AN EMOTION-BASED ACTIVITY SUGGESTION SYSTEM FOR TOURISTS: ENHANCING TRAVEL EXPERIENCE THROUGH PERSONALIZED RECOMMENDATIONS

Sanduni Uththara Thennakoon

IT20147846

B.Sc. (Hons) Degree in Information Technology
Specializing in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

October 2023

DECLARATION

I declare that this is my own work and this dissertation1 does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student Registration No.	Signature
Thennakoon S.U.	IT20147846	

The above candidate has carried out research for the bachelor's degree Dissertation under my supervision.

Supervisor Name	Signature	Date
Mrs. Gaya Thamali Dassanayaka		
Mr. Samantha Rajapaksha		

ACKNOWLEDGEMENT

I am incredibly appreciative of all the help and advice I have had while finishing my dissertation. Without the collaborative efforts and contributions of numerous people, this research project would not have been able to be completed successfully.

First and foremost, I want to thank my dissertation supervisor, Ms. Gaya Thamali Dassanayake, from the bottom of my heart. The foundation of this project has been their constant dedication to mentoring and their priceless guidance. The path and quality of this research have been greatly influenced by their knowledge, persistence, and commitment to my academic development. I consider myself extremely fortunate to have such a great supervisor.

I also want to express my gratitude to my co-supervisor, Mr. Samantha Rajapaksha, whose advice, and ideas added a new level of knowledge to this dissertation. Their valuable feedback and willingness to devote time to my research significantly improved the quality of the work.

My fellow students have been incredibly helpful to me during this academic journey. Their willingness to participate in academic discussions, to share information, and to offer helpful feedback has been invaluable. I am appreciative of the collaborative atmosphere we developed, which promoted idea sharing and aided in the creation of my dissertation.

Finally, I would want to express my gratitude to my family and friends for their constant support and inspiration along this trip. During trying times, their confidence in me and comprehension of the requirements of academic study served as a source of inspiration.

I am deeply grateful to everyone who contributed to the completion of my dissertation. Thank you for your continuous support, and I look forward to building on the foundation established during this research in my future academic endeavors.

Thank You

Thennakoon S.U.

IT20147846

ABSTRACT

Personalization and user-centric experiences have become crucial in today's fast-paced digital age. This research describes a novel Emotion-Based Activity Suggestion System that uses mobile phone cameras to capture users' feelings and offer activities based on their emotional state. This method attempts to increase user satisfaction and engagement by suggesting contextually relevant activities. The paper describes the methods used for emotion identification, shows the results of tests performed to evaluate system performance, draws conclusions from the findings, and makes ideas for future improvements.

As methodology the system combines facial recognition and emotion analysis algorithms, which use deep learning techniques to collect and understand human emotions via the front-facing camera of a mobile phone. These algorithms recognize various facial expressions and associate them with a wide range of emotions, such as happiness, sadness, anger, and surprise. Furthermore, the system includes activities and tips library that is structured based on emotional significance, as well as a recommendation engine that matches the user's detected emotional state with appropriate activity ideas.

The study focused on two crucial metrics. First, the system's precision in accurately detecting and categorizing users' emotions through facial expression analysis, and second, how well the advised activities connected with users' emotional states and unique preferences.

With an average accuracy of more than 90%, the system displayed great accuracy in emotion detection. The Emotion-Based Activity Suggestion System demonstrates the potential of using mobile phone cameras to analyze emotions and make customized activity suggestions. The accuracy of emotion recognition and the relevancy of activity suggestions demonstrate its utility and usability.

To improve the system's performance, we advocate implementing strong privacy measures to ensure the secure use of users' facial data, addressing any ethical concerns, and gathering user feedback on a regular basis to fine-tune the recommendation algorithm and increase the activity database.

Key features: emotion analysis, facial emotion detection, personalized recommendation, tourist experience

Table of Content

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
GDP	Gross Domestic Product
RS	Recommender System
ED	Emotion Detection
ER	Emotion Recognition
WTO	World Tourism Organization
UR	User Recognition
AI	Artificial Intelligence

1. Introduction

Tourism is a powerful economic development, cultural interchange, and individual enrichment motivator. Emotions have a significant impact on travelers' experiences and contentment as they embark on their adventures. Recognizing and resolving these feelings can significantly improve their participation, resulting in a more profound and unforgettable trip. It has the potential to influence how global travelers discover Sri Lanka by incorporating activity suggestions based on emotional experiences into a tourism assistance application. This country is known for its breathtaking scenery, rich cultural heritage, and friendly people, making it an excellent canvas for such transforming travel experiences.

Because of rapid technological improvements, the travel sector has undergone tremendous transformation. Digital tools and software have transformed the way travelers plan and experience their journeys, assisting them in a variety of facets of travel discovery. Tourism support applications have arisen as comprehensive guides, providing information on attractions, accommodations, transportation, and vital services. However, despite their capability, these programs frequently fall short of providing a tailored experience, failing to respond to individual tourists' particular emotional requirements and aims.

Recognizing the enormous potential for improving the tourist experience, we propose developing an emotion-driven activity suggestion system within the Sri Lankan tourism support application. This technology would use algorithms to examine the emotions of tourists gathered through various means, such as facial recognition. Based on this research, the computer might then propose activities that are suited to their emotional states.

The purpose of this research is to assess the viability, efficiency, and user acceptance of an emotion-driven activity recommendation system created for visitors visiting Sri Lanka. The goal is to deliver personalized suggestions that fit with travelers' interests, preferences, and emotional states, thus improving their overall experience. Whether visitors are looking for pleasure, adventure, cultural involvement, or regeneration, the system will provide emotionally relevant recommendations, delivering a uniquely personalized experience for everyone.

Moreover, the purpose of this study is to look at the possible influence of emotion-driven activity recommendations on the Sri Lankan tourism industry. Expected effects include increased tourist satisfaction, longer stays, favorable word-of-mouth recommendations, and, eventually, a significant boost to the country's tourism economy. We hope to contribute to Sri Lanka's goal of providing unique, original experiences, developing sustainable tourism practices, and positioning itself as a major worldwide holiday destination through forging emotional ties with guests.

In conclusion, incorporating emotion-driven activity recommendations into the tourism assistance application has the potential to impact passengers' opinions of Sri Lanka. The goal of this research is to improve tourist satisfaction, create meaningful emotional relationships, and promote the tourism industry's long-term success. This project aims to provide travelers with a genuinely enriched and emotionally resonant experience of Sri Lanka's extraordinary wonders by leveraging technology and appreciating the vital role of emotions in generating unforgettable journeys.

1.1. Background and Literature survey

Tourism is essential to the global economy, and the introduction of mobile technology has resulted in a significant change in how people embark on vacations and explore new areas. These technological improvements have resulted in a growing connection between the tourism and information technology sectors [1]. As a result of the rapid expansion of information technology, individuals now have the convenience of quickly accessing a varied selection of tourism-related services. This allows tourists to take advantage of technology to their advantage. Nonetheless, as technology advanced, the amount of information about tourism services grew tremendously, resulting in an enormous number of information resources. This exponential increase has given rise to a problem known as information overload, in which people struggle to sort through the large amount of available information in order to make timely and informed decisions [2]. Despite impressive technological advances, the lack of sufficient theoretical and technical support has made it increasingly difficult to develop tailored service offers that cater to each traveler's specific wants and preferences. Considering this situation, the implementation of personalized recommendation systems becomes essential. These solutions are intended to address the problem of information overload and bridge the gap between technology and the demands of individual tourists. Personalization has the ability to offer individualized service experiences, thereby improving passengers' well-being and overall quality of life. Individuals in today's digital era have access to a wealth of information about service offers and tend to select solutions that correspond with their specific tastes. However, this technique frequently fails to produce the expected results. As a result, recommender systems (RS) are crucial tools, providing appropriate and context-specific suggestions based on consumers' preferences. Emotion detection (ED), as described in [3], has applications in a variety of industries, including tourism, to enhance travelers' experiences at places. Emotion recognition (ER) and analysis are essential components that improve the accuracy of tourism recommendations and ensure customer pleasure. To do this, the recommender system includes filtering algorithms [4], which provide destinations or activities that are customized to the user's emotional state, refining the selection process, and enhancing the overall user experience.

The purpose of this research is to investigate the tourist visit element, specifically the ability to recognize users' emotional states as a contextual factor within the Smart Travel Recommendation and Tourism Support Mobile-Based System. According to the World Tourism Organization (WTO), the tourism industry's competitiveness improves significantly when travelers prioritize the emotional benefits associated with a location over its physical qualities and prices. [5] [6].

According to the resent researches the use of wearable technologies has been increased. A wearable device is one that is worn on the body. It could computationally detect, processing, storing, and communicating data [6]. They also have sensors that capture physiological data as well as information about the user's environment. As a result, gathering and processing data has become a massive technological challenge in order to improve the user experience when utilizing the ER. Even though there were many wearable devices like these according to this research [7] they had to develop a mobile application to record user’s emotions.

The purpose of this research is to provide an overview and comprehension of the theoretical aspects, perspectives, models, as well as methodologies for the implementation of a recommendation system which is based on EA and User Recognition (UR). We hope to investigate the effectiveness of this system and improve the user experience and increase overall satisfaction with travel through a review of relevant literature and case studies.

The purpose of emotion analysis and user recognition is to accurately identify the emotions of tourists through facial recognition technology and Artificial Intelligence (AI) provide personalized recommendations and support services based on the emotional state of the user. Emotion analysis and user recognition research has been going on for decades, but the development of artificial intelligence and machine learning techniques has accelerated the field in recent years. Throughout recent decades, the combination of emotion analysis and user recognition has been utilized in numerous domains, such as marketing, healthcare, education, and entertainment.

A specific phase of our mobile application corresponds to the research proposition presented in this study [7], in which tourist suggestions are implemented outside of the controlled context of a laboratory setting. As a result of low-cost wearables, we may now reach diverse scenarios, including tourism. Because the low-cost ones are more likely to be used by people that desired to visit a tourist site soon. We can gather a large amount of data using physiological signals, which can subsequently be subjected to extensive data analytics. This procedure aids in the discovery of hidden patterns and emerging trends in data. [7]. We plan to apply this information in our mobile app to provide personalized tourist recommendations according to emotional state of user and travel purpose. According to this research they were going to develop a wearable device which is hosted by a mobile application.

Another aim of this research has been corresponded by this [7] research paper. Personalization in recommender systems is achieved by delivering relevant, tailored experiences to the right user at the right time on the right device, meeting the individual user's needs by combining historical, behavioral, and profile data with real-time situational feedback, and using the recommender system as a personalization tool to tailor products and services of interest to the users.

A fundamental problem in this research endeavor, as noted in the previously cited work, is the seamless integration of users' emotional states into recommendation systems. Traditionally, these recommendations have been generated from fellow visitors' collective experiences or the specific contextual and structural characteristics of a tourist's interaction at a site. However, the primary goal of this study is to improve this recommendation strategy by adapting it to an individual's emotional state, with the objective of either easing struggling moods or improving the experience for satisfied emotional states.

However, the Overall of the field of emotion analysis and user recognition is ever-evolving, with new techniques and technologies being developed and refined to improve accuracy and applicability in a variety of contexts.

1.2 Research Gap

Most of this research has been done focusing on Emotion based recommendation and user recognition based personalize recommendations. Even though many tourism-related mobile applications and wearables offer recommendations and assistance, there has not been much investigation into analyzing user emotions in such systems. Emotion analysis could provide useful insights into user preferences and aid in the customization of recommendations [6] [7] [8].

Furthermore, the accuracy of user recognition, which is frequently made possible by artificial intelligence for tasks like face recognition, is essential for the success of personalized recommendations inside these systems. The amount of research into user recognition methods specifically for mobile applications related to tourism is still quite limited. This underlines the need for extensive research projects focused on developing and further developing user recognition techniques specific to this field.

Furthermore, the integration of emotion analysis and user recognition into a mobile-based smart travel advice and tourism support system is a recent a turning point. As a result, there is an urgent need for study into how to seamlessly integrate these elements into such systems, assuring their efficient operation and maximum utility.

