TMP - 2023 - 24 - 065

Multi-Model Approach to Recommend Personalized Music Playlist

Our Team

- Supervisor - Mr. Thusithanjana Thilakarathne
- Co - Supervisor - Dr. Dharshana Kasthurirathne
- External Supervisor - Mrs. Kusumanjalee Thilakarathne

Student ID	Student Name
IT20665616	Sumanasekara H. P.
IT20610852	Fernando M. P. T. K.
IT20665852	Gunasekara C. M.
IT20667078	Dhananjaya W. K. S.

Introduction

- A personalised music and song recommendation system aims to provide users with customised music suggestions based on few facts like emotion, age, gender, surroundings etc.

- The nature of the solution revolves around classifying user data, advanced algorithms like CNN/RNN, and machine-learning techniques to deliver a tailored music experience.

Main Objective

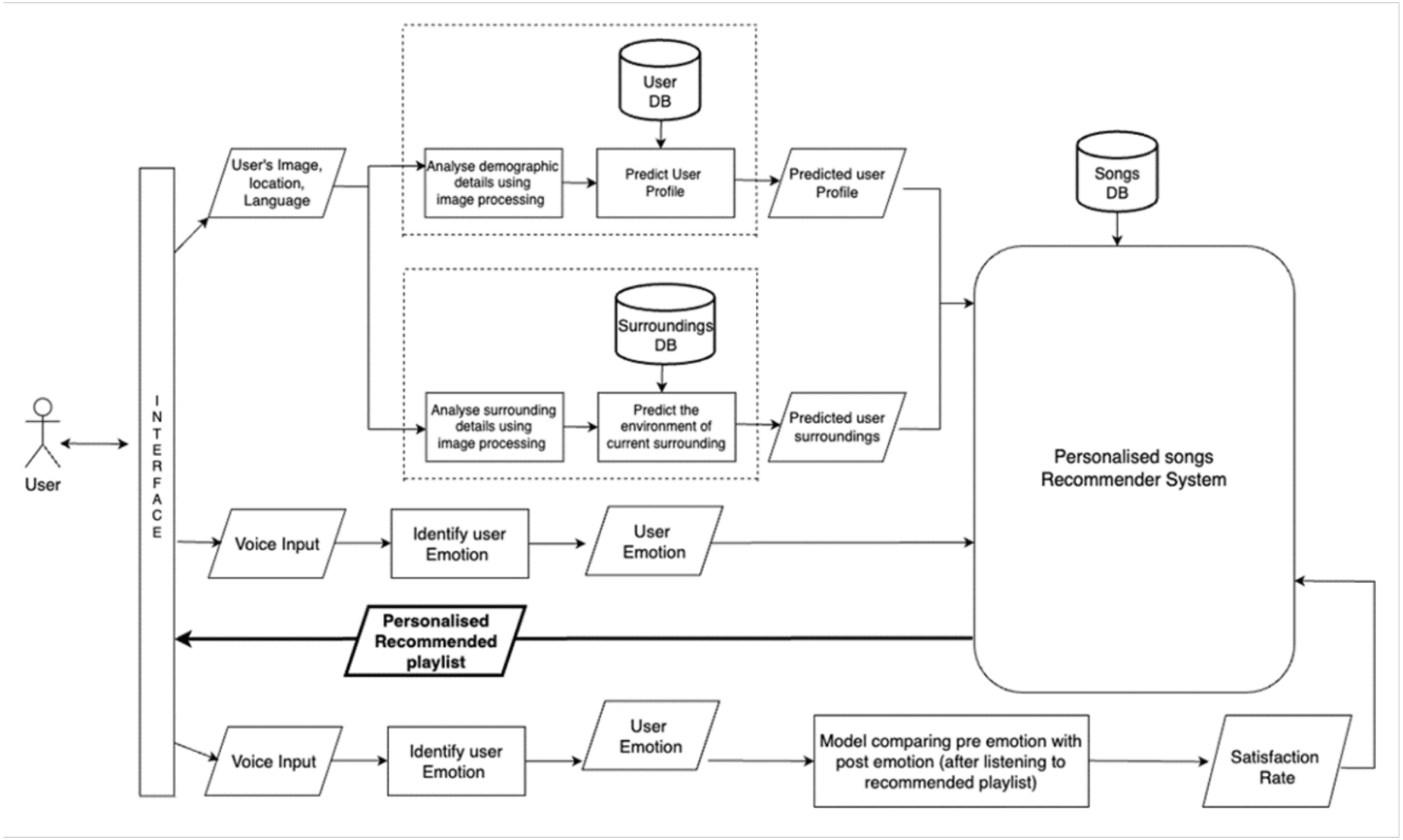
- To give user a personalised play list according to his or her current mood / emotion, personal data like age, gender, and current surroundings using voice recognition and image processing mechanisms.

- To assess the impact for the user after listening the recommended playlist.

Sub Objectives

1. **Analyse a user specific Profile** - This component tackles the "Cold-Start problem" in music recommendation by using selfie images for personalised profile analysis and better suggestions.
2. **Surroundings of the user** - Context-aware music recommendations consider user surroundings (time, setting, noise) for a personalised, engaging music streaming experience.
3. **Emotions using vocal responses** - improving emotion detection by learning user voice frequency and validating results with feedback to enhance accuracy and users' emotional satisfaction.
4. **Effects on user's emotional state** - Addressing the uncertainty of whether music recommendations effectively improve users' moods and feelings in daily life.

Overall System Architecture



IT20665852 - Gunasekara C. M.

Information Technology

Component 1

User emotion extraction through a voice clip.

Research Problem

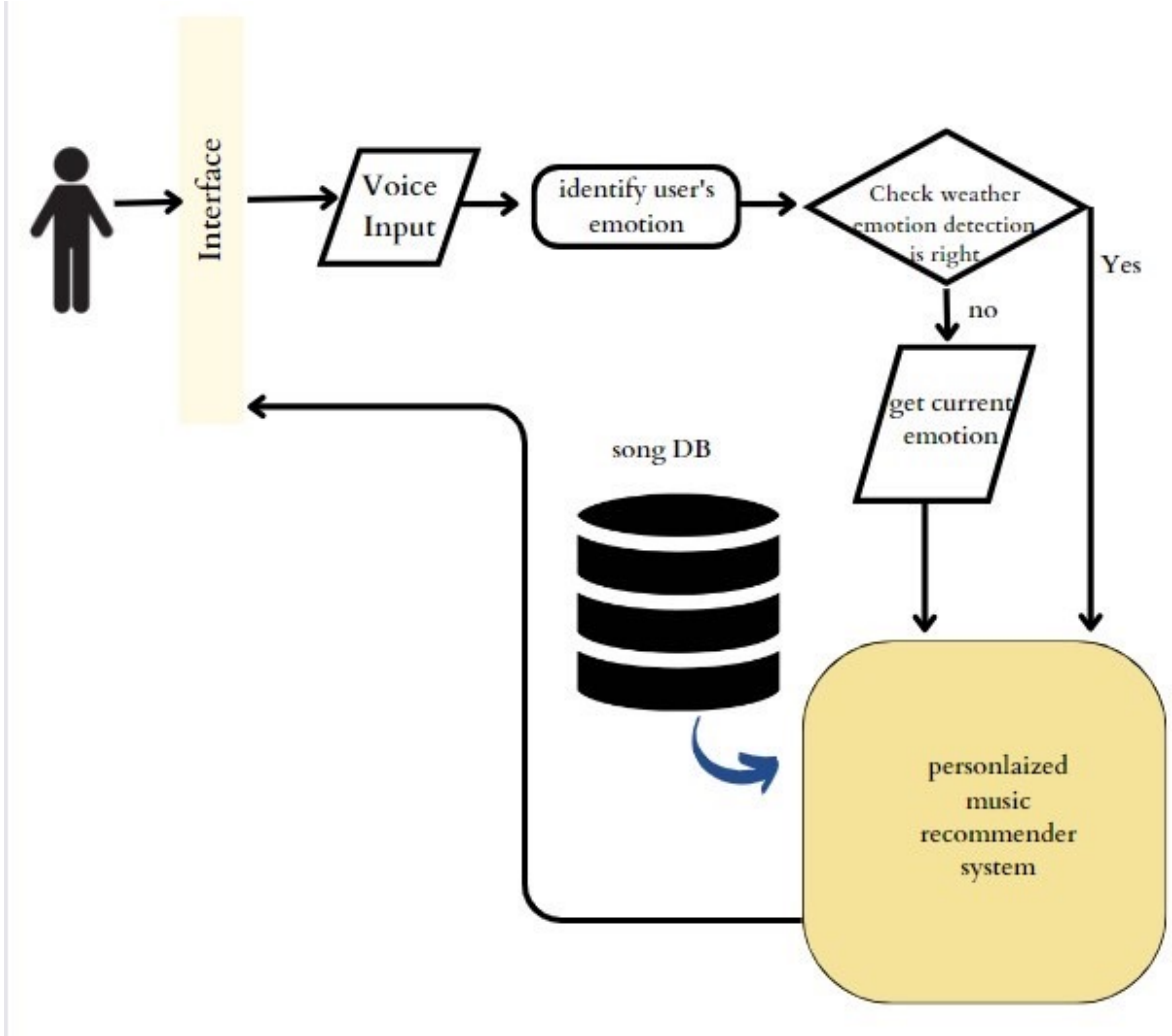
- Currently there is no system for identifying users emotion state using voice input and a system to recommend a playlist based on that emotion state.
- Some machine learning models are used in other industries but emotion based recommendation does not happen in music industry.
- There is no such system to Optimize real-time voice processing.

Research gaps

1. Detect emotions of the user using voice input

2. Generate personalize playlist based on users' emotion state.

Component Overview Diagram



Main Objective

Getting users voice input to extract the emotion state and suggest a playlist according to that emotion state.
Enhance the accuracy of the user's emotion by getting vocal feedbacks from user.

Sub Objectives

- Getting users voice inputs and used that inputs for further classifications
- Extract features.
- Remove noises from the voice clips.
- Identify pitch using voice inputs.
- Identify used emotion state.
- Recommend a playlist based on emotion state.

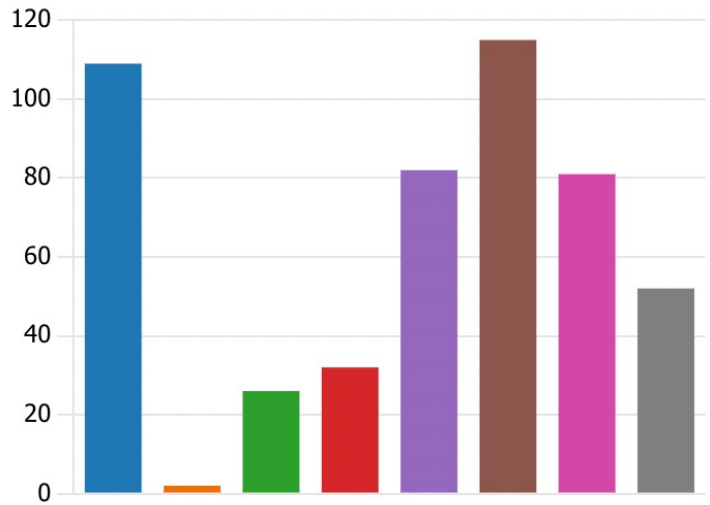
Completion..

