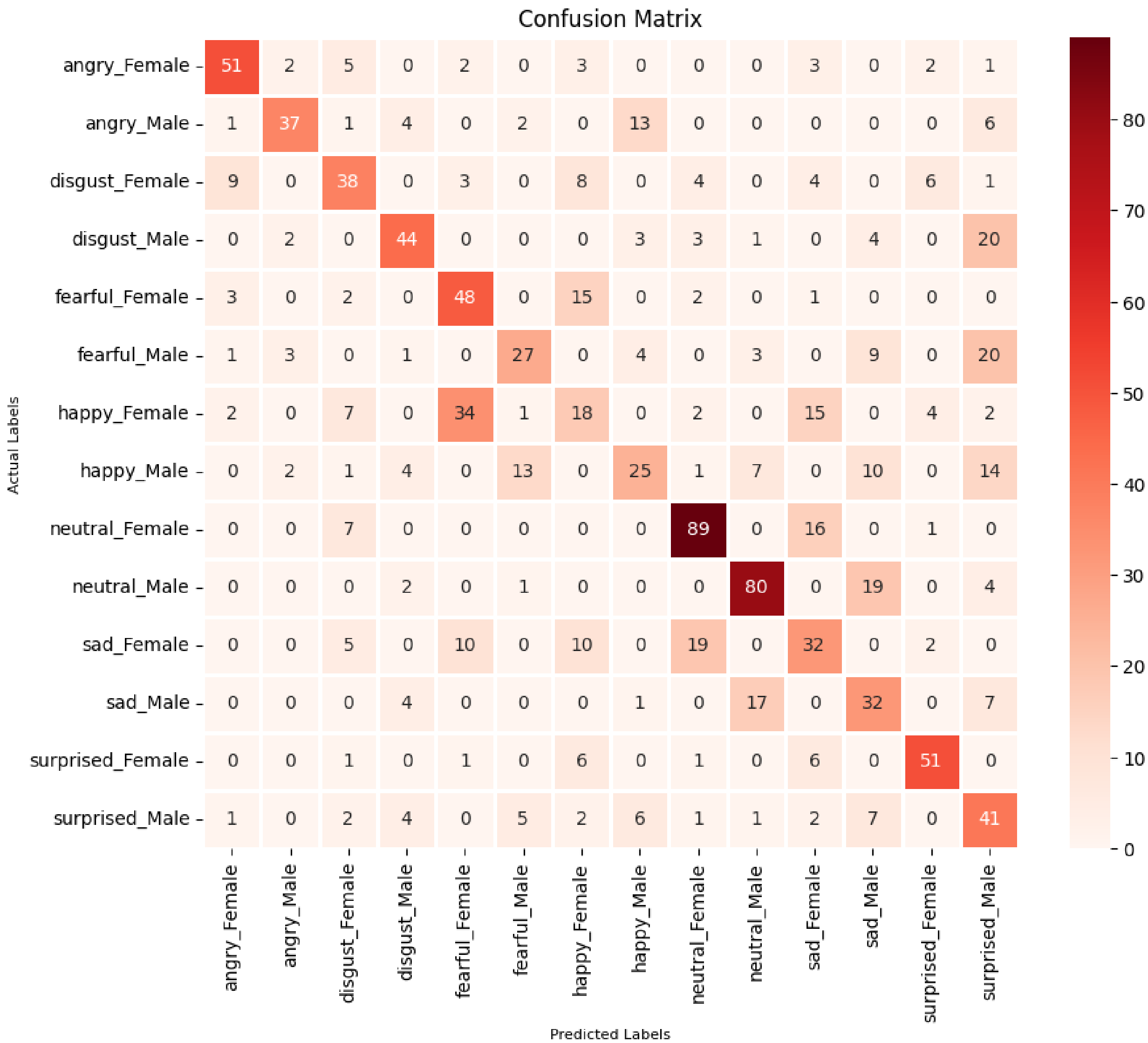
```
Epoch 100/100
102/102 [==============================] - 3s 30ms/step - loss: 0.9188
- accuracy: 0.6485 - val_loss: 1.2480 - val_accuracy: 0.5676 - lr:
4.0000e-04
34/34 [==============================] - 0s 8ms/step - loss: 1.2480 -
accuracy: 0.5676
Accuracy of our model on test data :  56.75926208496094 %
```






```
34/34 [==============================] - 0s 6ms/step
```

{"summary":"{\n \"name\": \" print(classification_report(y_test, y_pred))\",\n \"rows\": 10,\n \"fields\": [\n   {\n   \"column\": \"Predicted Labels\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 6,\n   \"samples\": [\n   \"happy_Male\",\n   \"surprised_Male\",\n   \"sad_Female\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"Actual Labels\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 6,\n   \"samples\": [\n   \"happy_Male\",\n   \"neutral_Male\",\n   \"sad_Female\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe"}



	precision	recall	f1-score	support
angry_Female	0.75	0.74	0.74	69
angry_Male	0.80	0.58	0.67	64
disgust_Female	0.55	0.52	0.54	73
disgust_Male	0.70	0.57	0.63	77
fearful_Female	0.49	0.68	0.57	71
fearful_Male	0.55	0.40	0.46	68
happy_Female	0.29	0.21	0.24	85
happy_Male	0.48	0.32	0.39	77
neutral_Female	0.73	0.79	0.76	113
neutral_Male	0.73	0.75	0.74	106
sad_Female	0.41	0.41	0.41	78
sad_Male	0.40	0.52	0.45	61
surprised_Female	0.77	0.77	0.77	66

```
  surprised_Male       0.35      0.57      0.44        72

        accuracy                           0.57      1080
       macro avg       0.57      0.56      0.56      1080
    weighted avg       0.58      0.57      0.57      1080
```