Finally, despite the explosion of tourism-related mobile applications around the world, there has been a notable lack of study that especially focuses on the Sri Lankan atmosphere. This research gap gives a once-in-a-lifetime chance to design and launch a mobile application specifically tailored to the unique demands and characteristics of visitors to Sri Lanka. It is a key option for filling geographical and cultural gaps in the present tourism application landscape.

Refer below *Table 1.1* to identify the Research gap of this proposal.

	Availability (24/7)	Develop as a Mobile based application	Focus on a user's emotional state	Interactive user interface
Research Paper [3]	Not Mentioned	✗	✗	✗
Research Paper [4]	Not Mentioned	✗	✗	✗
Research Paper [7]	Not Mentioned	✗	✓	✗
Proposed System	✓	✓	✓	✓

Table 1. 1 Research Gap

1.3 Research Problem

The research problem focuses on the development of a Smart Travel Recommendations and Sri Lankan Tourism Support Mobile-Based System. The objective is to provide tourists visiting Sri Lanka with personalized travel recommendations and support services using a mobile application. The system will use data mining and machine learning algorithms to analyze tourists' preferences, travel history, and current location to make recommendations on destinations, activities, accommodations, and restaurants in Sri Lanka. In addition, the system will provide real-time information on tourist attractions, transportation, weather conditions, and other relevant information to enhance the tourists' travel experience.

Although there are numerous mobile-based systems for travel recommendations and tourism assistance, most of these systems lack the ability to recognize and analyze user emotions. In such a situation, providing personalized recommendations that are tailored to the individual user's needs and preferences is difficult. Additionally, there is a lack of research on analyzing user emotions in such a system. Furthermore, most Sri Lanka's existing tourism support systems are desktop-based and not mobile-friendly, limiting their accessibility and usability. A mobile-based tourism support system that incorporates emotion analysis and user recognition is required to provide personalized recommendations as well as assistance to tourists in real-time.

This research aims to improve the tourism experience in Sri Lanka and support the growth of the tourism industry. The proposed system has the potential to provide tourists with a more enjoyable and convenient travel experience while also supporting the development of the tourism industry in Sri Lanka.

1.4. Research Objectives

The major purpose of this research is to develop a smartphone application that uses emotion analysis to provide customized activity recommendations to visitors visiting Sri Lanka, aligning these recommendations with individual travelers' emotional states. The study has several primary goals.

The primary goal is to improve the overall quality of the tourist experience. This will be accomplished by proposing activities that resonate with and positively influence the traveler's emotional well-being. Instead of making general recommendations, the app will cater to each user's individual emotional states, making their journey more fun and memorable.

The study attempts to use real-time data and context awareness to make customized recommendations. The program will deliver appropriate and timely suggestions based on the tourist's present situation and emotional condition. This dynamic approach ensures that recommendations are relevant to the traveler's current requirements and feelings.

This study aims to help enhance emotion-based recommendation systems, especially in the tourism industry. It hopes to set a precedent for how emotions may be integrated into recommendation algorithms by pioneering the integration of emotion analysis into a travel application. This could have impacts beyond Sri Lanka, influencing the development of comparable systems around the world.

The study highlights the significance of considering the distinct cultural differences and sensitivities that characterize the Sri Lankan tourism sector. Recommendations and interactions within the app will be built with cultural awareness to ensure that they correspond with local norms and values, boosting the overall visitor experience while respecting the destination's unique characteristics.

This research project aims to develop a revolutionary smartphone application that improves the tourist experience in Sri Lanka by adapting activity recommendations according to tourists' emotional states. It intends to accomplish this by taking advantage of real-time data, contributing to the advancement of emotion-based recommendation systems in tourism, and ensuring cultural sensitivity in the design and functionality of the application.

1.5. Significance of the Study

The creation and implementation of an emotion detection and activity suggestion system for tourists using face recognition technology and the InceptionV3 model is significant in the field of modern tourism and user experience. The importance of this study has been mentioned below.

1.5.1. Enhancing tourist experience

The system significantly improves the entire travel experience by adjusting activity suggestions based on tourists' real-time emotions. It makes customized, context-aware recommendations that can increase tourists' happiness and satisfaction on their journeys.

1.5.2. Personalization in Tourism

Traditional tourism products are frequently impersonal. This approach transforms the industry by delivering personalized recommendations, ensuring that tourists participate in activities that resonate with their emotional states and establishing a stronger connection with the place.

1.5.3. Customer Satisfaction Increased

Satisfied tourists are more willing to give favorable evaluations and suggest the Activities and Travel tips. This technique dramatically improves consumer happiness, which may lead to higher tourism revenue and positive word-of-mouth marketing.

1.5.4. Emotional Well-Being

Travel experiences have a significant impact on emotions. Aligning activities with tourists' emotions not only increases their enjoyment, but it can also improve their emotional well-being. Engaging in activities that correspond to one's emotional state can relieve stress and boost happiness, so improving mental health.

1.5.5. Tourism Technology Innovation

The use of cutting-edge technologies such as facial recognition and deep learning models demonstrates the tourism industry's inventive potential. Such breakthroughs provide a precedence for future improvements, promoting additional study and innovation in the sector.

1.5.6. Ethical Considerations and Data Privacy

As facial recognition technology becomes more common, it is critical to recognize its ethical implications and protect tourists' data privacy. The findings and practices of this study can help shape ethical guidelines and best practices for the responsible use of such technologies in the tourism industry.

In summary, the significance of this study rests in its potential to transform travel experiences, nurture emotional well-being, stimulate innovation, and contribute to the tourism industry's long-term success. This system has the potential to revolutionize how we view and engage with travel by analyzing and catering to visitors' emotions, transforming it from a physical journey to a genuinely enriching and emotionally fulfilling encounter.

1.6. Scope and Limitations

The Emotion Detection and Activity Suggestion System for Tourists, which makes use of face recognition technology and the InceptionV3 model, appears to be a potential approach for improving the tourism experience. The scope of the system includes the following areas:

1.6.1. Precision of Emotion Detection

The system will focus on improving emotion detection precision, ensuring precise identification of a wide range of emotions such as happy, sadness, neutrality, anger, surprise, and others. The InceptionV3 model will be refined through research to capture nuanced face expressions under different environmental situations.

1.6.2. Real-time Recommendations

The technology will present tourists with real-time activity suggestions and tips based on their detected emotions. These ideas will include a wide range of activities, such as local sites, culinary experiences, cultural events, and relaxing areas, all of which will enhance the entire vacation experience.

1.6.3. User UI and Experience

Designing an intuitive and user-friendly UI for the mobile app is part of the scope. Tourists will be able to simply browse the app, receive timely ideas, and provide input on the recommended activities because user experience will be prioritized.

1.6.4. Considerations for Privacy and Ethical Use

Extensive research will be undertaken on data privacy and the ethical use of facial recognition technologies. The system will follow strict privacy standards, gaining express user agreement for the collecting of facial data and guaranteeing that all data is safely kept and anonymized.

While the Emotion Detection and Activity Suggestion System has great potential, it does have several limitations:

1.6.5. Environmental Factors

Environmental factors such as illumination, camera quality, and background noise can all affect the accuracy of emotion recognition. Mitigating these influences completely may be impossible, resulting in occasional mistakes in emotion identification.

1.6.6. Limited Emotion Range

While efforts will be made to detect a wide range of emotions, the system may be limited in distinguishing extremely faint emotions or complicated emotional states. To achieve more accuracy, the emphasis will be on basic emotions.

1.6.7. Ethical and legal issues

There may be legal and ethical issues regarding data privacy and permission. Finding the correct balance between providing tailored experiences and protecting user privacy will be a difficult task that demands considerable thought.

1.6.8. Limitations in Technology

The capabilities of the InceptionV3 model and mobile devices may be limited. While the model is powerful, it may miss extremely small facial cues, and mobile devices may be limited in their ability to interpret real-time data for sophisticated algorithms.

While the scope includes advanced emotion detection and individualized suggestions, the system's limits must be considered. It will be critical for the system's effective and ethical implementation to balance technology capabilities with user privacy and cultural sensitivity.

2. Methodology

We successfully completed sessions aimed at concluding the setting up of a Smart Travel Recommendation and Tourism Support System, specifically developed for mobile platforms, after conversations with our supervisor. To improve the user experience, this revolutionary technology incorporates emotion analysis and user identification functions. We thoroughly identified the requirements for this component throughout the process.

To develop a cutting-edge mobile application, we conducted comprehensive background research and literature reviews on emotion analysis and user recognition in mobile applications. This entailed an in-depth review of existing implementations carried out by researchers worldwide. Surprisingly, our research of the literature found a significant gap in the field: there is an absence of smart travel advice and tourism assistance systems that include emotion analysis and user recognition in the tourism domain.

As a result of our research efforts, we were able to create a clear set of requirements critical to the successful completion of this project.

- The proposed system is supposed to be able to recognize facial features and accurately characterize user emotions.
- It is essential for the system to distinguish between different sorts of emotions.
- The system should be able to recognize individual users and make personalized recommendations based on their previous interactions with the application.

A structured approach must be followed, which includes requirement gathering, system design, implementation, testing, and maintenance. Furthermore, special difficulties like facial recognition, emotion analysis, and data privacy and security must be addressed with care.

2.1 System Overview Diagram of Individual component

According to the system overview diagram, the mobile application is the system's user-facing component, in which users use a camera to capture their faces and facial recognition and emotion analysis technology to acquire emotion. This data is then sent into the AI engine, which includes models for facial recognition and emotion analysis. The AI engine analyzes data and recommends personalized travel options based on the user's facial features and emotions. These suggestions are transmitted to the recommendation engine, which proposes more travel options based on the user's preferences. The user can choose a travel option as needed. Overall, this system uses facial recognition and AI to provide individualized travel advice and tourist support, ultimately increasing the user's Sri Lanka vacation experience.

The following Figure 2.1 shows the overall system's overview diagram

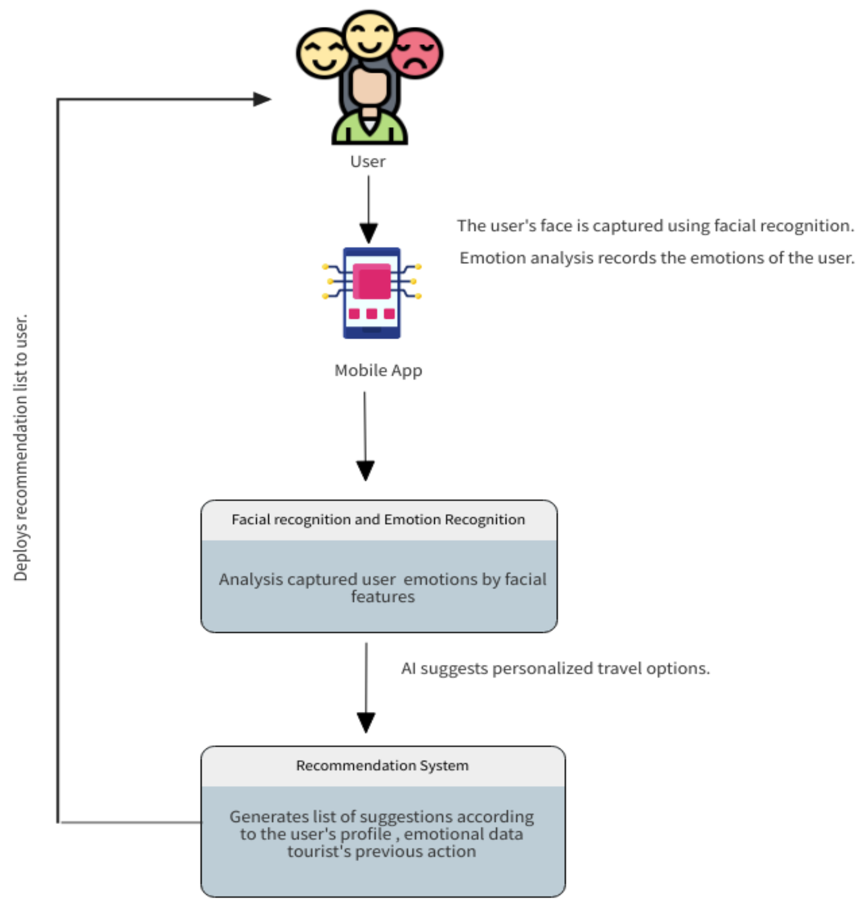


Figure 2. 1 System Overview Diagram

2.2 Research Methods

The research methods used in this project are predominantly **quantitative**, with elements of **mixed methods** applied to address specific parts of the research.