1. Collected real world data related to music preferences based on emotion state.

10. How does music make you feel?

More Details

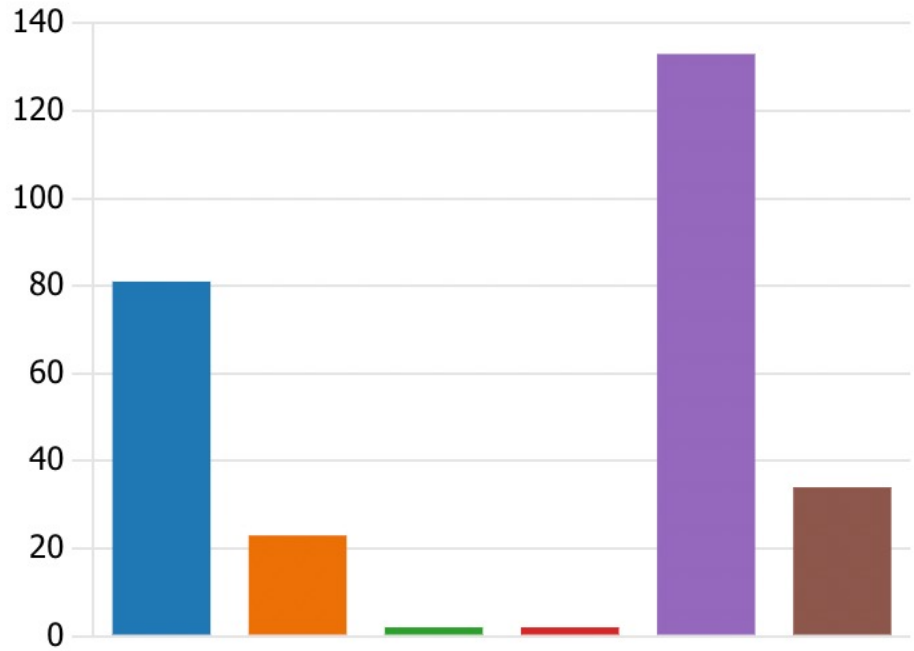
Happy	109
Angry	2
Sad	26
Energised	32
Calm	82
Relaxed	115
Refreshed	81
Motivated	52



11. How do you usually discover/listening new music?

More Details

Spotify	81
iTunes	23
Amazon Music	2
Deezer	2
YouTube	133
Other	34



Completion.. (cont')

2. Find a dataset - **Speech Emotion Recognition**

3. Model creation related to detect emotional state of user.

- Model – CNN - LSTM model(CNN 1D)
- Completion – 50%

Feature extraction

```
[24] def zcr(data,frame_length,hop_length):
         zcr=librosa.feature.zero_crossing_rate(data,frame_length=frame_length,hop_length=hop_length)
         return np.squeeze(zcr)
     def rmse(data,frame_length=2048,hop_length=512):
         rmse=librosa.feature.rms(y=data,frame_length=frame_length,hop_length=hop_length)
         return np.squeeze(rmse)
     def mfcc(data,sr,frame_length=2048,hop_length=512,flatten:bool=True):
         mfcc=librosa.feature.mfcc(y=data,sr=sr)
         return np.squeeze(mfcc.T)if not flatten else np.ravel(mfcc.T)

     def extract_features(data,sr=22050,frame_length=2048,hop_length=512):
         result=np.array([])

         result=np.hstack((result,
                           zcr(data,frame_length,hop_length),
                           rmse(data,frame_length,hop_length),
                           mfcc(data,sr,frame_length,hop_length)
                          ))
         return result

     def get_features(path,duration=2.5, offset=0.6):
         data,sr=librosa.load(path,duration=duration,offset=offset)
         aud=extract_features(data)
         audio=np.array(aud)

         noised_audio=noise(data)
         aud2=extract_features(noised_audio)
         audio=np.vstack((audio,aud2))

         pitched_audio=pitch(data,sr)
         aud3=extract_features(pitched_audio)
         audio=np.vstack((audio,aud3))

         pitched_audio1=pitch(data,sr)
         pitched_noised_audio=noise(pitched_audio1)
         aud4=extract_features(pitched_noised_audio)
         audio=np.vstack((audio,aud4))

         return audio
```

Data preparation

```
[ ] #taking all rows and all cols without last col for X which include features
    #taking last col for Y, which include the emotions


    X = Emotions.iloc[: ,:-1].values
    Y = Emotions['Emotions'].values
```

```
[ ] # As this is a multiclass classification problem onehotencoding our Y
    from sklearn.preprocessing import StandardScaler, OneHotEncoder
    encoder = OneHotEncoder()
    Y = encoder.fit_transform(np.array(Y).reshape(-1,1)).toarray()
```

```
[ ] print(Y.shape)
    X.shape
```

```
[ ] from sklearn.model_selection import train_test_split

    x_train, x_test, y_train, y_test = train_test_split(X, Y, random_state=42,test_size=0.2, shuffle=True)
    x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

```
[ ] #reshape for lstm
    X_train = x_train.reshape(x_train.shape[0] , x_train.shape[1] , 1)
    X_test = x_test.reshape(x_test.shape[0] , x_test.shape[1] , 1)
```

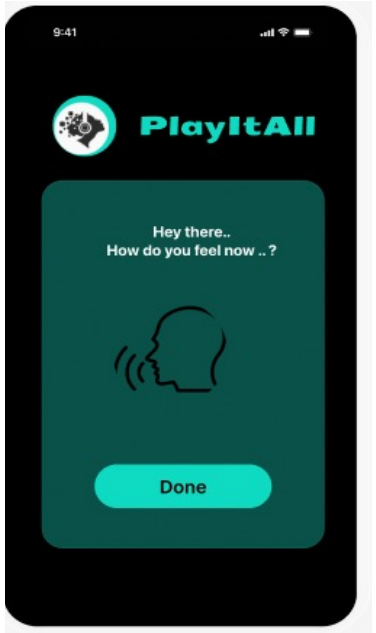
```
[ ] # scaling our data with sklearn's Standard scaler
    scaler = StandardScaler()
    x_train = scaler.fit_transform(x_train)
    x_test = scaler.transform(x_test)
    x_train.shape, y_train.shape, x_test.shape, y_test.shape
```

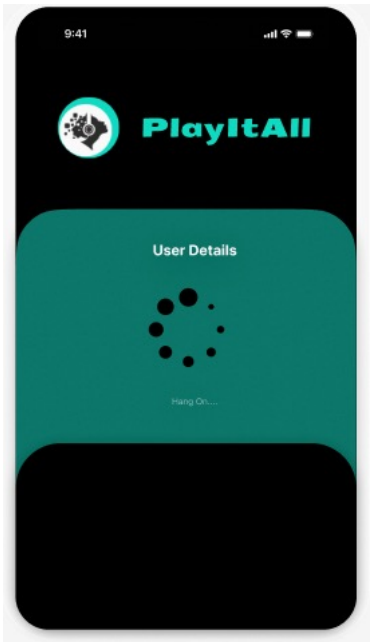
```
[ ]  #Reshape for CNN_LSTM MODEL

     x_traincnn =np.expand_dims(x_train, axis=2)
     x_testcnn= np.expand_dims(x_test, axis=2)
     x_traincnn.shape, y_train.shape, x_testcnn.shape, y_test.shape
```

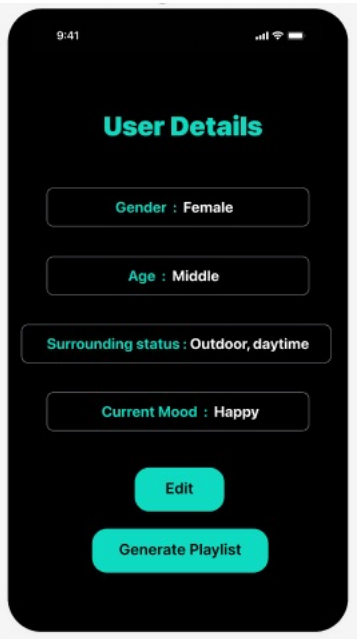
```
import tensorflow.keras.layers as L

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

    L.Conv1D(512,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    Dropout(0.2),  # Add dropout layer after the second max pooling layer

    L.Conv1D(256,kernel_size=5,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),

    L.Conv1D(256,kernel_size=3,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=5,strides=2,padding='same'),
    Dropout(0.2),  # Add dropout layer after the fourth max pooling layer

    L.Conv1D(128,kernel_size=3,strides=1,padding='same',activation='relu'),
    L.BatchNormalization(),
    L.MaxPool1D(pool_size=3,strides=2,padding='same'),
    Dropout(0.2),  # Add dropout layer after the fifth max pooling layer