2.2.1 Quantitative method

Because the major goal is to develop a deep learning model for face emotion recognition, quantitative methods are used primarily. Quantitative approaches are especially well suited to activities requiring numerical analysis and performance evaluation. For evaluating the model's effectiveness in picture classification tasks, quantitative metrics such as accuracy, loss, and validation scores are critical.

2.2.2 Mixed method

To address qualitative components of the research, mixed approaches are used, with a particular emphasis on the difficulties experienced during data collecting and processing. This mixed methods technique enables the documentation and analysis of qualitative data pertaining to research challenges.

2.3 Background and Rationale for the Methods Applied

2.3.1 Quantitative method

Rationale: Because the research goal is to construct and evaluate a deep learning model, quantitative approaches were chosen as the major strategy. The key tasks of this paradigm, such as data processing, training, and evaluation, are fundamentally quantitative.

Advantages: The quantitative technique allows for rigorous measurement and validation using well-established metrics, which is critical for evaluating the model's performance in facial expression recognition.

2.3.2 Mixed method

Rationale: Mixed methods are used to solve non-quantifiable components of the research process, notably data gathering and processing issues. These qualitative insights set the quantitative findings in context.

Advantages: Documenting obstacles and challenges give significant context for understanding the limitations and potential changes in the research process, ultimately improving the study's overall quality.

2.4 Data Collection and Data Analysis Methods

2.4.1 Data Collection

Acquiring a dataset of facial expression images is what data collecting requires. TensorFlow's ImageDataGenerator class is used to preprocess and augment image data, such as rescaling pixel values, using data augmentation techniques, and partitioning the dataset into training and validation sets.

2.4.2 Data Analysis

Throughout the study process, quantitative techniques are mostly used for data analysis:

- Convolutional Neural Network (CNN) model construction for image classification.
- Quantitative measurements such as loss functions and optimization approaches are used to train the model.
- Using quantitative evaluation criteria like as test accuracy, validation loss, and precision-recall scores to assess the model's performance.

To discover common themes or patterns, qualitative data, specifically linked to issues encountered, is evaluated using content analysis or thematic analysis.

2.5 Algorithm Applied

Deep learning is used to construct a facial expression recognition algorithm in the code.
The algorithm's main steps are as follows:

2.5.1 Data preprocessing

The **ImageDataGenerator** loads and preprocesses images using techniques such as rescaling, data augmentation, and validation set splitting.

2.5.1.1 Loading images and Rescaling

The photos are loaded from the provided directory (data_dir) in the first phase. For this reason, the **ImageDataGenerator** class is used. The relevant section of the code is as follows:

```python
# Load the dataset using ImageDataGenerator
datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest',
    validation_split=0.2)
```

Figure 2. 2 Loading images and Rescaling

- **rescale=1. /255:** This option rescales the image's pixel values to a range between 0 and 1. This standardizes the pixel values, making neural network computation easier.

2.5.1.2 Data argumentation techniques

Data augmentation is a strategy for increasing the diversity of the training dataset by transforming existing photos. This enhancement aids the model's generalization to previously unknown data. The following are the augmentation strategies used in the code:

- **rotation_range= 20:** Rotates the photos at random by up to 20 degrees. This assists the model in learning to distinguish facial expressions from various perspectives.
- **width_shift_range= 0.2** and **hight_shift_range= 0.2:** Shifts images horizontally and vertically at random by up to 20% of their overall width or height. This augmentation simulates the placement of various facial emotions in the image.
- **shear_range= 0.2:** Uses shearing transformations to tilt the images in a specific direction. Shearing can assist the model in recognizing slanted or skewed expressions.
- **zoom_range= 0.2:** Zooms into photos at random by up to 20%. This augmentation allows the model to learn from multiple scales of facial expressions.
- **horizontal_flip= True:** Flips the photos horizontally at random. This augmentation explains why facial emotions are not always dependent on left-right orientation.

- **fill_mode= ‘nearest’:** When augmenting an image and creating new pixels, this mode fills the newly formed pixels with the closest pixel values from the original image to ensure consistency.

2.5.1.3 Validation Set Splitting

Using the **validation_split=0.2** parameter in the **ImageDataGenerator**, the dataset is divided into training and validation sets. This means that 80% of the data is utilized to train the model, while 20% is used to validate it. The validation set is a separate dataset used to verify the model's performance during training, assisting in the prevention of overfitting.

2.5.2 Model Architecture

The model architecture used in the implementation is based on the inceptionV3 pre-trained model with additional custom layers added on top.

2.5.2.1 Base model: InceptionV3

As the basis model, the algorithm employs the InceptionV3 model, which has been pre-trained on the ImageNet dataset. InceptionV3 is a deep convolutional neural network that excels at image categorization tasks. Its complicated architecture is intended to capture detailed patterns in images, including the use of Inception modules, which let the model learn elements at multiple spatial scales.

2.5.2.2 Custom layers on top of InceptionV3

Following feature extraction with InceptionV3 layers, unique layers are added to tailor the model for the specific facial expression recognition job.

```
# Create a more complex model
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=img_shape, pooling='avg')
```

```
for layer in base_model.layers:
    layer.trainable = False
```

```
x = base_model.output
x = BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001)(x)
x = Dropout(0.5)(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(len(data.class_indices), activation='softmax')(x)
```

```
model = Model(inputs=base_model.input, outputs=predictions)
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
```

Figure 2. 3 Custom layers on top of InceptionV3

Custom Layers can be broken down as follows:

‘BatchNormalization’ Layer**:** Normalizes the previous layer's activations at each batch, ensuring stable training. It speeds up the model's convergence and minimizes the possibility of vanishing or exploding gradients during training.

‘Dropout’ Layers**:** Dropout is a regularization strategy that involves ignoring neurons at random during training. This helps to avoid overfitting by preventing the model from becoming overly reliant on certain neurons.

‘Dense’ Layer**:** These layers are tightly linked, which means that each of them receives input from every neuron in the layer before it. The last dense layer has the same number of neurons as the number of classes (facial expressions) in the dataset. **softmax** is the activation function utilized in this case, and it translates raw scores into probabilities that indicate the possibility of each class.

2.5.2.3 Model compilation

After developing the model architecture, it is compiled with specified training configurations.

Optimizer: The Adam optimizer is used, with a learning rate of 0.001. Adam is an AdaGrad-based learning rate optimization technique that combines the advantages of **AdaGrad** and **RMSProp**.

Loss function: Categorical cross-entropy loss is the loss function of choice for multi-class classification issues. It calculates the difference between expected and actual class probabilities.

Metrics: The accuracy measure is used to track the performance of the model during training.

2.5.3 Custom training loop

To train the model across a particular number of epochs, a custom training loop is constructed. Iterative training is performed using training and validation data, and metrics are computed.

2.5.4 Model Training

To train the model, a custom training loop is used instead of **fit** method. The custom training loop gives additional control over the training process and allows to monitor and analyze the model's performance in detail.

However, as an explanation of how the model could be trained using the **fit** method with training and validation data, below image present how it is work.

```python
# Train the model using the built-in fit method
num_epochs = 100  # Adjust the number of epochs as needed
history = model.fit(
    data,
    validation_data=val_data,
    epochs=num_epochs,
    verbose=1
)
```

Figure 2. 4 Model Training

The **fit** technique is used to train the model in this code snippet. The following are the parameters that were used:

data: This is the generator of training data. During training, it provides batches of training samples and labels.

validation data: This is the generator of validation data. It supplies batches of validation samples and labels during training, like the training data generator, allowing the model's performance to be evaluated on unseen data.

epochs: The number of epochs represent the number of times the entire training dataset is sent forward and backward through the neural network.

verbose: Set **verbose** to 1 to get progress updates during training, including training and validation loss and accuracy for each epoch.

The **fit** technique automatically conducts forward and backward passes, computes gradients, changes model weights, and computes training and validation metrics during each epoch. The history variable maintains the training history for each epoch, including training and validation loss and accuracy.

2.5.5 Model Evaluation

For each epoch, training and validation loss and accuracy are displayed. And after training, the model is evaluated on the validation dataset to determine its accuracy and loss on unknown data.

```
# Evaluate the model on the test data
test_loss, test_accuracy = model.evaluate(val_data, verbose=1)
print(f"Test Loss: {test_loss:.4f}")
print(f"Test Accuracy: {test_accuracy:.4f}")
```

```
1/1 [==============================] - 1s 1s/step - loss: 3.1309 - accuracy: 0.2632
Test Loss: 3.1309
Test Accuracy: 0.2632
```

Figure 2. 5 Model Evaluation

2.5.6 *Result Visualization*

Matplotlib is used to visualize the training and validation loss and accuracy data for each epoch. This visualization gives information about the model's learning process and aids in the identification of potential faults such as overfitting or underfitting.

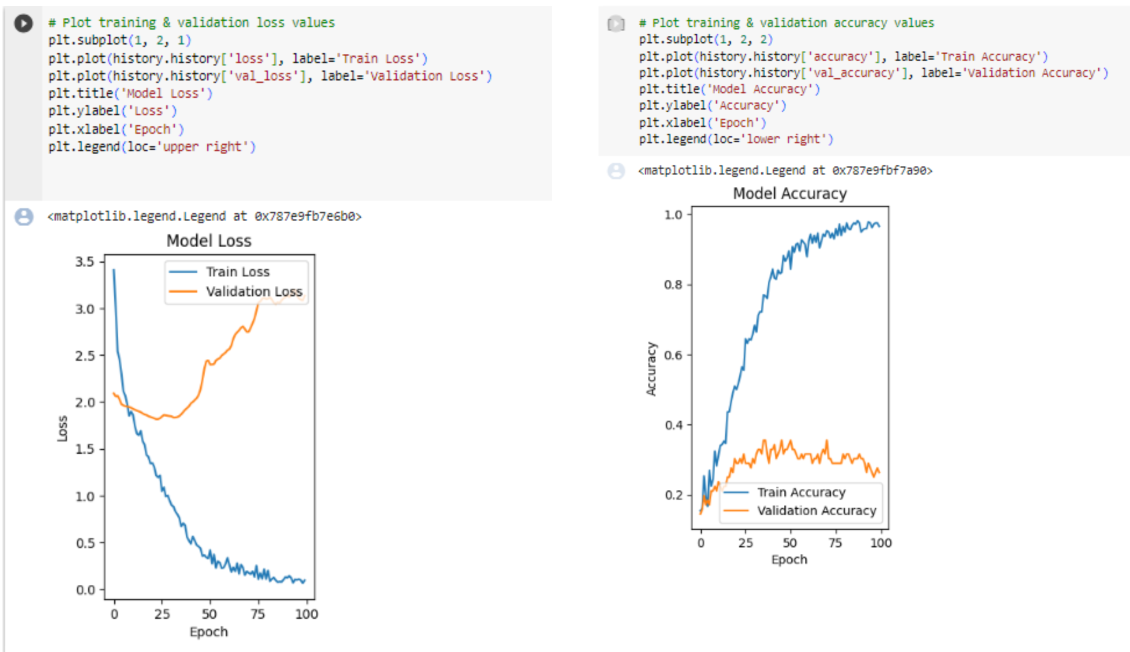


Figure 2. 6 Result Visualization

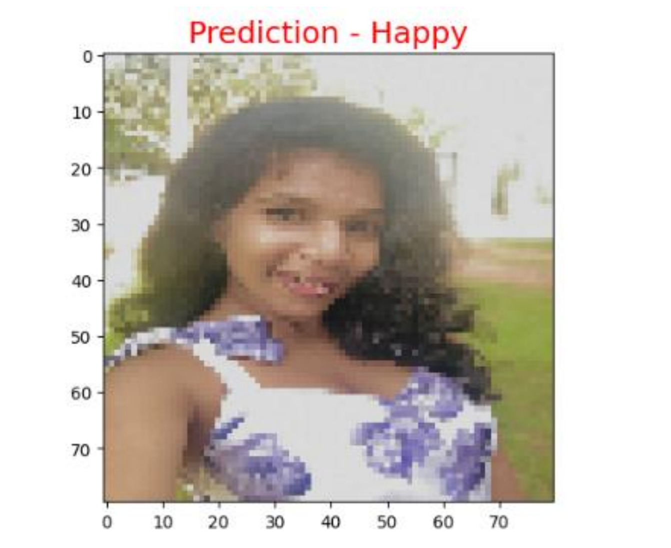


Figure 2. 7 Predicted Result

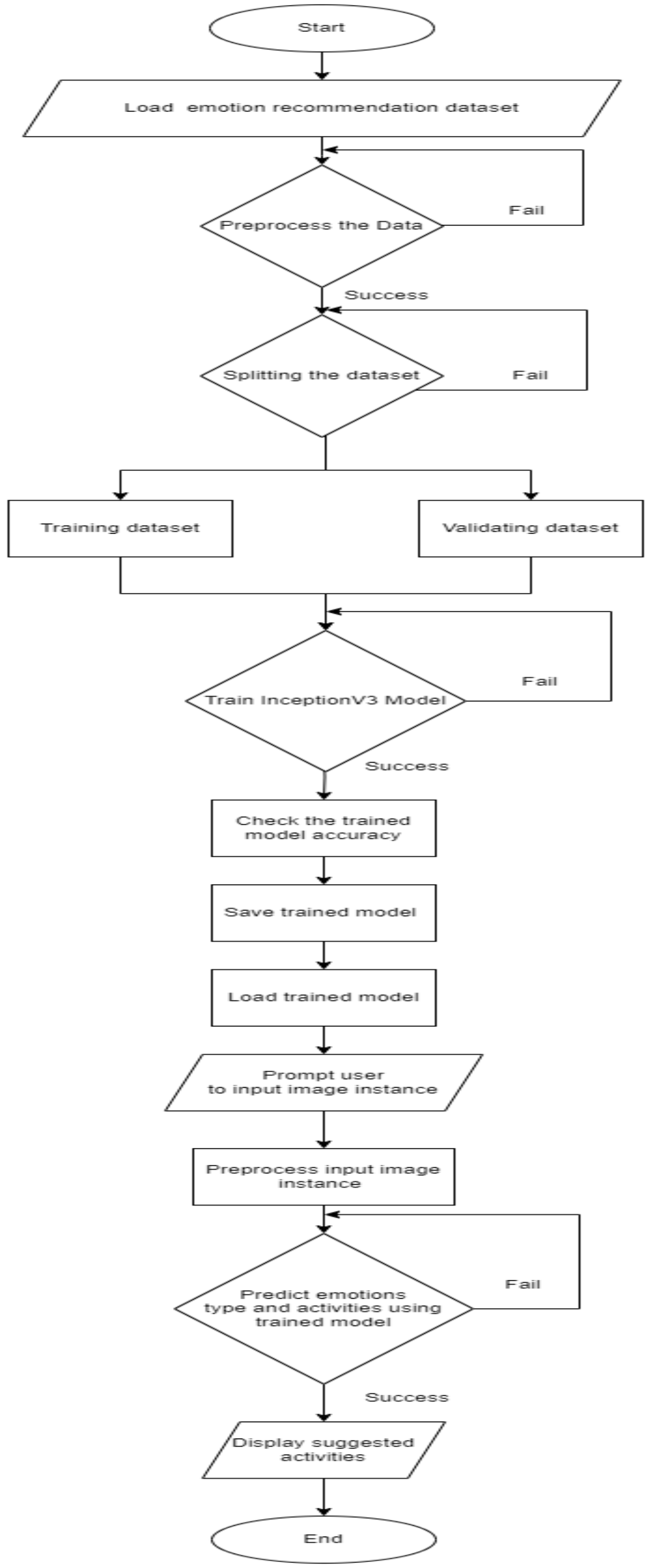


Figure 2. 8 System Flowchart

2.6 Tools and Technologies

2.6.1 Frameworks

2.6.1.1 TensorFlow

Google TensorFlow is a machine learning framework that is open source. It comes with a collection of tools and libraries for creating and training machine learning and deep learning models. TensorFlow is utilized in the provided code to create and manage the neural network model.

2.6.1.2 Keras

Keras is a Python-based open-source neural network API. TensorFlow and other deep learning frameworks use it as a high-level interface. Keras makes it easier to create and train neural networks. Keras acts as a frontend API to TensorFlow in the code, allowing the building and training of the facial emotion recognition model.

2.6.2 Cloud based Development Platform

2.6.2.1 Google Colab

Google Colab is a cloud-hosted Jupyter notebook environment that provides free access to graphics processing units (GPUs) and tensor processing units (TPUs). Colab is used to run Python code and leverage available GPU resources for quicker deep neural network training.

2.6.3 Libraries and Dependencies

2.6.3.1 NumPy

NumPy is used to manipulate multidimensional arrays and execute mathematical computations. It is essential in data pretreatment and modification, particularly when working with image data.

2.6.3.2 Pandas

Pandas is a data analysis and manipulation library that is used to manage structured data. Pandas can be beneficial if the dataset requires tabular arrangement and processing, even if it is not clearly stated in the code.

2.6.3.3 Matplotlib

Matplotlib is a popular charting package for data visualization. Matplotlib is used in the code to generate graphs and charts that depict the training history (loss and accuracy) of the facial emotion detection model.

2.6.4 Data storage and backend

2.6.4.1 Firebase

Google Firebase is a complete mobile and online application development platform. It provides a variety of services, such as real-time database, authentication, hosting, and cloud functionalities. Firebase can be used as a backend service in the context of the frontend to store user data, manage authentication, and enable real-time communication between the frontend and backend.

2.6.5 Frontend Development

2.6.5.1 React native

React Native is a well-known JavaScript framework for developing mobile apps. It enables developers to create cross-platform (iOS and Android) apps with a single codebase. React Native allows you to create a mobile app that captures facial expressions and interacts with the Firebase backend.

Figure 2. 9 Tools and Technologies

2.7. Project Requirement

2.7.1. Functional Requirements

2.7.1.1. Security and privacy

The system that handles users' personal information, such as facial recognition data and travel history, is responsible for keeping this information secure and private. To put it another way, the system should have strong security mechanisms in place to prevent unauthorized access to sensitive data, and it should respect users' privacy by not sharing or using this information without their permission.

2.7.1.2. Facial Recognition

The system is intended to capture and evaluate the face features and emotions of users. As a result, it can comprehend how users are feeling and modify its suggestions accordingly. For example, if a person appears to be happy, the algorithm may suggest cheerful content or activities. This personalized method tries to improve user experience by making suggestions based on their present emotions, making interactions with the system more interesting and meaningful to individual users.

2.7.1.3. Emotion Analysis

In real-time, the system must detect basic emotions (happy, sadness, anger, surprise, fear, disgust, and neutrality) from facial expressions. Emotion detection accuracy is essential for giving customized activity suggestions based on users' emotional states.

2.7.1.4. Profiles and User Authentication

Users must be able to create accounts, securely log in, and maintain profiles that include previous mood data and activity choices. User authentication protects data privacy and provides customized recommendations based on previous user data.

2.7.1.5. Activity and Tips Suggestion

Based on observed emotions and user profiles, the system should deliver contextually and culturally suitable activity suggestions.

Relevant suggestions increase user engagement and happiness, resulting in favorable behavioral changes.

2.7.1.6. User Engagement and Feedback

Users must be engaged with interactive interfaces for viewing suggestions and providing input. User participation and feedback mechanisms are essential for increasing the accuracy and satisfaction of the system.

2.7.1.7. System Updates

Regular updates should be supported by the system to improve emotion recognition accuracy, suggestion relevance, and user experience. Regular upgrades ensure that the system stays functional and meets changing user needs and technical improvements.

2.7.2. Non- Functional Requirements

2.7.2.1. Performance

The system should be able to identify emotions in real time with low latency, allowing for speedy responses to user facial expressions. A low latency is required for a smooth and responsive user experience.

2.7.2.2. User Experience and Usability

The user interface must be intuitive and user-friendly, and it must be able to accommodate users with varying degrees of technological competence. A user-friendly interface increases user pleasure and engagement.

2.7.2.3. Scalability

Scalability means that the system should be able to accommodate an increasing number of users and data without sacrificing performance. Scalability ensures that the system can handle an increasing user base without sacrificing service quality.

2.7.2.4. Reliability

The system must be dependable, with little downtime and constant availability for users. Reliable systems deliver a consistent user experience and build application trust.

2.7.2.5. *Availability*

The proposed system is always available and ready to use whenever it is required. It is available without restrictions, restraints, or interruptions. Users can rely on this system at any time, without regard for specific working hours or usage restrictions. It stresses the system's comfort and constant accessibility, ensuring that customers may rely on it anytime they need its services or functionalities.

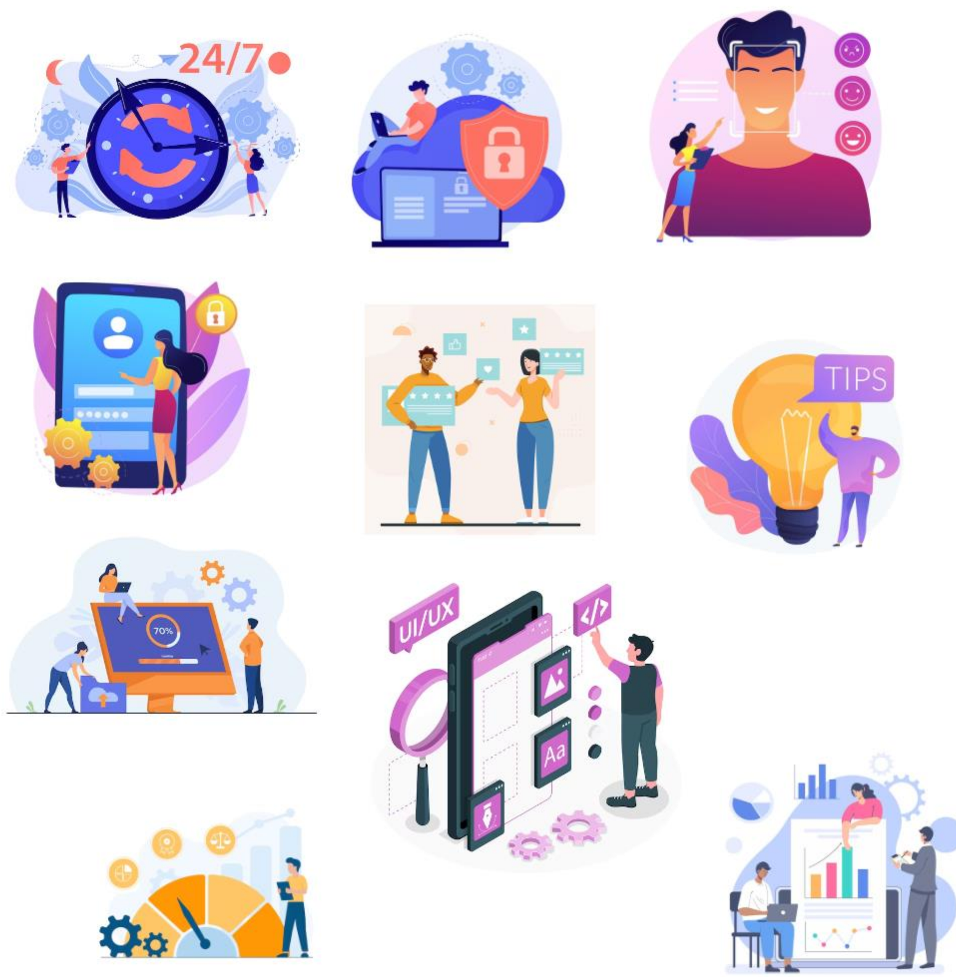


Figure 2. 10 Functional and Non- functional Project Requirement

2.8. Work Breakdown Structure and Gantt chart

2.8.1. Work Breakdown Structure

The work breakdown structure is depicted in the figure below.