    L.Flatten(),
    L.Dense(512,activation='relu'),
    L.BatchNormalization(),
    L.Dense(8,activation='softmax')
])
#compile the model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics='accuracy')
model.summary()
```

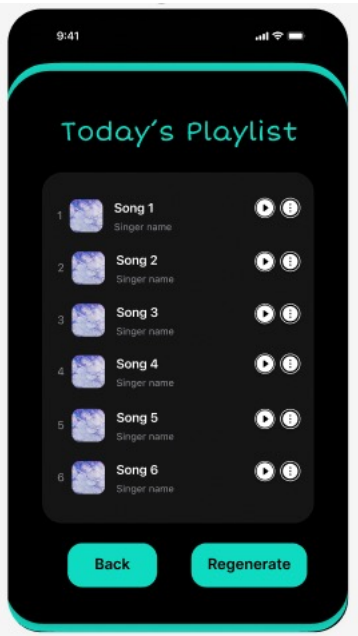
Completion.. (cont')

3. Create UI prototypes for the mobile application.
   - Technology - Figma, Flutter









Next Steps

Accuracy Improvements

Completion of the Music Recommendation model

Overral System Integration

IT20610852 - Fernando M. P. T. K.

Information Technology

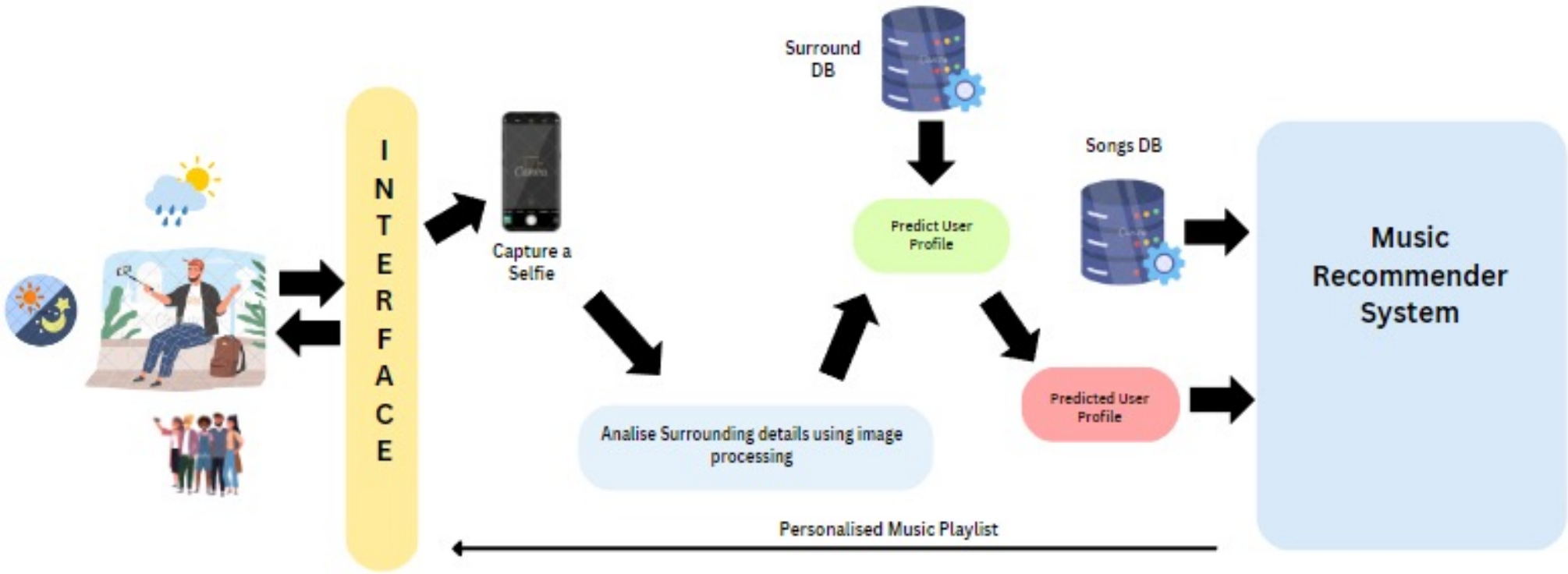
Component 2

Music Recommendations based on the surrounding of the user.

Introduction

- Music recommendations enhance the user's personalize experience by considering surroundings.
- Music recommendation system analyses user surroundings for personalised suggestions.
- Surrounding factors inform music choice, including day/night, weather, and more.
- Approach improves personalised music experience, enhancing user engagement and Journey.

Component Overview Diagram



Research background

- Music recommendation based on Surrounding.
- Surrounding factors inform music choice, including day/night, weather
- Involves categorising the image into different scenes or contexts.
- Training images of the user's surroundings for personalised playlist suggestions.

Specific and sub objectives

- **Specific Objective**

Using image processing and deep learning to recognise the surroundings of the user's selfie photo and suggest a personalised playlist

- **Sub Objective**

1. Obtaining Data Sets
2. Image Analysis and Context Detection
3. Validating the performance of the developed models
4. Real-time data integration

Current Progress

- Implement machine learning model for classifying the context of the user.

- Data collection: Manuel data generation

Current Progress

Data set- human manual choices by using google form

2. During which time of the day do you prefer to listen to music the most ?

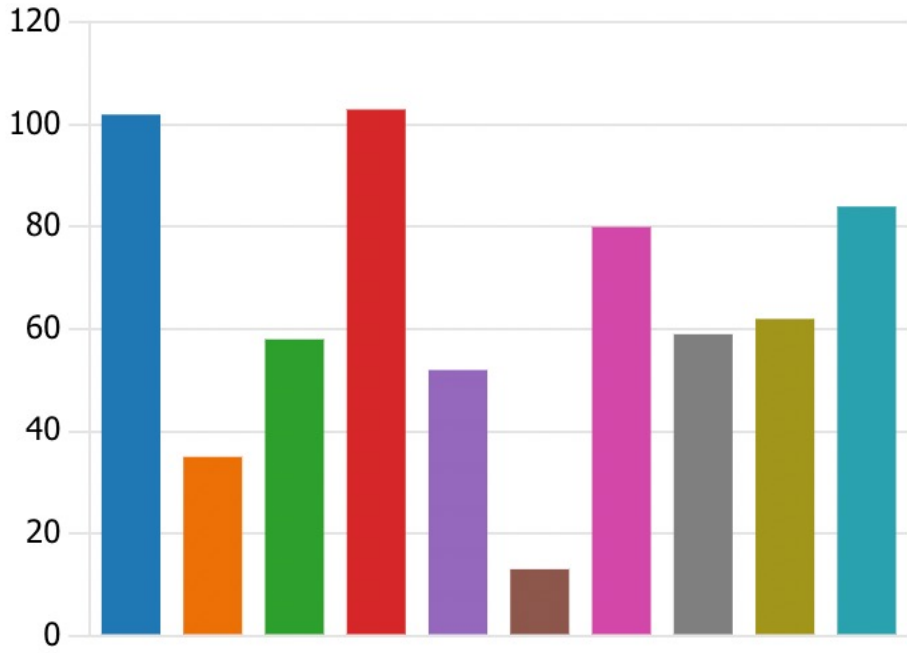
More Details

Morning	40
Afternoon	18
Evening	64
Night	115
Day	29

8. Which situation do you prefer to listen to songs?

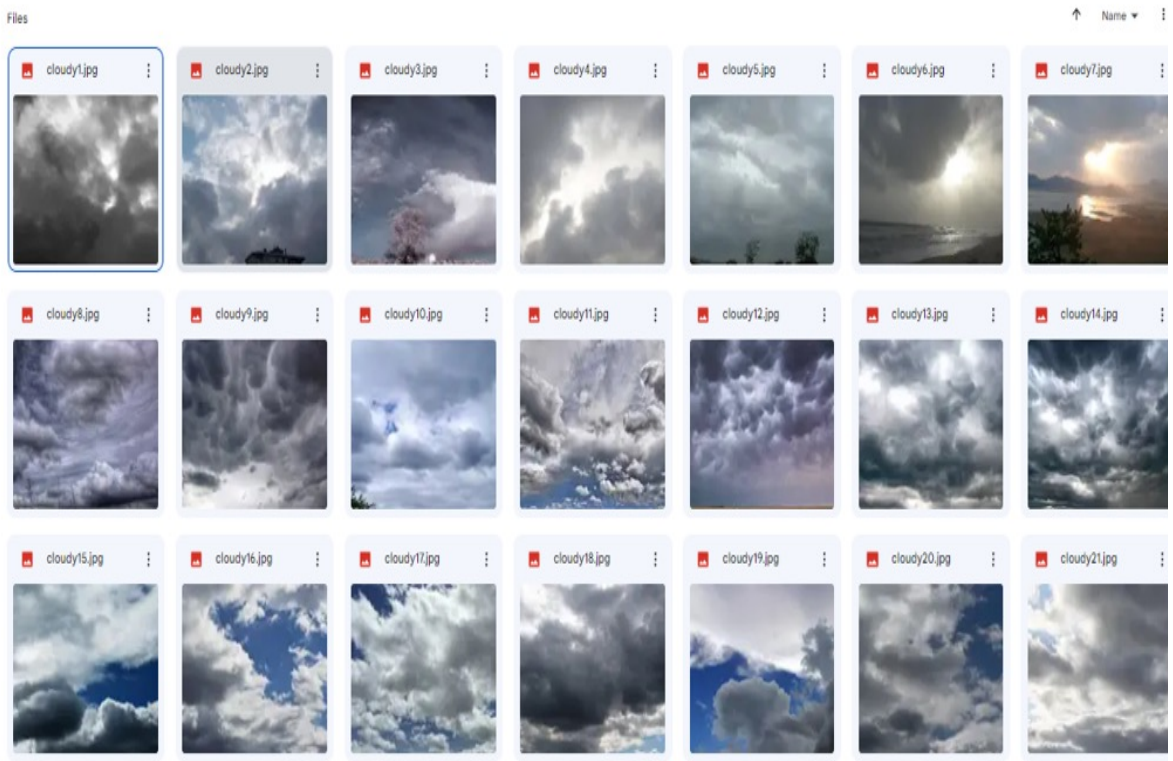
More Details

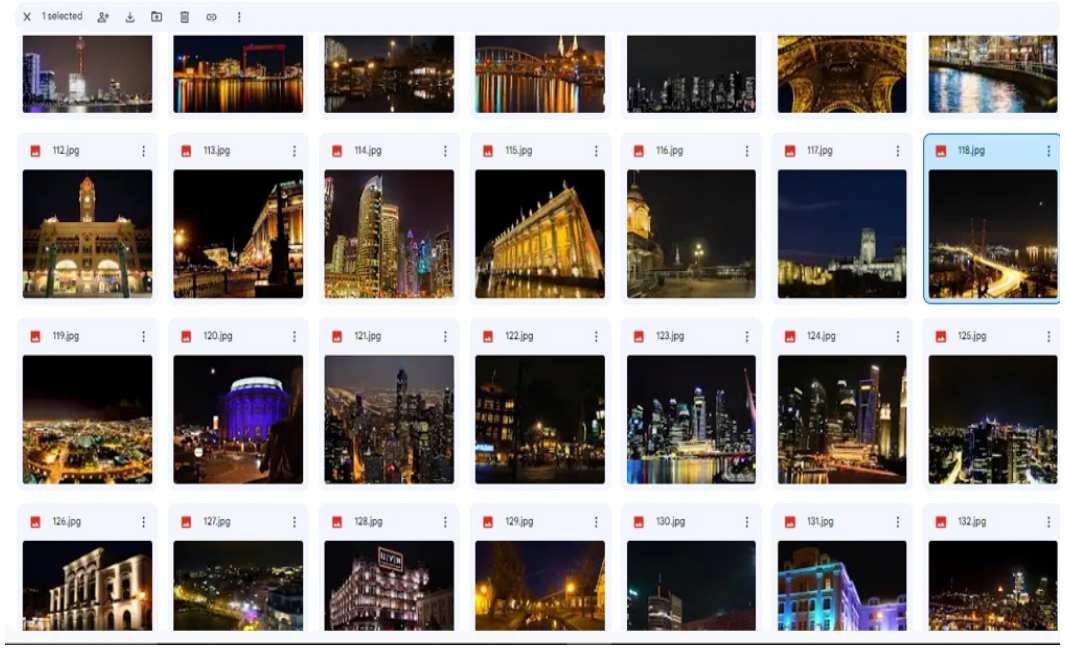
While Travelling	102
While Studying	35
While doing workouts	58
During free time	103
Before going to sleep	52
While having meals	13
When you feel stressed	80
When you feel happy	59
When you feel sad	62
When you feel alone	84



Current Progress

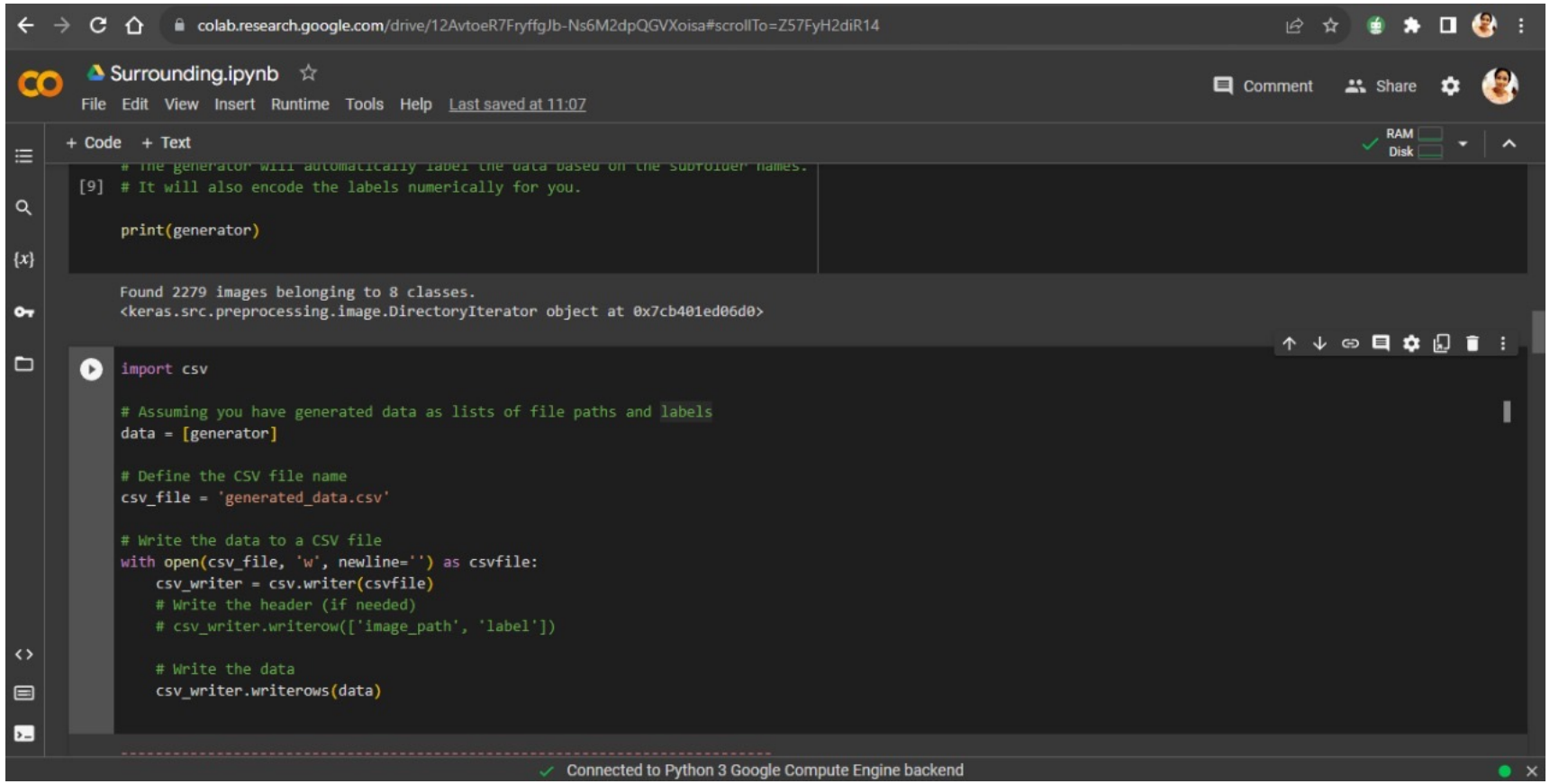
Dataset - Google drive

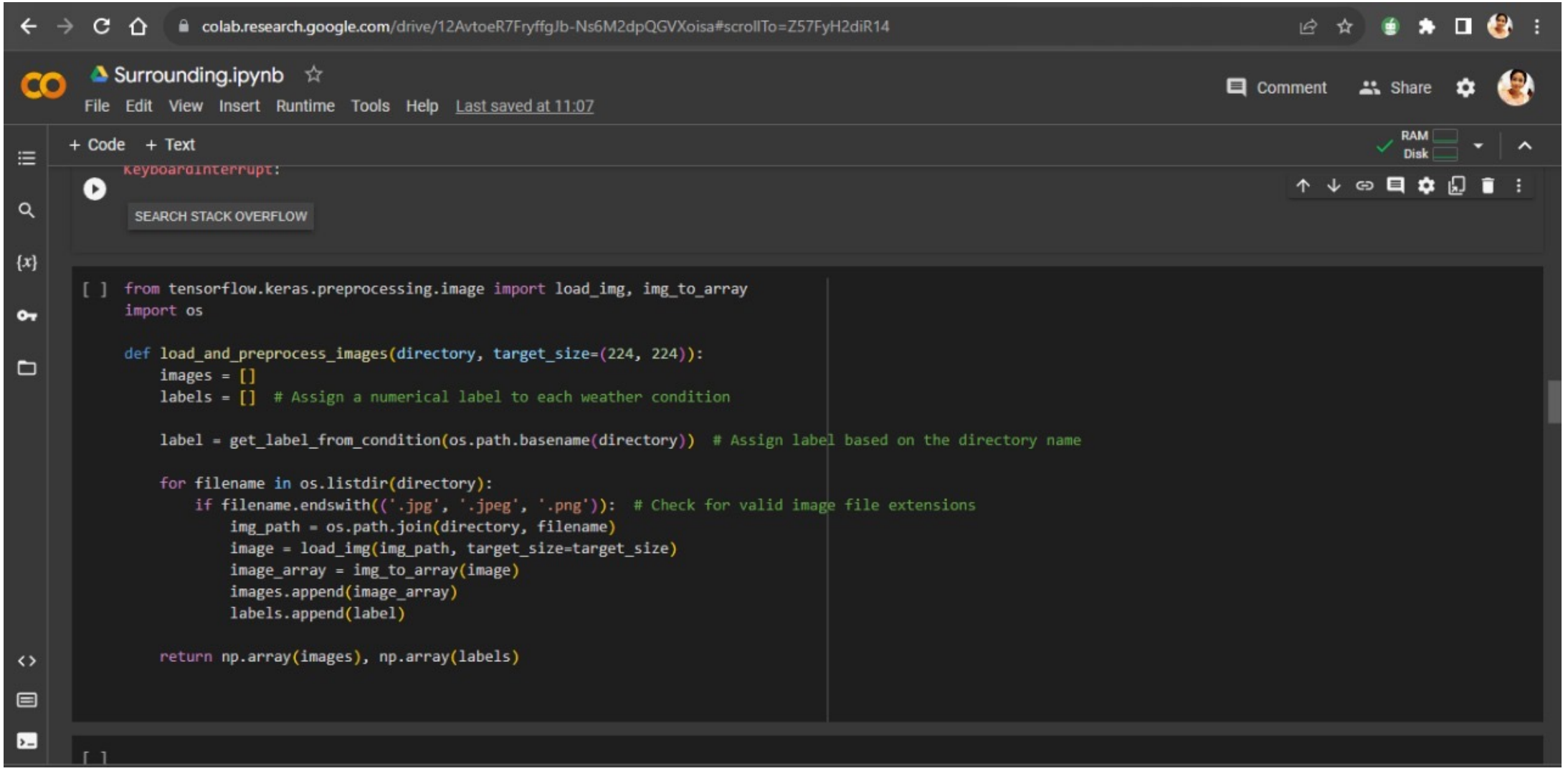


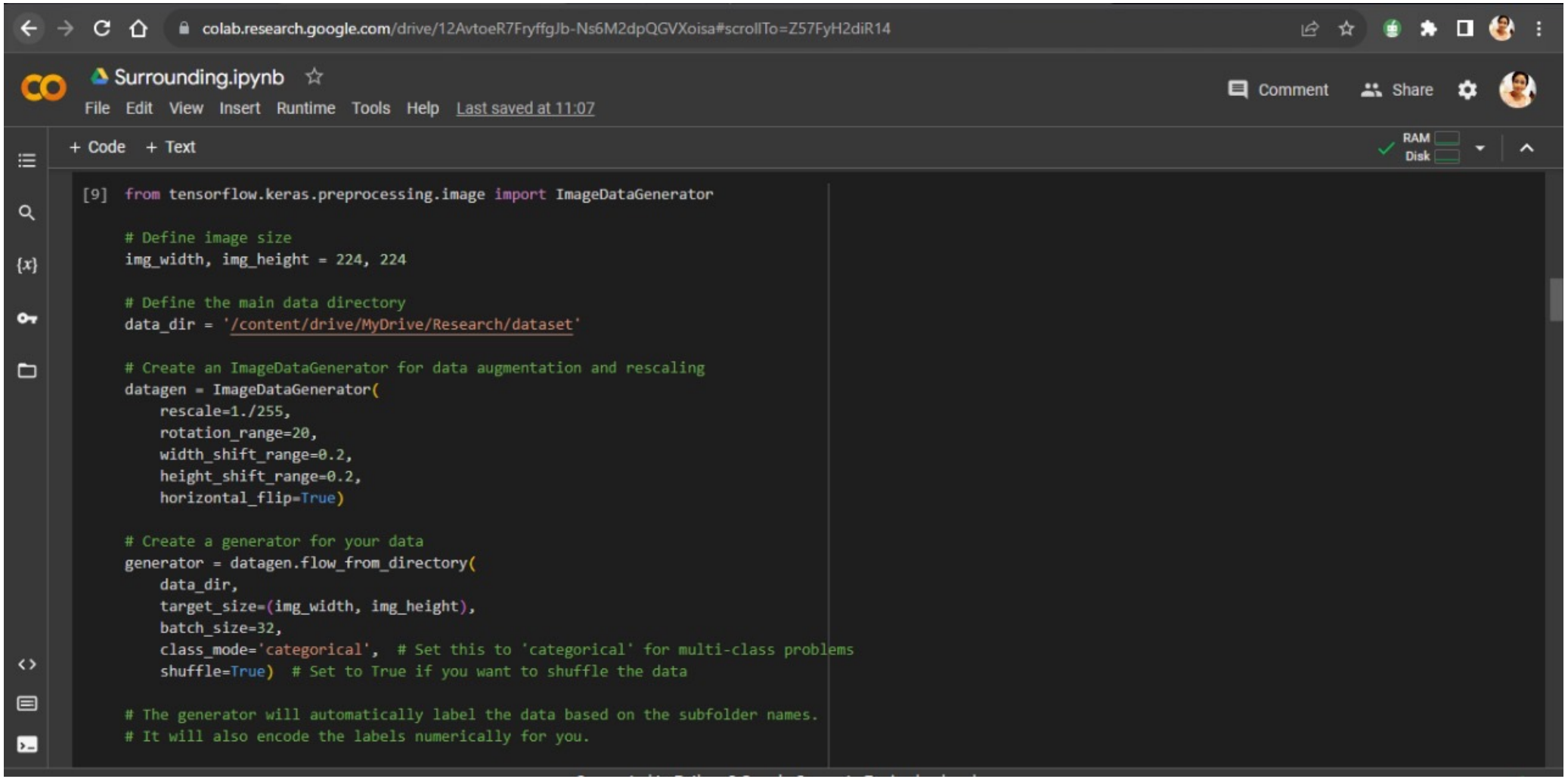


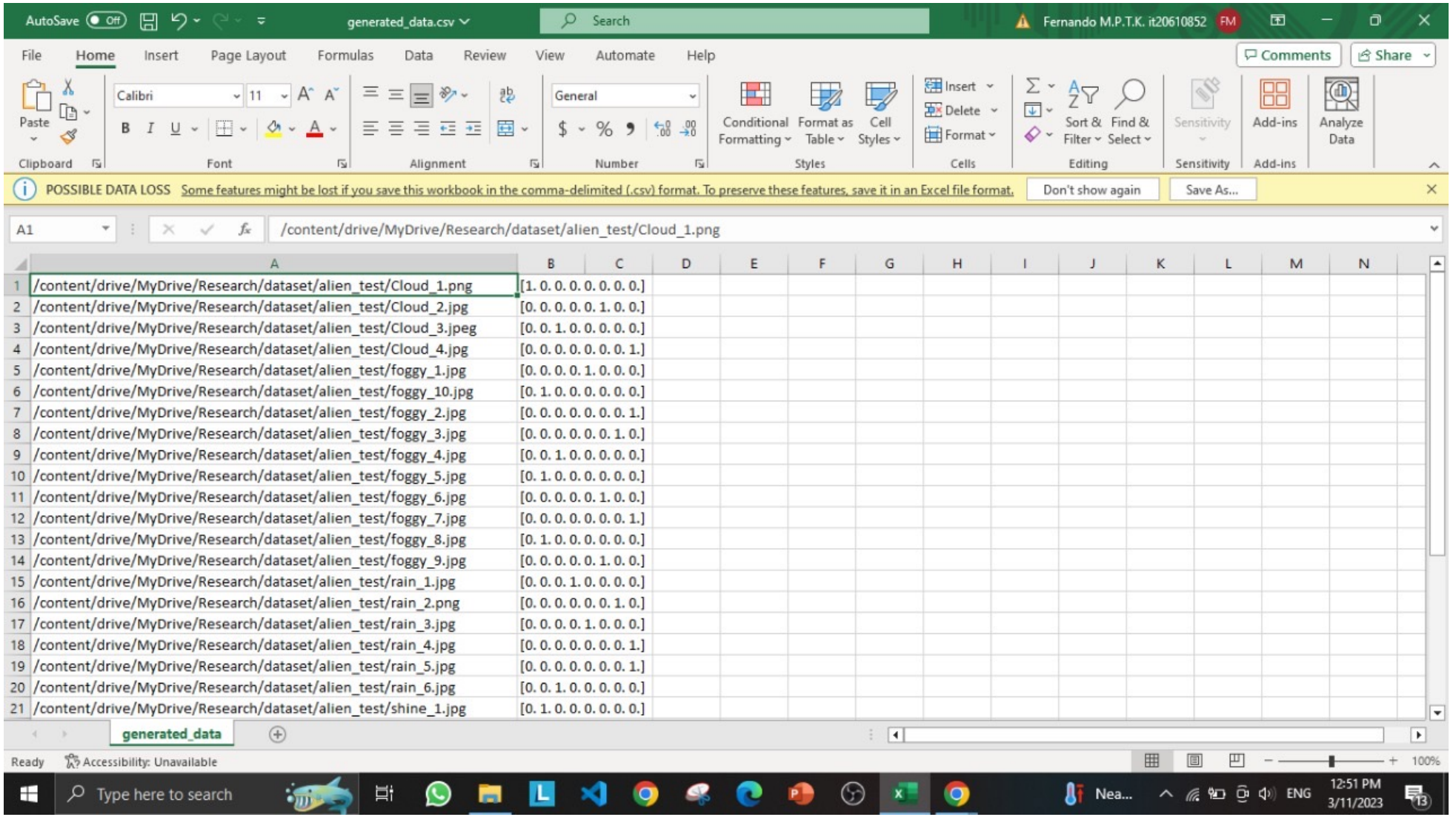
Current Progress