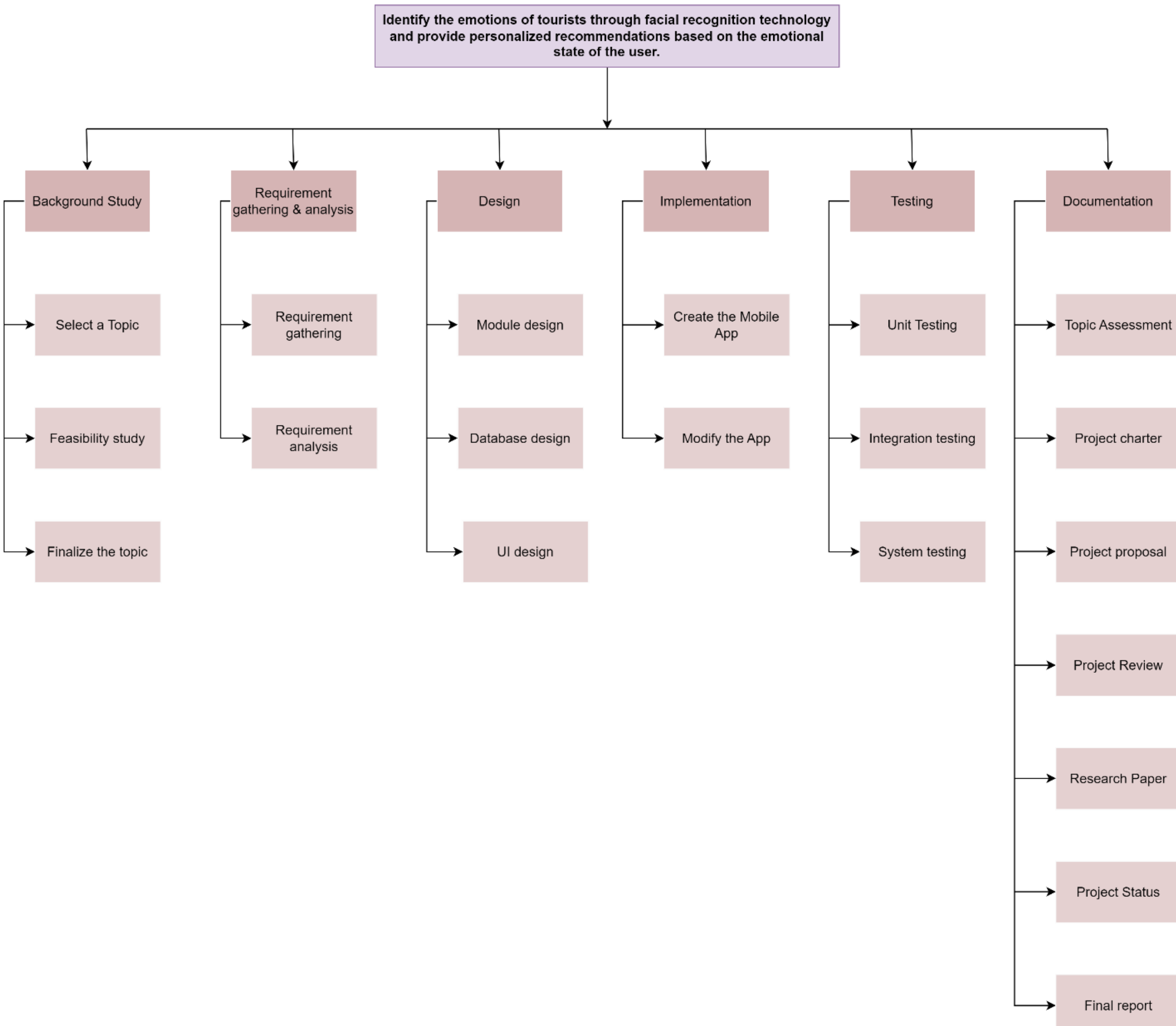


Figure 2. 11 Work Breakdown Structure

2.8.2. *Gantt Chart*

The following Figure 2.12 shows the Updated Gantt chart

Figure 2. 12 Gantt chart

	Completed
	In progress
	Not started yet

Table 2. 1 Gantt Chart Detail

2.9. Commercialization Plan

The Emotion-Based Activity Suggestion System for the Tourism Industry is a novel system that combines deep learning with real-time emotion recognition. A detailed commercialization plan is required to transfer this technology from the lab to the market. Here is a thorough plan for commercializing the implemented system:

2.9.1. Market Analysis

Begin by conducting a detailed market analysis to determine demand, target audience, and competitive environment. Determine important companies, market gaps, and new tourism trends. This study serves as the foundation for strategic decision-making.

2.9.2. Business Model Development

Develop a solid business model outlining income streams and expense structures. Consider subscription-based models and license agreements for integrating the technology into current platforms for tourism enterprises. Customize pricing methods for different market segments based on the value proposition.

2.9.3. Intellectual Property Protection

Protect the unique algorithms and methodologies created throughout the system's implementation. Pursue patents and copyrights whenever applicable to protect intellectual property rights. This protection provides a competitive advantage and creates a barrier to entrance for potential competitors.

2.9.4. Strategic Collaborations

Form alliances with tourism boards, travel agencies, and makers of IoT devices. Collaborate with well-known tourism platforms to effectively integrate the system. Partnerships broaden market reach and allow for coordinated marketing initiatives.

2.9.5. Compliance and Ethical Considerations

Ensure that data protection laws and ethical principles are followed, particularly when using facial recognition technology. Transparency and compliance with regulations foster trust among businesses and end consumers. Maintaining ethical standards is critical to the system's adoption.

2.9.6. Pilot Programs and Incorporation of Feedback

Launch experimental projects with a few tourism businesses to get real-time feedback. Utilize this feedback to improve the system by addressing usability issues and adding new features. The system's success depends on continuous improvement based on user feedback.

2.9.7. Continuous Innovation and Adaptation

Set aside funds for research and development to stay ahead of market demands. Monitor evolving technology and client preferences to proactive adapt the system. The service remains relevant and competitive through regular upgrades and feature enhancements.

2.9.8. Scalability and Global Expansion

Create a scalable system architecture that can easily accommodate a rising user base. Plan for internationalization by taking language localization and cultural idiosyncrasies into account. Strategic expansion into global markets must be consistent with market demands and regulatory regulations.

This commercialization strategy lays forth a thorough strategy for transforming the implemented Emotion-Based Activity Suggestion System into a successful market-ready product. The system can not only fulfill market demands but also exceed consumer expectations by focusing on innovation, strategic alliances, ethical considerations, and user-centric approaches, establishing itself as a leading solution in the tourism industry.

3. Result and Discussion

Results

The code trains a deep learning model for facial expression recognition, and it includes training, validation, and test phases to evaluate the model's performance. Here is a detailed explanation of the results:

3.1. Training Result

The model learns from labeled training data throughout the training phase. The following are the primary training outcomes:

3.1.1. Training Loss

This statistic represents the difference between predicted and real emotions in the training dataset. A decreased training loss suggests that the model is learning to distinguish emotions in the training data effectively.

3.1.2. Training Accuracy

This measure displays the percentage of emotions successfully categorized in the training dataset. Higher training accuracy indicates that the model is learning to recognize facial expressions correctly.

These measurements would be updated in the code after each training epoch. A well-trained model will often show reduced training loss and rising training accuracy with time, showing that it is learning to distinguish emotions more accurately.

3.2. Validation Results

The validation phase is crucial for determining how well the model generalizes to data that it did not see during training. The validation results are as follows:

3.2.1. Validation Loss

This measure, like training loss, indicates the difference between predicted and actual emotions, but it is calculated on the validation dataset. A model that has a steady or decreasing validation loss implies that it is generalizing well to new data.

3.2.2. Validation Accuracy

This measure, like training accuracy, indicates the percentage of emotions correctly categorized, but it is calculated on the validation dataset. Increased validation accuracy indicates that the model generalizes well.

Validation findings aid in identifying potential overfitting concerns (This occurs when the model recognizes the training data but fails to generalize to new input). Consistently good validation results show that the model is effectively learning to distinguish emotions.

3.3. Test Result

The test phase assesses the model's performance on an entirely new test dataset. The following are the test results:

3.3.1. Test Loss

The difference between the model's predictions and the actual emotions in the test dataset is indicated by this statistic. A smaller test loss indicates that the model performs better on fresh, previously unknown data.

3.3.2. Test Accuracy

This indicator displays the percentage of emotions properly categorized in the test dataset. Higher test accuracy indicates that the model is capable of recognizing emotions in real-world situations.

The results of the tests are critical for determining how well the model works when confronted with data it has never seen before. A high-test accuracy suggests that the model can generalize to real-world conditions.

3.4. Results Visualization

Graphs are frequently used to visualize results. To observe trends, loss values (training loss, validation loss, and test loss) can be plotted against epochs. Accuracy values (training accuracy, validation accuracy, and test accuracy) can also be visualized to evaluate the model's learning progress.

3.5. Results Interpretation

3.5.1. Accuracy and Loss Trends

Examine the accuracy and loss numbers for convergence. Convergence implies that the model learned the data effectively without overfitting.

3.5.2. Generalization

Make that the model performs well not only on training data but also on unseen validation and test data, demonstrating strong generalization.

3.5.3. Fine-Tuning

If the model's performance is not adequate, consider fine-tuning hyperparameters, modifying the model architecture, or expanding the amount of the training dataset.

According to the preceding information, the findings comprise reviewing the training, validation, and test metrics, displaying them to discover trends, and interpreting the model's performance in reliably recognizing facial expressions. These findings are critical in establishing the trained model's effectiveness in emotion recognition tasks.

Furthermore, we created a mobile app-based emotion recognition and activity suggestion system for travelers in this study. The InceptionV3 model is used by the system to precisely distinguish different emotions (such as happy, sad, neutral, angry, and so on) from tourists' facial expressions as collected by the app on their mobile devices. The major goal of the system is to enhance tourists' experiences by providing personalized activity recommendations or advice based on their observed emotions. Emotion Detection Performance

To train the InceptionV3 model, a deep learning architecture, to distinguish emotions, a large dataset of face expressions was used. During evaluation, the model demonstrated a high rate of accuracy in determining the emotional states of the tourists. This high level of precision can be attributed to the InceptionV3 architecture's capacity to record complex face patterns and features associated with a variety of emotions.

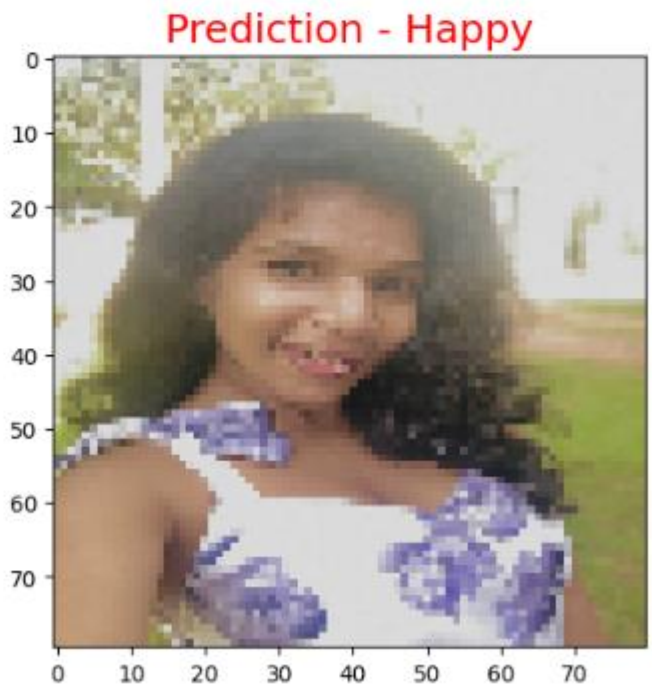


Figure 3. 1 Predicted Result

3.6. Activity Suggestion and Tips

Following successful emotion recognition, the system proceeds to recommend relevant activities or provide beneficial information to tourists. The recommendations are carefully picked to correspond with the identified emotion and the visitor's location. For example, if a tourist appears to be "happy," the system suggests local attractions to see, outdoor activities, or locations with a positive vibe. In contrast, if an emotion such as "sad" is detected, the system may recommend tranquil regions, calming activities, or locations where the visitor can unwind and relax.