Sample output




colab.research.google.com/drive/12AvtoeR7FryffgJb-Ns6M2dpQGVXoisa#scrollTo=Z57FyH2diR14
Surrounding.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:07
Comment
Share
+ Code + Text
RAM
Disk
KeyboardInterrupt:
SEARCH STACK OVERFLOW
[ ] from tensorflow.keras.preprocessing.image import load_img, img_to_array
import os
def load_and_preprocess_images(directory, target_size=(224, 224)):
images = []
labels = [] # Assign a numerical label to each weather condition
label = get_label_from_condition(os.path.basename(directory)) # Assign label based on the directory name
for filename in os.listdir(directory):
if filename.endswith(('.jpg', '.jpeg', '.png')): # Check for valid image file extensions
img_path = os.path.join(directory, filename)
image = load_img(img_path, target_size=target_size)
image_array = img_to_array(image)
images.append(image_array)
labels.append(label)
return np.array(images), np.array(labels)
[ ]


colab.research.google.com/drive/12AvtoeR7FryffgJb-Ns6M2dpQGVXoisa#scrollTo=Z57FyH2diR14
Surrounding.ipynb
File Edit View Insert Runtime Tools Help Last saved at 11:07
Comment
Share
+ Code + Text
RAM
Disk
[9] from tensorflow.keras.preprocessing.image import ImageDataGenerator
Define image size
img_width, img_height = 224, 224
# Define the main data directory
data_dir = '/content/drive/MyDrive/Research/dataset'
# Create an ImageDataGenerator for data augmentation and rescaling
datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
horizontal_flip=True)
# Create a generator for your data
generator = datagen.flow_from_directory(
data_dir,
target_size=(img_width, img_height),
batch_size=32,
class_mode='categorical', # Set this to 'categorical' for multi-class problems
shuffle=True) # Set to True if you want to shuffle the data
# The generator will automatically label the data based on the subfolder names.
# It will also encode the labels numerically for you.



	A	B
1	/content/drive/MyDrive/Research/dataset/alien_test/Cloud_1.png	[1. 0. 0. 0. 0. 0. 0.]
2	/content/drive/MyDrive/Research/dataset/alien_test/Cloud_2.jpg	[0. 0. 0. 0. 0. 1. 0.]
3	/content/drive/MyDrive/Research/dataset/alien_test/Cloud_3.jpeg	[0. 0. 1. 0. 0. 0. 0.]
4	/content/drive/MyDrive/Research/dataset/alien_test/Cloud_4.jpg	[0. 0. 0. 0. 0. 0. 1.]
5	/content/drive/MyDrive/Research/dataset/alien_test/foggy_1.jpg	[0. 0. 0. 0. 1. 0. 0.]
6	/content/drive/MyDrive/Research/dataset/alien_test/foggy_10.jpg	[0. 1. 0. 0. 0. 0. 0.]
7	/content/drive/MyDrive/Research/dataset/alien_test/foggy_2.jpg	[0. 0. 0. 0. 0. 0. 1.]
8	/content/drive/MyDrive/Research/dataset/alien_test/foggy_3.jpg	[0. 0. 0. 0. 0. 1. 0.]
9	/content/drive/MyDrive/Research/dataset/alien_test/foggy_4.jpg	[0. 0. 1. 0. 0. 0. 0.]