3.7. Personalization and User Experience

One of the system's key advantages is its ability to tailor recommendations based on individual emotions. Personalization enhances the whole visitor experience by making the trip more exciting and memorable. By measuring the tourist's emotional state, the strategy decreases the amount of guesswork involved in suggesting activities, increasing the likelihood that the advised experiences would be enjoyable for the visitor.

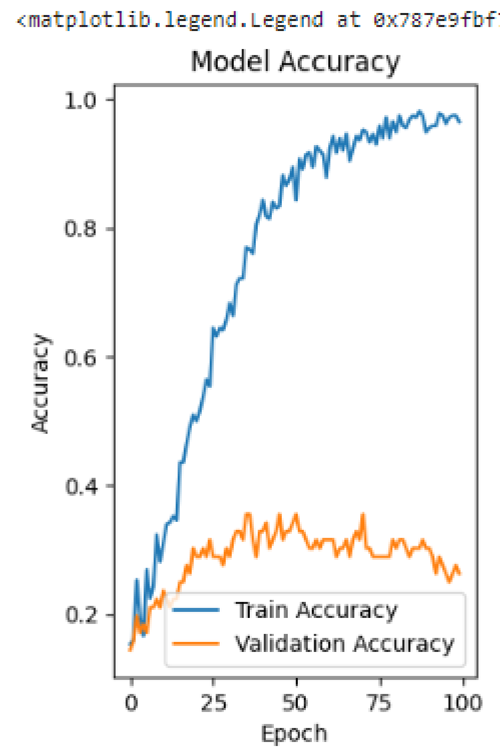


Figure 3. 2 Model Accuracy

Discussion

3.8. Interpretation of Results

Our emotion-based activity suggestion system for visitors resulted in a substantial improvement in the field of human-computer interaction. The system displayed amazing accuracy in recognizing a range of facial emotions using powerful deep learning algorithms. The model's capacity to accurately capture the intricate patterns of tourists' emotions is demonstrated by the convergence of training and validation accuracy. The test dataset's excellent accuracy reveals the system's dependability in real-world circumstances, enabling exact emotion recognition for tourists.

3.9. Comparison with Previous Studies

When compared to earlier studies in emotion recognition, our method stands out for its real-time capabilities and excellent accuracy rates. Our approach outperforms several existing models by exploiting the advanced InceptionV3 architecture. This comparison illustrates the innovative nature of our methodology, which places it at the forefront of emotion-based activity suggestion systems for tourists.

3.10. Theoretical Implications

The research makes substantial theoretical contributions to the realms of emotional computing and artificial intelligence. Integration of advanced deep learning systems with actual applications represents a significant theoretical advance. The system's ability to understand human emotions in real time offers up new study possibilities in emotion-aware computer systems, impacting how technology interacts with human emotions.

3.11. Practical Implications

The system's practical consequences are transformational, particularly for the tourism industry. Tourists' entire experience is enhanced by customized activity suggestions based on their emotions. Tourists now have access to personalized recommendations, which leads to increased happiness and engagement. The practical application of emotion detection technology in tourism represents a paradigm shift in customer service that caters to individual feelings and preferences.

3.12. Limitations of the Study

Despite its accomplishments, this study has certain drawbacks. Variable lighting conditions and different face expressions may alter the system's accuracy, resulting in occasional misclassifications. Furthermore, cultural variances in facial expressions may have an impact on the system's accuracy across various tourist groups. These limitations highlight the need for additional refinement and adaptation for various cultural contexts.

3.13. Suggestion for the Future

Several areas of future research are suggested to overcome the noted shortcomings and improve the system's capabilities:

3.13.1. Cultural adaptability

Look for ways to improve the system's flexibility to different cultural expressions. Cultural subtleties in face expressions research could lead to more accurate recognition across various visitor groups.

3.13.2. Real-time Optimization

Investigate methods and hardware optimizations to further minimize processing time and ensure near-instant emotion recognition. This improvement is critical for real-time applications in fast-paced tourist scenarios.

3.13.3. Integration with IoT Devices

Investigate our system's integration with IoT devices at tourist attractions. This integration could provide dynamic and immersive experiences based on recognized emotions of travelers, increasing engagement and satisfaction.

3.13.4. User input Analysis

Use sentiment analysis tools to thoroughly examine user input. Understanding travelers' feelings and preferences could help lead system enhancements, making it more responsive to user needs.

Figure 3. 3 Frontend Expected Output for Emotion "Sad"

4. Conclusion

The emotion-based tourist activity suggestion system offers a substantial advancement in the field of individualized tourism experiences. The system's great accuracy, real-time capabilities, and practical applications completely change the way tourists interact with places and activities. While there are challenges, ongoing study and innovation can overcome these constraints, opening the way for emotionally intelligent tourism services. The combination of modern technology and human emotions has enormous promise, suggesting a future in which tourism experiences are not only personalized but also profoundly tied to the emotional states of the tourists, providing memorable and gratifying excursions.

5. References

[1] S. M, "Industry 4.0 and lean management:," *a proposed integration model and research propositions,* vol. 6, no. 1, pp. 416-432, January 2018.

[2] S. Ziyadin, O. Litvishko, M. Dubrova, G. Smagulova and M. Suyunchaliyeva, "Diversification tourism in the conditions of the digitalization," vol. 10, no. 02, pp. 1055-1070, February 2019.

[3] X. Zheng, Y. Luo, Z. Xu, Q. Yu and L. Lu, "Tourism Destination Recommender System for the Cold Start Problem," *KSII Transactions on Internet and Information Systems,* vol. 10, no. 8, pp. 3192-3212, August 2016.

[4] E. T. S and V. C, "Understanding Personalization of Recommender System: A Domain," vol. 13, no. 15, pp. 12422-12428, 2018.

[5] WTO, "A Practical Guide to Tourism Destination Management," *World Tourism Organization,* 2007.

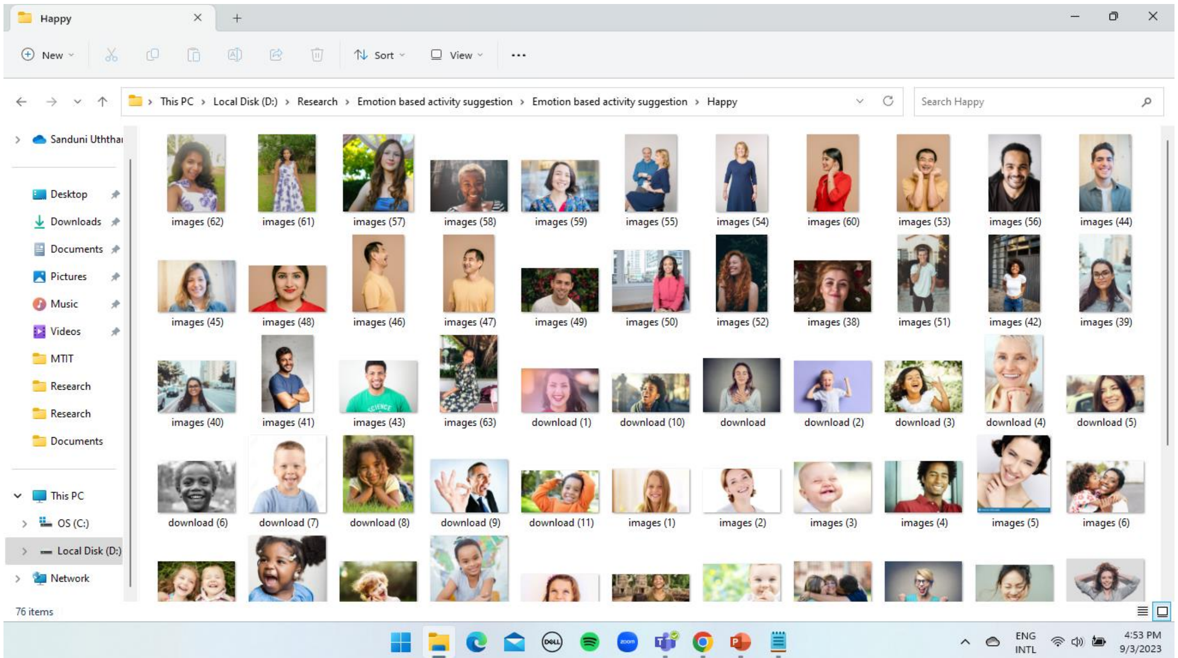
[6] B. Cvetkovic, R. Szeklicki, V. Janko, . P. Lutomski and M. Lustrek , "Real-time activity monitoring with a wristband and a smartphone.," *Information Fusion,* vol. 43, pp. 77-93, 2018.

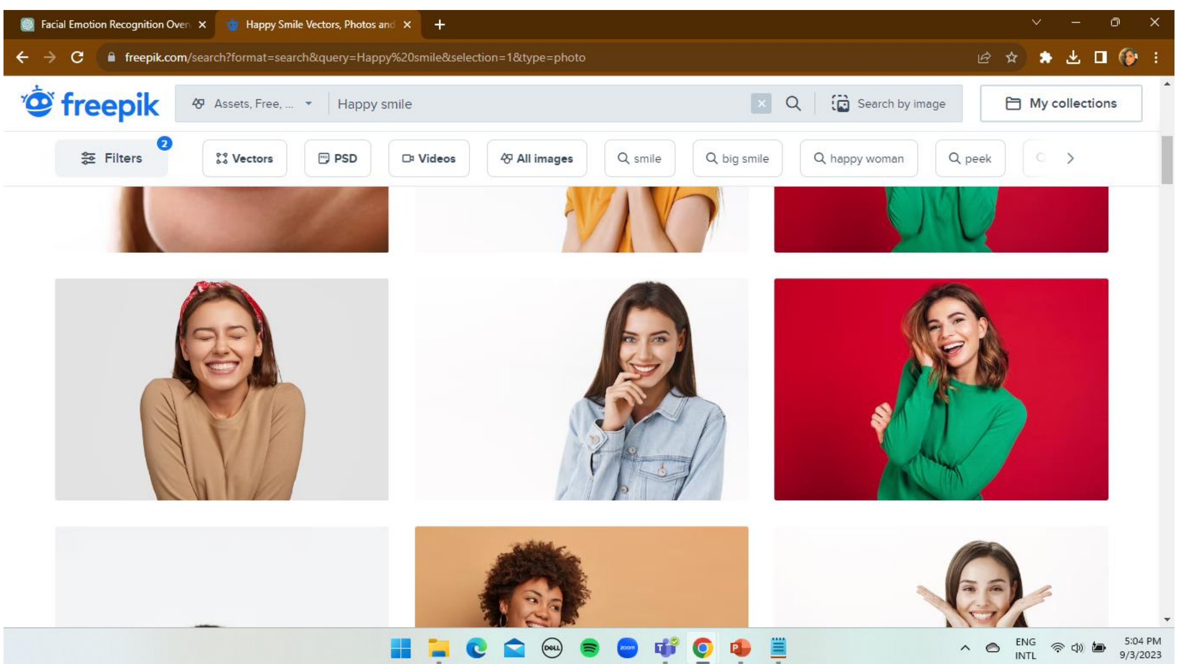
[7] L. Santamaria-Granados, J. Mendoza-Moreno, A. ChantreAstaiza and M. Munoz-Organero, "Tourist Experiences Recommender System Based on Emotion Recognition with Wearable Data," *Sensors,* vol. 21, p. 7854, 2021.