10	/content/drive/MyDrive/Research/dataset/alien_test/foggy_5.jpg	[0. 1. 0. 0. 0. 0. 0.]
11	/content/drive/MyDrive/Research/dataset/alien_test/foggy_6.jpg	[0. 0. 0. 0. 0. 1. 0.]
12	/content/drive/MyDrive/Research/dataset/alien_test/foggy_7.jpg	[0. 0. 0. 0. 0. 0. 1.]
13	/content/drive/MyDrive/Research/dataset/alien_test/foggy_8.jpg	[0. 1. 0. 0. 0. 0. 0.]
14	/content/drive/MyDrive/Research/dataset/alien_test/foggy_9.jpg	[0. 0. 0. 0. 0. 1. 0.]
15	/content/drive/MyDrive/Research/dataset/alien_test/rain_1.jpg	[0. 0. 0. 1. 0. 0. 0.]
16	/content/drive/MyDrive/Research/dataset/alien_test/rain_2.png	[0. 0. 0. 0. 0. 1. 0.]
17	/content/drive/MyDrive/Research/dataset/alien_test/rain_3.jpg	[0. 0. 0. 0. 1. 0. 0.]
18	/content/drive/MyDrive/Research/dataset/alien_test/rain_4.jpg	[0. 0. 0. 0. 0. 0. 1.]
19	/content/drive/MyDrive/Research/dataset/alien_test/rain_5.jpg	[0. 0. 0. 0. 0. 0. 1.]
20	/content/drive/MyDrive/Research/dataset/alien_test/rain_6.jpg	[0. 0. 1. 0. 0. 0. 0.]
21	/content/drive/MyDrive/Research/dataset/alien_test/shine_1.jpg	[0. 1. 0. 0. 0. 0. 0.]

Next Steps

Data collection: collect more local data sets.

Completion of the classification model model

Accuracy improvments

Overall system integration.

IT20667078 - Dhananjaya W. K. S.

Information Technology

Component 3

Speech Emotion Recognition to detect post emotional state of a user.

Research gaps

1. There is currently no established recommendation system for identify post user emotional state.

2. Current recommendation systems only consider the pre emotion of user.

3. There are less focus on personalized music playlists based on emotion.

Specific and sub objectives

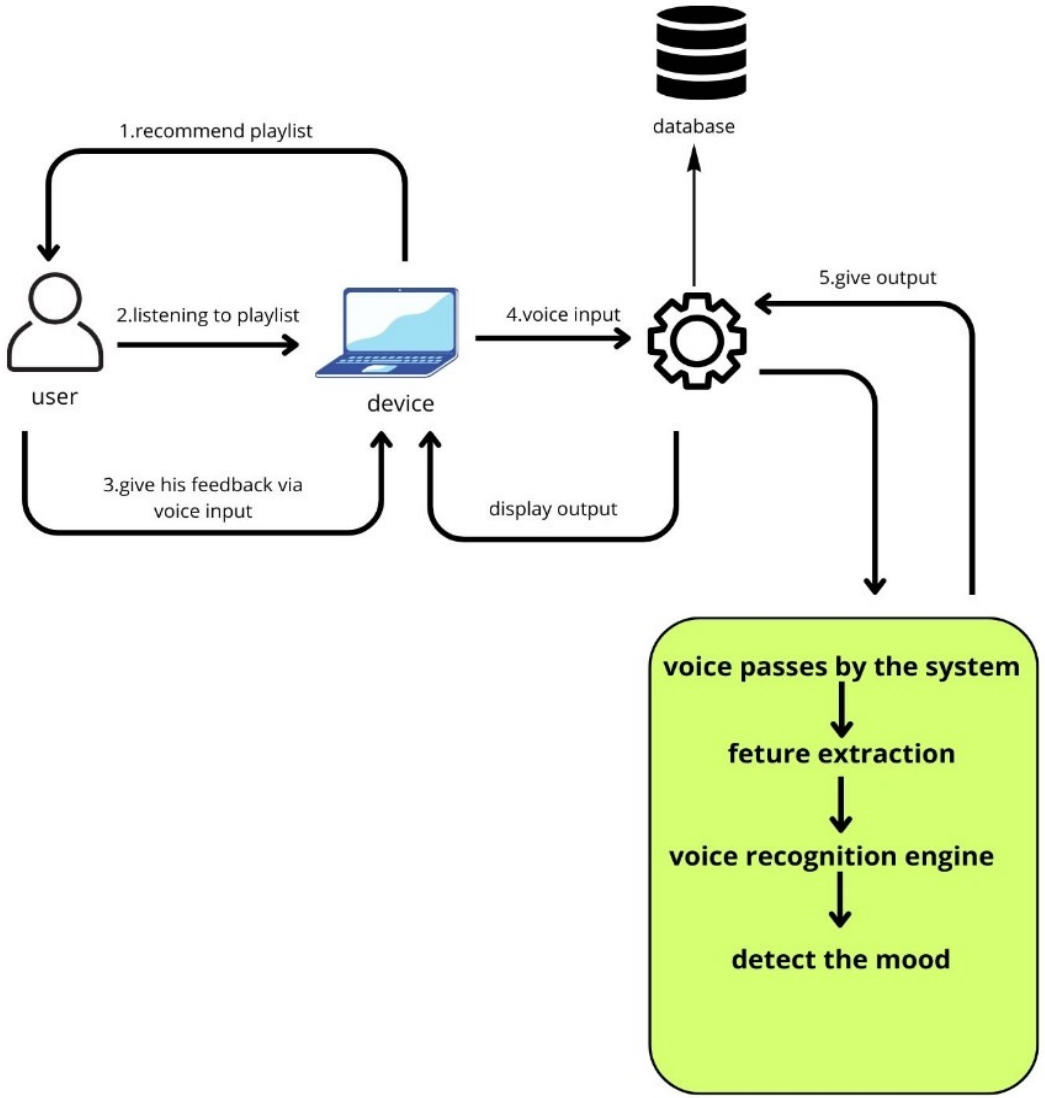
- **Specific Objective**

Improve user experience by understanding their emotional state and Improve more personalised experience

- **Sub Objective**

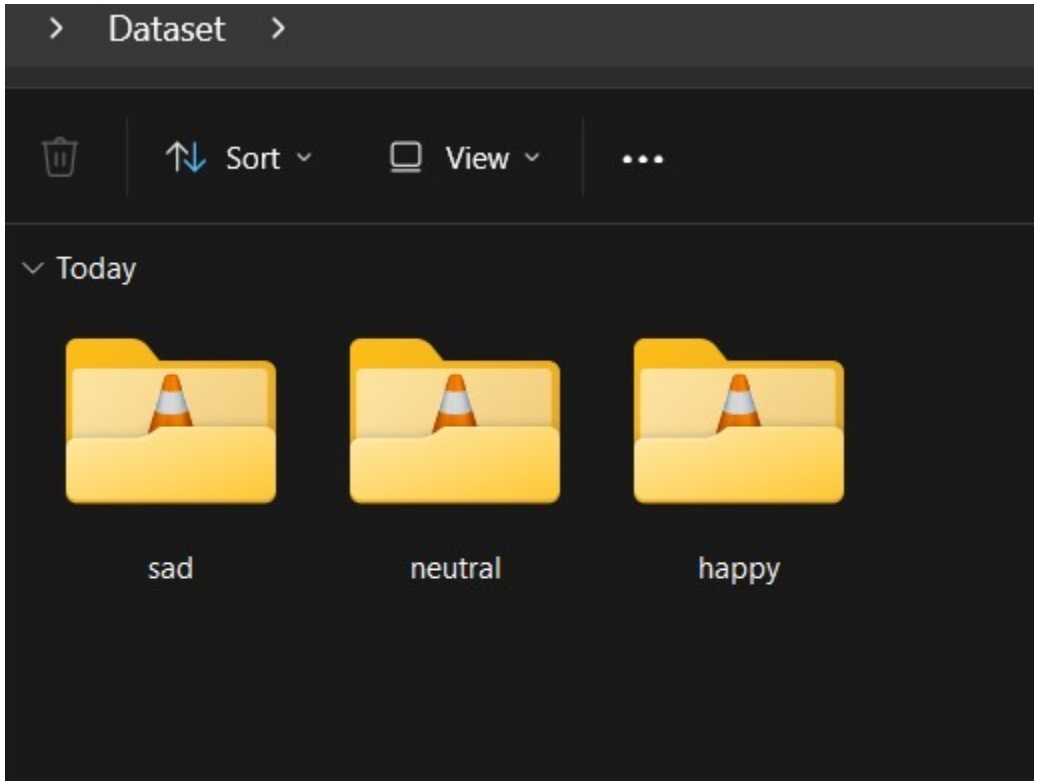
Identify post emotional state of user.
Obtaining data sets, Identification and classification of post emotion through audio processing.

Component Overview Diagram



Completion..

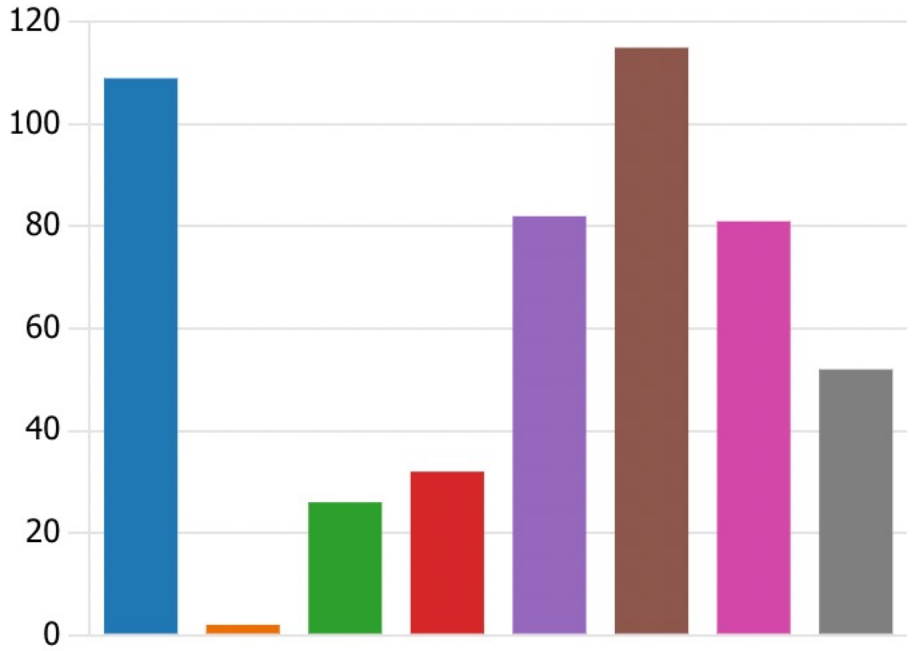
1. Collected real world data related to post emotions and make dataset.



10. How does music make you feel?

More Details

Happy	109
Angry	2
Sad	26
Energised	32
Calm	82
Relaxed	115
Refreshed	81
Motivated	52



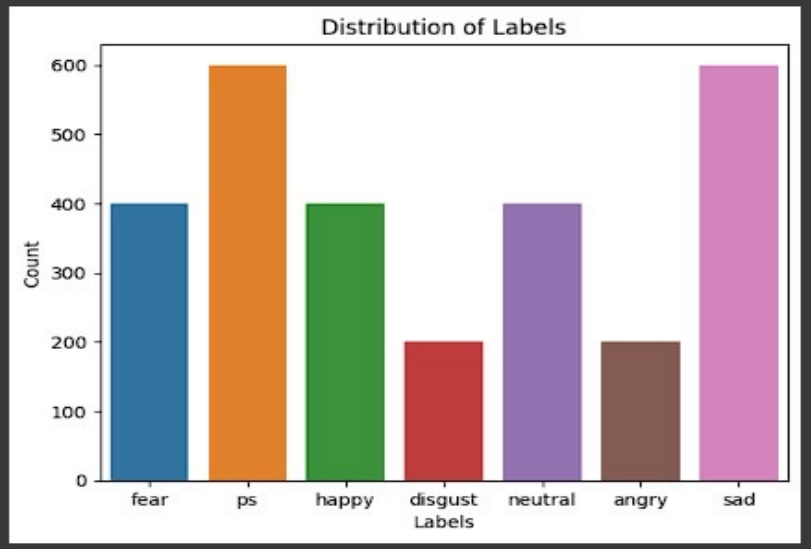
Current Progress.. (cont')

2. Find a dataset - Toronto emotional speech set (TESS)
   Kaggle link - *https://www.kaggle.com/datasets/ejlok1/toronto-emotional-speech-set-tess*

3. Model creation related to detect post emotional state of user.

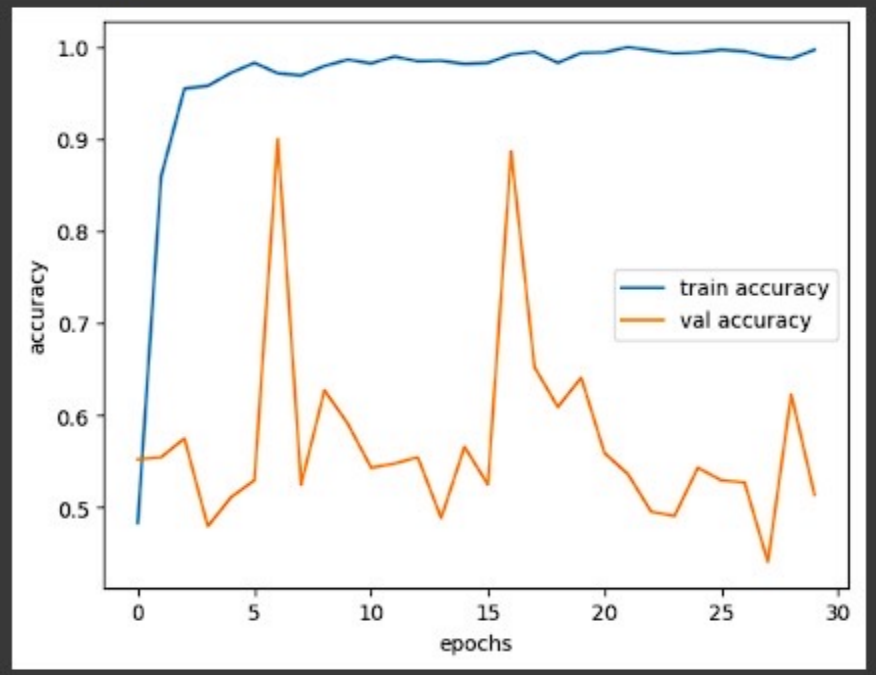
- Model – Implement speech emotional recognition model.
- Completion – 50%

```python
import seaborn as sns
import matplotlib.pyplot as plt

# Create a count plot
sns.countplot(data=df, x='label')
plt.xlabel('Labels')
plt.ylabel('Count')
plt.title('Distribution of Labels')
plt.show()
```



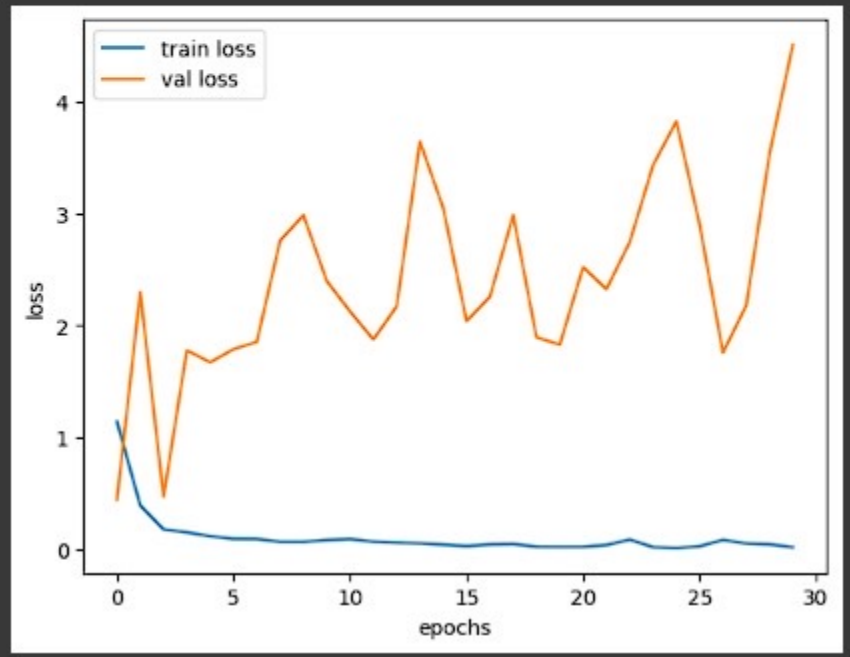

```python
[9] import librosa.display
    import matplotlib.pyplot as plt

    def waveplot(data, sr, emotion):
        plt.figure(figsize=(10, 4))
        plt.title(emotion, size=20)
        librosa.display.waveshow(data, sr=sr)
        plt.show()

    def spectrogram(data, sr, emotion):
        x = librosa.stft(data)
        xdb = librosa.amplitude_to_db(abs(x))
        plt.figure(figsize=(11, 4))
        plt.title(emotion, size=20)
        librosa.display.specshow(xdb, sr=sr, x_axis='time', y_axis='hz')
        plt.colorbar()
```

```
epochs = list(range(30))
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

plt.plot(epochs, acc, label='train accuracy')
plt.plot(epochs, val_acc, label='val accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
```

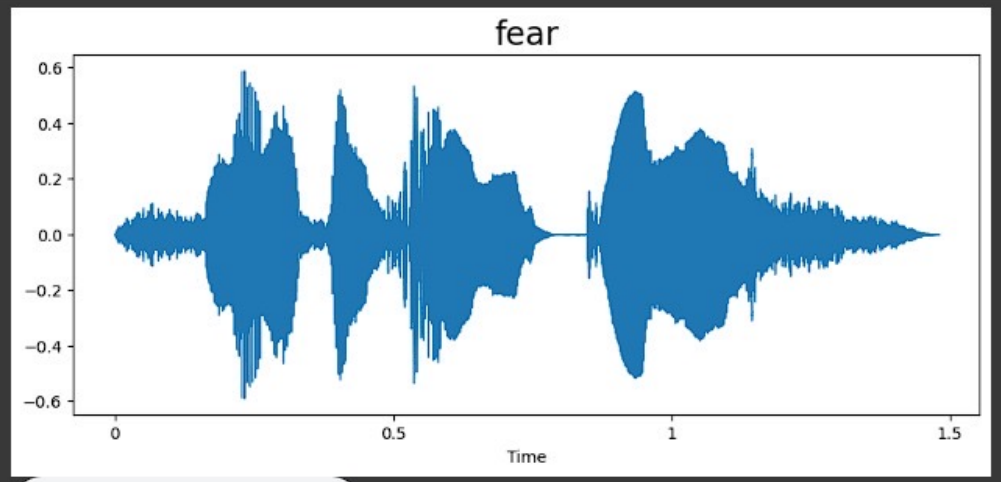


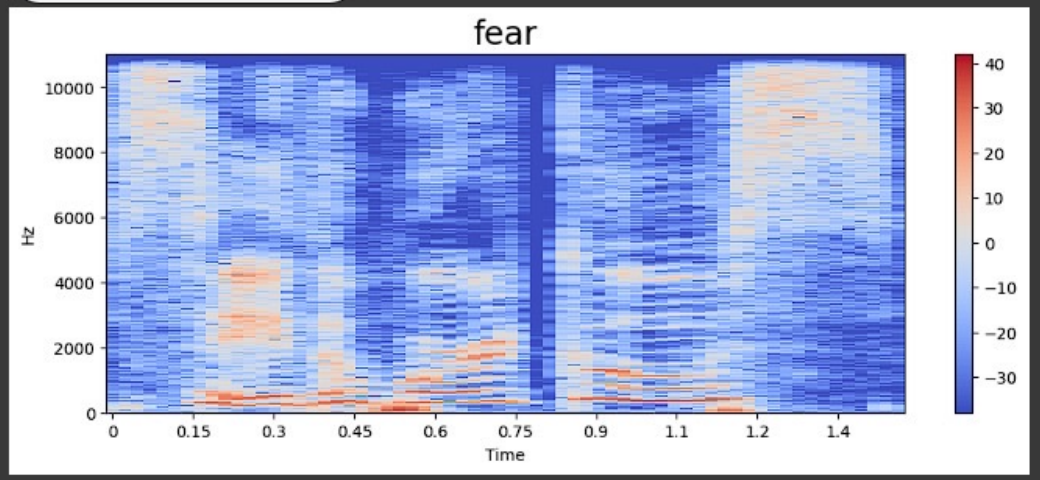