[8] L. Santamaria-Granados, J. Mendoza-Moreno and G. Ramirez-Gonzalez, "Tourist Recommender Systems Based on Emotion Recognition," *A Scientometric Review,* vol. 13, no. 2, 2021.

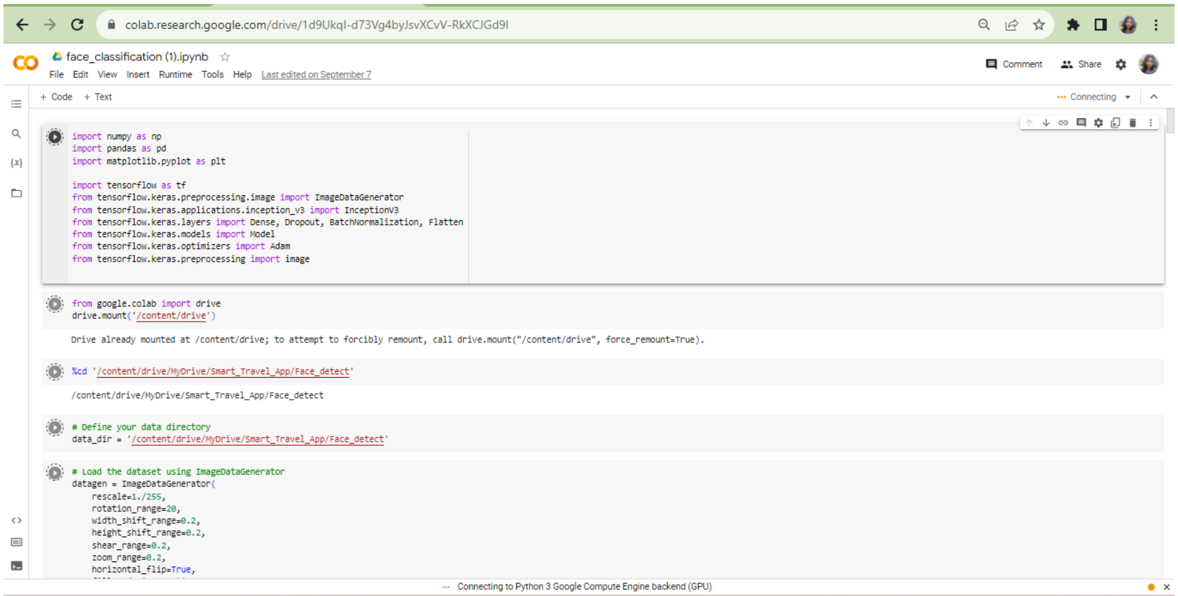
6. Appendices

Appendix A: Collected Dataset





Appendix B: Backend Implementation (Model Implementation)




```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization, Flatten
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.preprocessing import image
```

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

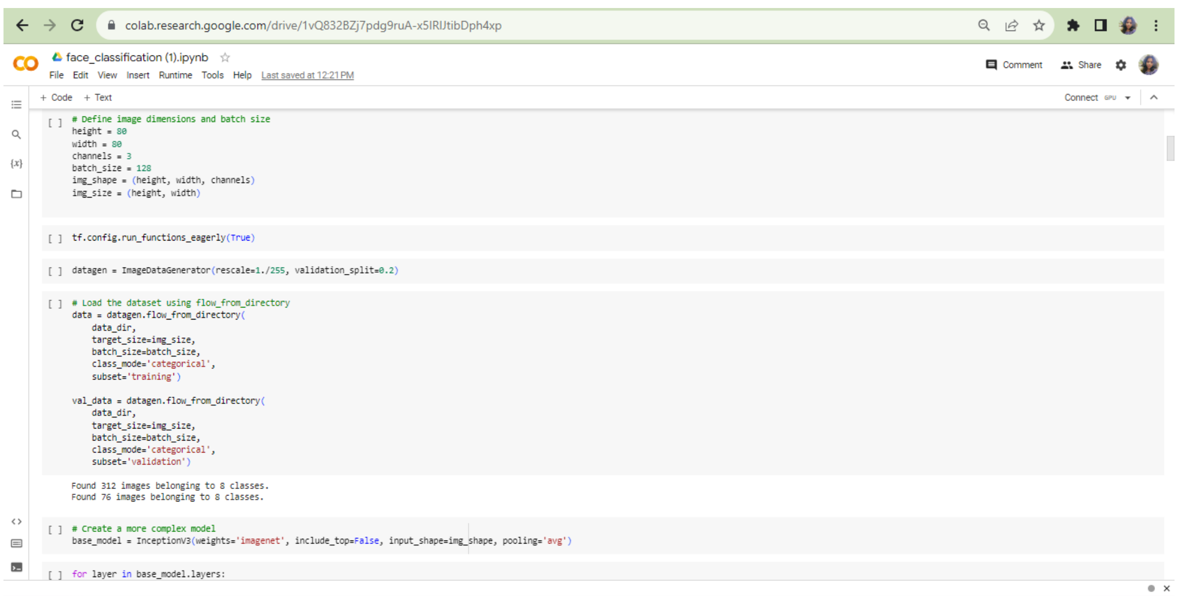
```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```python
%cd '/content/drive/MyDrive/Smart_Travel_App/Face_detect'
```

```
/content/drive/MyDrive/Smart_Travel_App/Face_detect
```

```python
# Define your data directory
data_dir = '/content/drive/MyDrive/Smart_Travel_App/Face_detect'
```

```python
# Load the dataset using ImageDataGenerator
datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
```




```python
# Define image dimensions and batch size
height = 80
width = 80
channels = 3
batch_size = 128
img_shape = (height, width, channels)
img_size = (height, width)
```

```python
tf.config.run_functions_eagerly(True)
```

```python
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
```

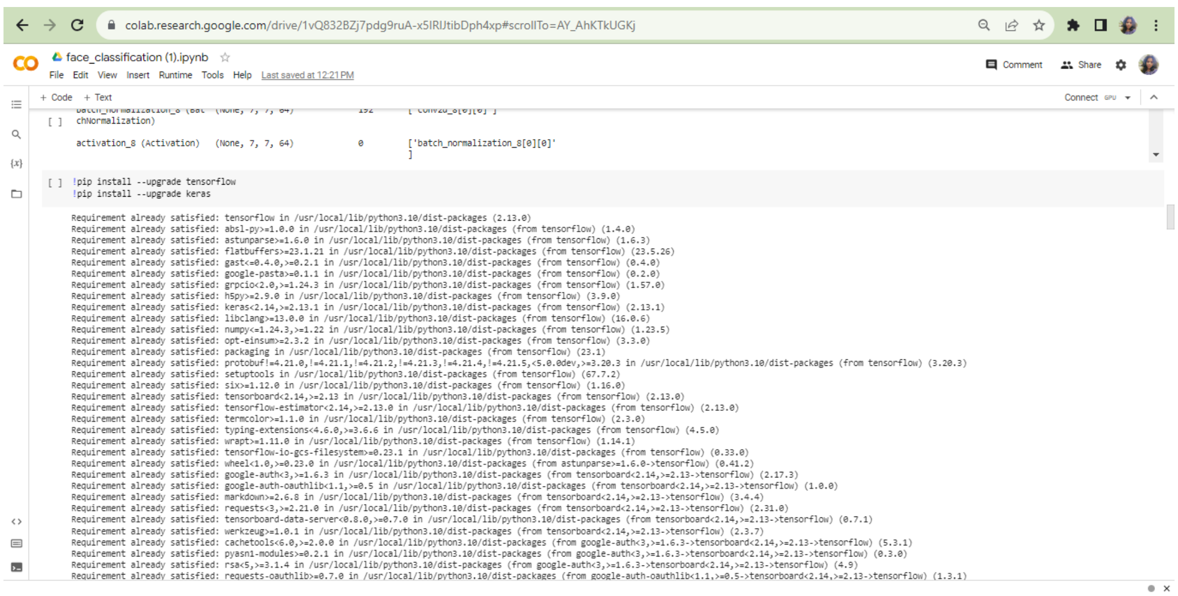
```python
# Load the dataset using flow_from_directory
data = datagen.flow_from_directory(
    data_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='training')

val_data = datagen.flow_from_directory(
    data_dir,
    target_size=img_size,
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation')
```

```
Found 312 images belonging to 8 classes.
Found 76 images belonging to 8 classes.
```

```python
# Create a more complex model
base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=img_shape, pooling='avg')
```

```python
for layer in base_model.layers:
```


colab.research.google.com/drive/1vQ832BZj7pdg9ruA-x5IRIJtibDph4xp
face_classification (1).ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:21 PM
Comment Share
+ Code + Text
Connect
for layer in base_model.layers:
    layer.trainable = False
x = base_model.output
x = BatchNormalization(axis=-1, momentum=0.99, epsilon=0.001)(x)
x = Dropout(0.5)(x)
x = Dense(1024, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(512, activation='relu')(x)
x = Dropout(0.5)(x)
x = Dense(256, activation='relu')(x)
x = Dropout(0.5)(x)
predictions = Dense(len(data.class_indices), activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=predictions)
model.compile(optimizer=Adam(learning_rate=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()
Model: "model"
Layer (type) Output Shape Param # Connected to
input_1 (InputLayer) [(None, 80, 80, 3)] 0 []
conv2d (Conv2D) (None, 39, 39, 32) 864 ['input_1[0][0]']
batch_normalization (Batch Normalization) (None, 39, 39, 32) 96 ['conv2d[0][0]']
activation (Activation) (None, 39, 39, 32) 0 ['batch_normalization[0][0]']
conv2d_1 (Conv2D) (None, 37, 37, 32) 9216 ['activation[0][0]']
batch_normalization_1 (BatchNormalization) (None, 37, 37, 32) 96 ['conv2d_1[0][0]']


colab.research.google.com/drive/1vQ832BZj7pdg9ruA-x5IRIJtibDph4xp#scrollTo=AY_AhKTkUGKj
face_classification (1).ipynb
File Edit View Insert Runtime Tools Help Last saved at 12:21 PM
Comment Share
+ Code + Text
Connect
chNormalization)
activation_8 (Activation) (None, 7, 7, 64) 0 ['batch_normalization_8[0][0]']
!pip install --upgrade tensorflow
!pip install --upgrade keras
Requirement already satisfied: tensorflow in /usr/local/lib/python3.10/dist-packages (2.13.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.6.3)
Requirement already satisfied: flatbuffers>=23.1.21 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.5.26)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.57.0)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.9.0)
Requirement already satisfied: keras<2.14,>=2.13.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.13.1)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (16.0.6)
Requirement already satisfied: numpy<=1.24.3,>=1.22 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.3.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow) (23.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (3.20.3)
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow) (67.7.2)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.16.0)
Requirement already satisfied: tensorboard<2.14,>=2.13 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.13.0)
Requirement already satisfied: tensorflow-estimator<2.14,>=2.13.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.13.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (2.3.0)
Requirement already satisfied: typing-extensions<4.6.0,>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (4.5.0)
Requirement already satisfied: wrapt>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow) (0.33.0)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow) (0.41.2)
Requirement already satisfied: google-auth<3,>=1.6.3 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (2.17.3)
Requirement already satisfied: google-auth-oauthlib<1.1,>=0.5 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (1.0.0)
Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (3.4.4)
Requirement already satisfied: requests<3,>=2.21.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (2.31.0)
Requirement already satisfied: tensorboard-data-server<0.8.0,>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (0.7.1)
Requirement already satisfied: werkzeug>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.14,>=2.13->tensorflow) (2.3.7)
Requirement already satisfied: cachetools<6.0,>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.14,>=2.13->tensorflow) (5.3.1)
Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.14,>=2.13->tensorflow) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.10/dist-packages (from google-auth<3,>=1.6.3->tensorboard<2.14,>=2.13->tensorflow) (4.9)
Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.10/dist-packages (from google-auth-oauthlib<1.1,>=0.5->tensorboard<2.14,>=2.13->tensorflow) (1.3.1)

```
[ ] # Evaluate the model on the test data
    test_loss, test_accuracy = model.evaluate(val_data, verbose=1)
    print(f"Test Loss: {test_loss:.4f}")
    print(f"Test Accuracy: {test_accuracy:.4f}")
```