```
[40] loss = history.history['loss']
     val_loss = history.history['val_loss']

     plt.plot(epochs, loss, label='train loss')
     plt.plot(epochs, val_loss, label='val loss')
```

```
loss = history.history['loss']
val_loss = history.history['val_loss']

plt.plot(epochs, loss, label='train loss')
plt.plot(epochs, val_loss, label='val loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



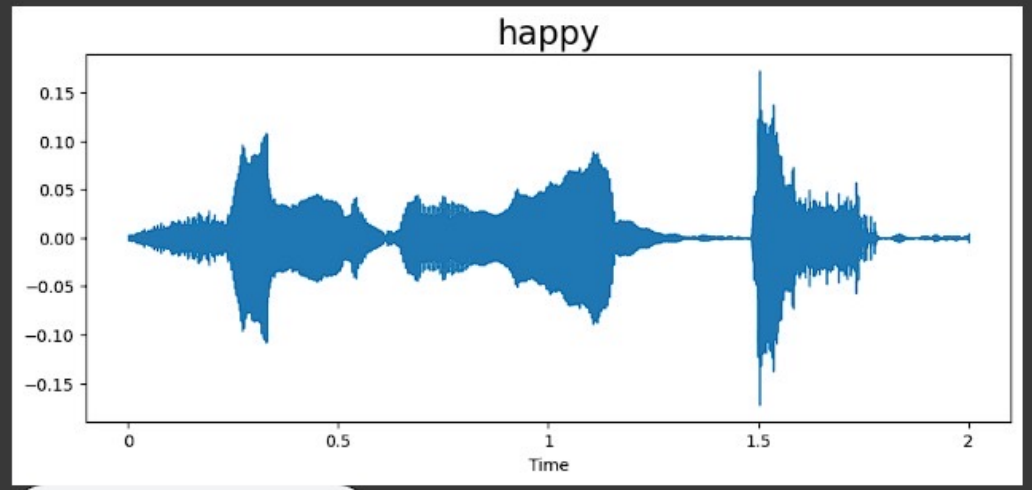
```
[ ]  emotion = 'fear'
     path = np.array(df['speech'][df['label']==emotion])[0]
     data, sampling_rate = librosa.load(path)
     waveplot(data, sampling_rate, emotion)
     spectrogram(data, sampling_rate, emotion)  # Corrected function name
     Audio(path)
```

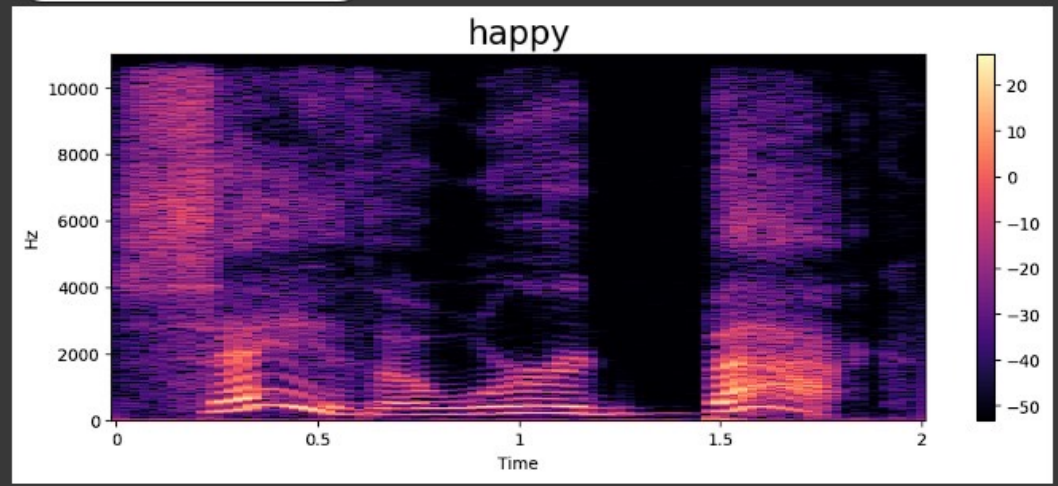





```
[ ]  emotion = 'disgust'
```

```
[31] emotion = 'happy'
     path = np.array(df['speech'][df['label']==emotion])[0]
     data, sampling_rate = librosa.load(path)
     waveplot(data, sampling_rate, emotion)
     spectrogram(data, sampling_rate, emotion)
     Audio(path)
```





Next Steps

Accuracy Improvements

Completion of the Music Recommendation model

Overral System Integration

IT20665616 - Sumanasekara H. P.

Information Technology

Component 4

Image classification model to predict a user profile.

Research gaps

1. Music Recommendation based on age and gender.
2. Collecting user details using a selfie rather than filling lengthy forms.
3. Predict a user profile
4. Reduce cold start problem
5. Generate personalized Playlist

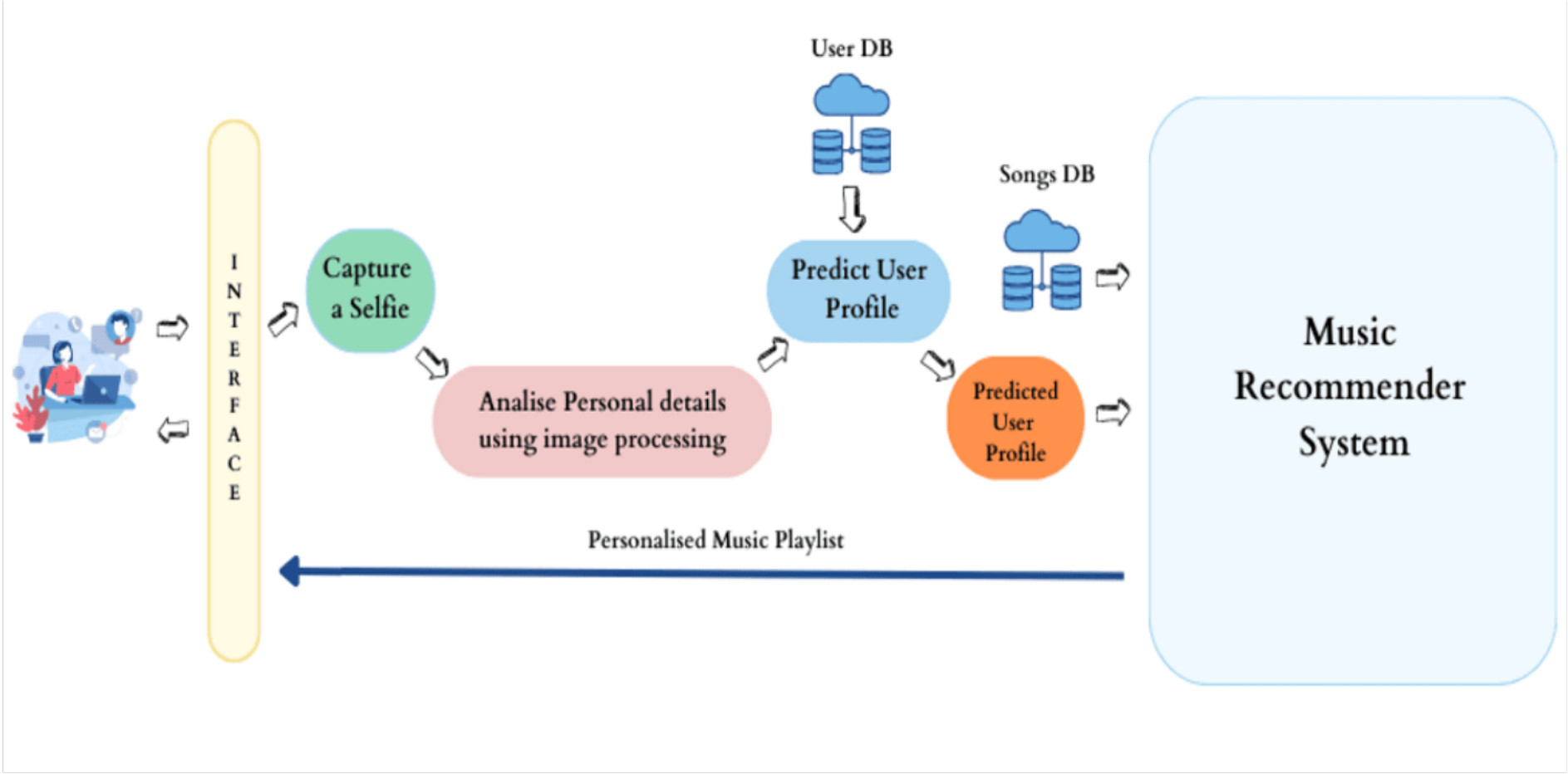
Main Objective

They are capturing a high-quality selfie, analysing the facial details, analysing the features to predict the gender and the age of the user to feed to the music recommendation model.

Sub Objectives

- Collect a proper image to process facial data into image classification model to identify the gender and age of the user.
- Train the image classification model to increase accuracy.
- Predict the user profile based on extracted details.
- Send the user profile as an input to recommender system.
- Integrate above features into the developed mobile application.

Component Overview Diagram



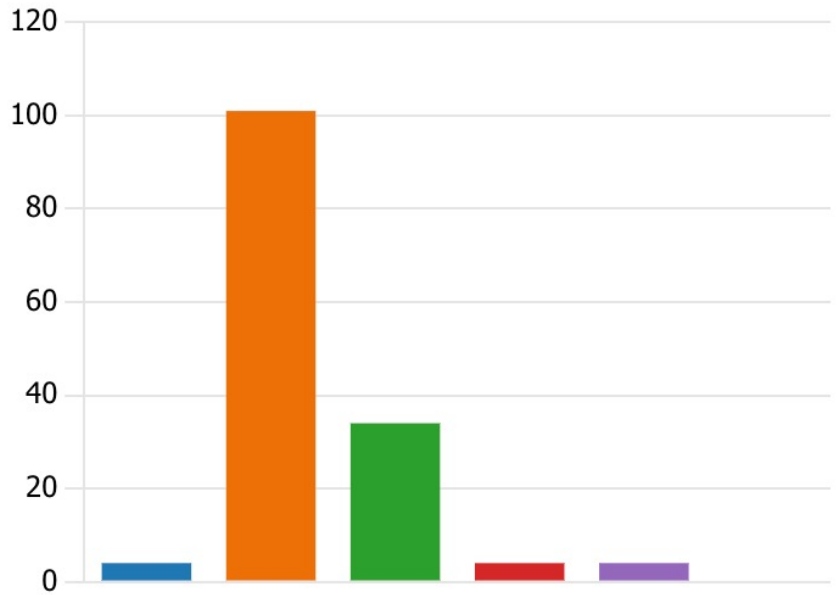
Current Progress

1. Data Collection

5. What is your age group ?

More Details

13-17	4
18-24	101
25-34	34
35-44	4
45-54	4
Above 55	0



Multi-Model Approach for music recommendation survey.

147	03:01	Active
Responses	Average time to complete	Status

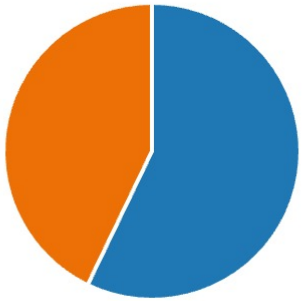
View results

Open in Excel ...

1. What is your gender ?

More Details

Male	84
Female	63



Current Progress.. (cont')

2. Datasets

- **Faces: Age Detection Dataset** - *https://shorturl.at/dqxL5*
- **UTKFace** - *https://www.kaggle.com/datasets/jangedoo/utkface-new*

3. Data preprocessing and annotation

4. Implement the facial image classification model that outputs the age of the user.
- Model – CNN - Age identification through facial image.
- Completion – 50%

```
[3]  csv_file = '/content/drive/MyDrive/Age_Dataset/train.csv'
     data = pd.read_csv(csv_file)
```

```
[4]  image_dir='/content/drive/MyDrive/Age_Dataset/Train'
```

Image preprocessor

```
[5]  X=[]
     Y=[]
```

```
image_count=0
for index, row in data.iterrows():
  if image_count >=3500:
    break;
  # if image_count == 0:
    # continue;
  image_path = os.path.join(image_dir, row['ID'])
  image = load_img(image_path, target_size=(224, 224))
  # Load and resize the image
  image = img_to_array(image) / 255.0  # Normalize pixel values
  X.append(image)
  Y.append(row['Class'])
  image_count+=1
```

Creating CNN model

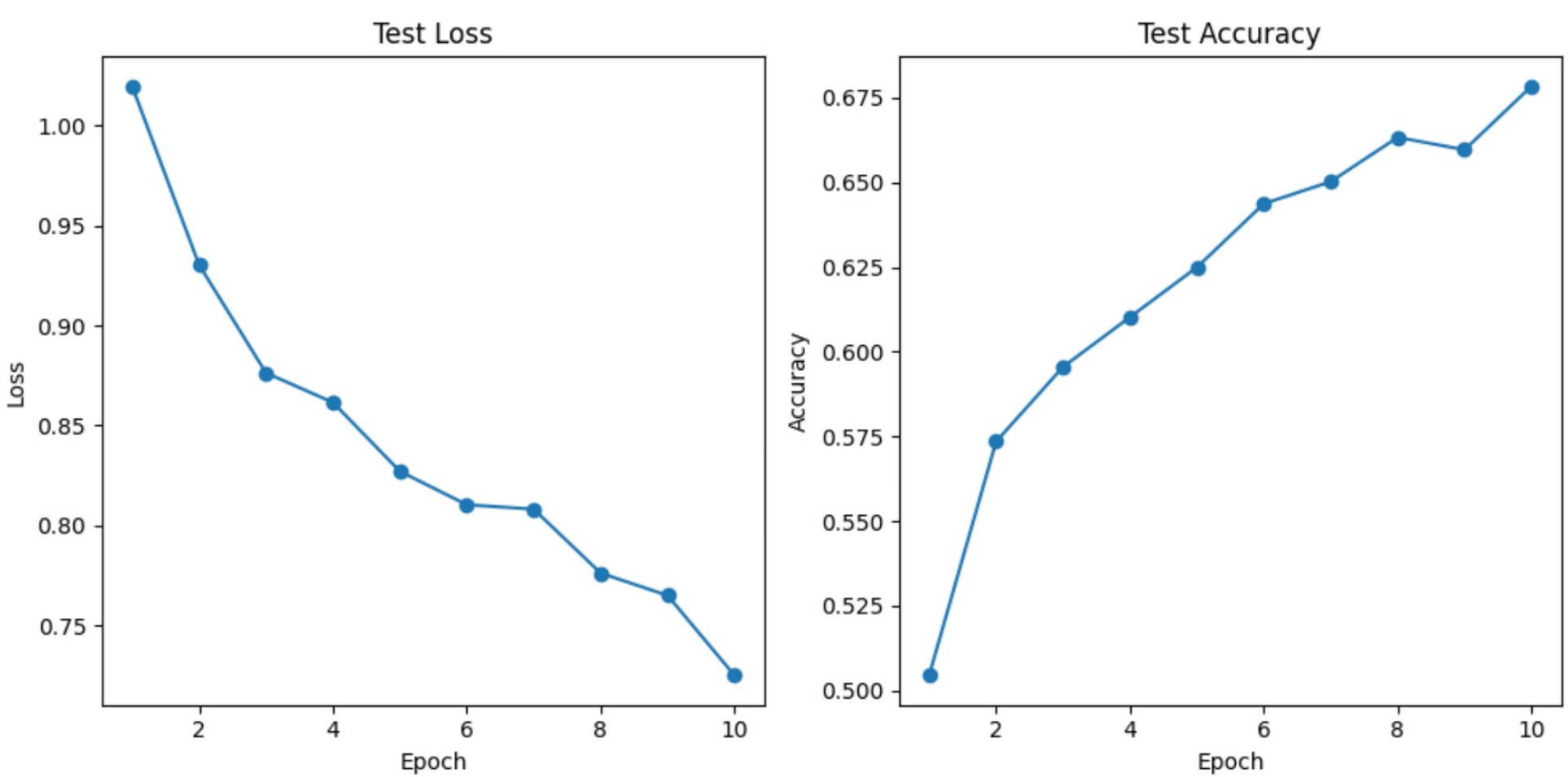
```
[ ]  from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout

     model = Sequential()

     # Convolutional layers
     model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(224, 224, 3)))
     model.add(MaxPooling2D((2, 2)))
     model.add(Conv2D(64, (3, 3), activation='relu'))
     model.add(MaxPooling2D((2, 2)))
     model.add(Conv2D(128, (3, 3), activation='relu'))
     model.add(MaxPooling2D((2, 2)))

     # Flatten the output for fully connected layers
     model.add(Flatten())

     # Fully connected layers
     model.add(Dense(128, activation='relu'))
     model.add(Dropout(0.7))  # Dropout layer to prevent overfitting
     model.add(Dense(3, activation='softmax'))  # 3 output neurons for age categories

     # Compile the model
     model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['a
```

Training model with early stopping

```
[ ]  from tensorflow.keras.callbacks import EarlyStopping

     # Define EarlyStopping
     early_stopping = EarlyStopping(monitor='val_loss', patience=2, restore_best_weights=

     # Train your model with early stopping
     history = model.fit(X_train, Y_train, validation_data=(X_val, Y_val), epochs=10, cal

     # Access the loss and accuracy history
     train_loss = history.history['loss']
     val_loss = history.history['val_loss']
     train_accuracy = history.history['accuracy']
     val_accuracy = history.history['val_accuracy']

     # Convert the lists to NumPy arrays if needed
     train_loss = np.array(train_loss)
     val_loss = np.array(val_loss)
     train_accuracy = np.array(train_accuracy)
     val_accuracy = np.array(val_accuracy)
```

```
[ ]  test_loss, test_accuracy = model.evaluate(X_test, Y_test)
     print(f"Test Loss: {test_loss}, Test Accuracy: {test_accuracy}")
```

Visualize Test Loss and Accuracy in each epoch

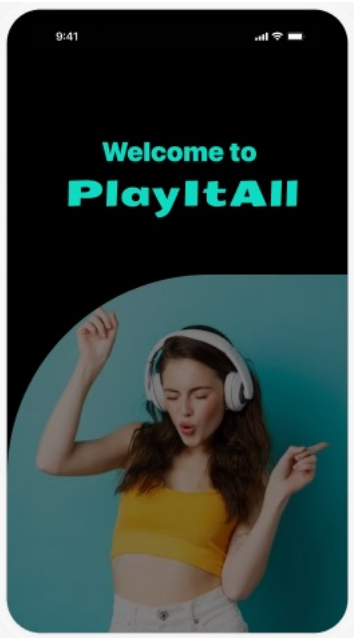


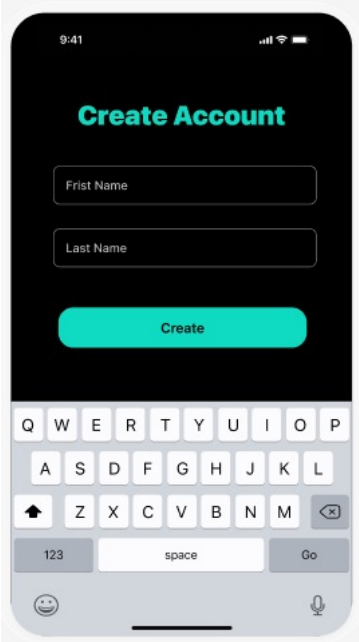

```
Average Test Loss: 0.8386545181274414
Average Test Accuracy: 0.6399999856948853
```

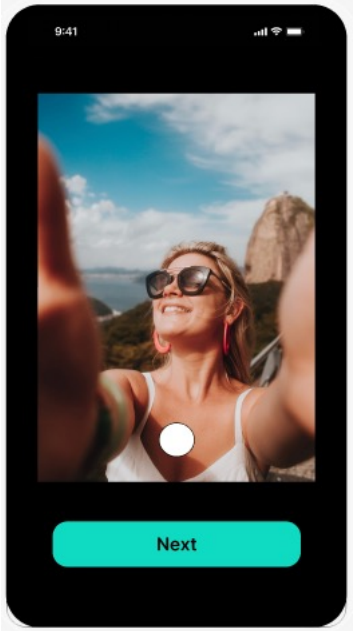
Cheking the Age

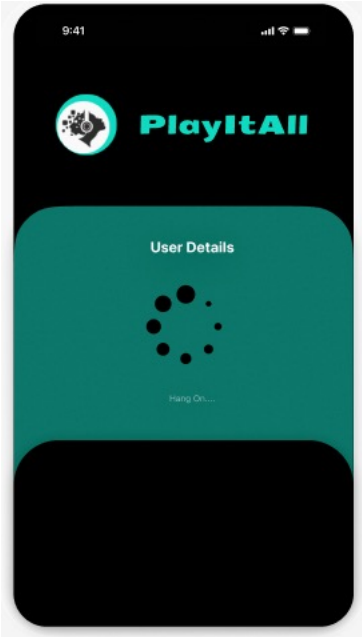
Current Progress.. (cont')

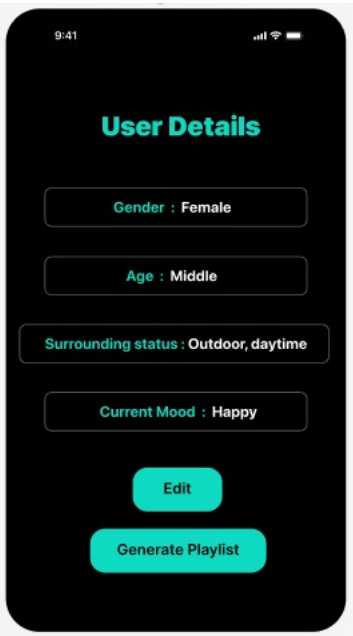
5. Create UI prototypes for the mobile application.
   - Technology - Figma, Flutter

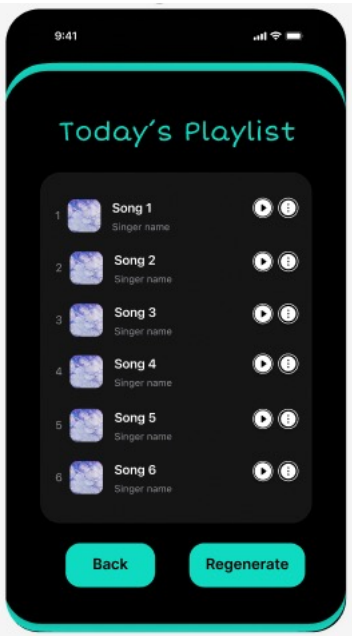












Next Steps

Accuracy Improvements

Completion of the Music Recommendation model

Overral System Integration

Any Questions ???

Thank you