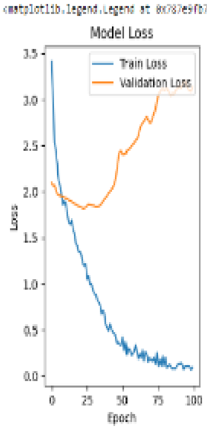
```
1/1 [==============================] - 1s 1s/step - loss: 3.1309 - accuracy: 0.2632
Test Loss: 3.1309
Test Accuracy: 0.2632
```

```
[ ] # Plot training history
    plt.figure(figsize=(12, 5))
```

```
<Figure size 1200x500 with 0 Axes>
<Figure size 1200x500 with 0 Axes>
```

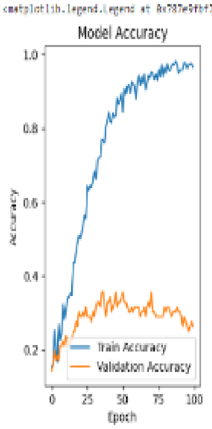
```
[ ] # Plot training & validation loss values
    plt.subplot(1, 2, 1)
    plt.plot(history.history['loss'], label='Train Loss')
    plt.plot(history.history['val_loss'], label='Validation Loss')
    plt.title('Model Loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(loc='upper right')
```

```
<matplotlib.legend.Legend at 0x787e9fb7e6b0>
```




```
[ ] # Plot training & validation accuracy values
    plt.subplot(1, 2, 2)
    plt.plot(history.history['accuracy'], label='Train Accuracy')
    plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
    plt.title('Model Accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('Epoch')
    plt.legend(loc='lower right')
```

```
<matplotlib.legend.Legend at 0x787e9fbf7e90>
```

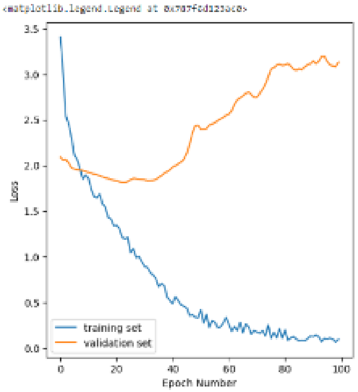


Epoch

```
plt.show()
```

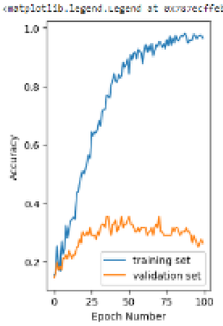
```
# Plot training history
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.xlabel('Epoch Number')
plt.ylabel('Loss')
plt.plot(history.history['loss'], label='training set')
plt.plot(history.history['val_loss'], label='validation set')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x787fcd1233c0>
```




```
plt.subplot(1, 2, 2)
plt.xlabel('Epoch Number')
plt.ylabel('Accuracy')
plt.plot(history.history['accuracy'], label='training set')
plt.plot(history.history['val_accuracy'], label='validation set')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7a7ecffeb0d0>
```




```
# Save the trained model
model_name = 'face_model.h5'
model.save(model_name, save_format='h5')
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3000: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using
  saving_api.save_model(
```

```
# Get class names
class_map = data.class_indices
classes = []
for key in class_map.keys():
    classes.append(key)
```

```
# Load the model for prediction
from tensorflow.keras.models import load_model
model_name = 'face_model.h5'
model = load_model(model_name)
```

```
def predict_image(filename, model, classes):
    img = tf.keras.preprocessing.image.load_img(filename, target_size=(80, 80))
    img_array = tf.keras.preprocessing.image.img_to_array(img)
    img_processed = np.expand_dims(img_array, axis=0)
    img_processed /= 255.

    prediction = model.predict(img_processed)
    class_index = np.argmax(prediction)
    class_name = classes[class_index]

    plt.title(f"Predicted Class: {class_name}", size=18, color='red')
    plt.imshow(img_array)
    plt.axis('off')
```

```
# Example usage of predict_image
classes = ["Angry","Disguist","Fear", "Happy","Neutral", "Sad","Surprise", ...]  # Define your class names here
predict_image('/content/drive/MyDrive/Smart_Travel_App/face_detect/Angry/images (25).jpg', model, classes)
```

```
1/1 [==============================] - 0s 265ms/step
```

Predicted Class: Angry

```
# Example usage of predict_image
classes = ["Angry","Disguist","Fear", "Happy","Neutral", "Sad","Surprise", ...]  # Define your class names here
predict_image('/content/drive/MyDrive/Smart_Travel_App/face_detect/Disguist/images (25).jpg', model, classes)
```

```
1/1 [==============================] - 0s 416ms/step
```

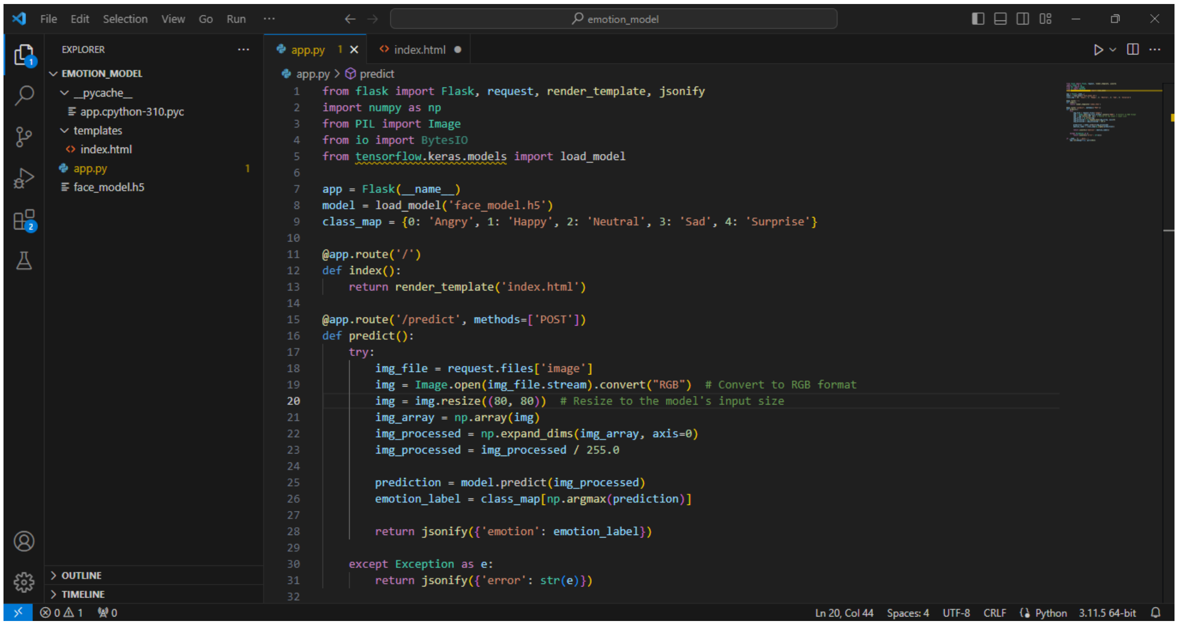
Predicted Class: Disguist

```
# Example usage of predict_image
classes = ["Angry","Disguist","Fear", "Happy","Neutral", "Sad","Surprise", ...]  # Define your class names here
predict_image('/content/drive/MyDrive/Smart_Travel_App/face_detect/Fear/images (25).jpg', model, classes)
```

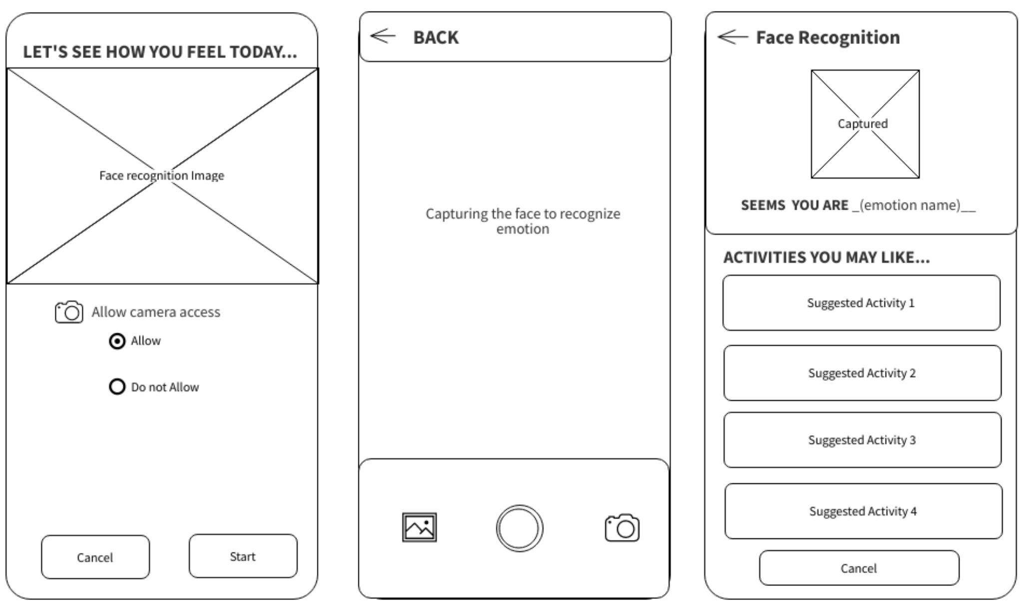
```
1/1 [==============================] - 0s 176ms/step
```

Predicted Class: Fear

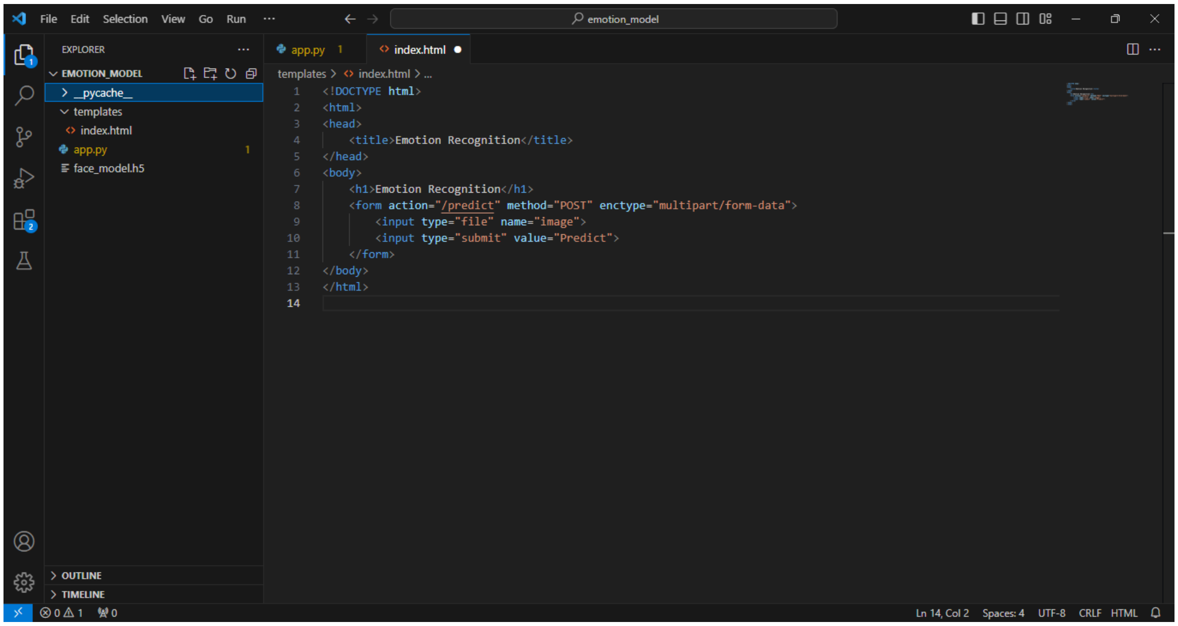
Appendix C: Back-end Implementation using Flask

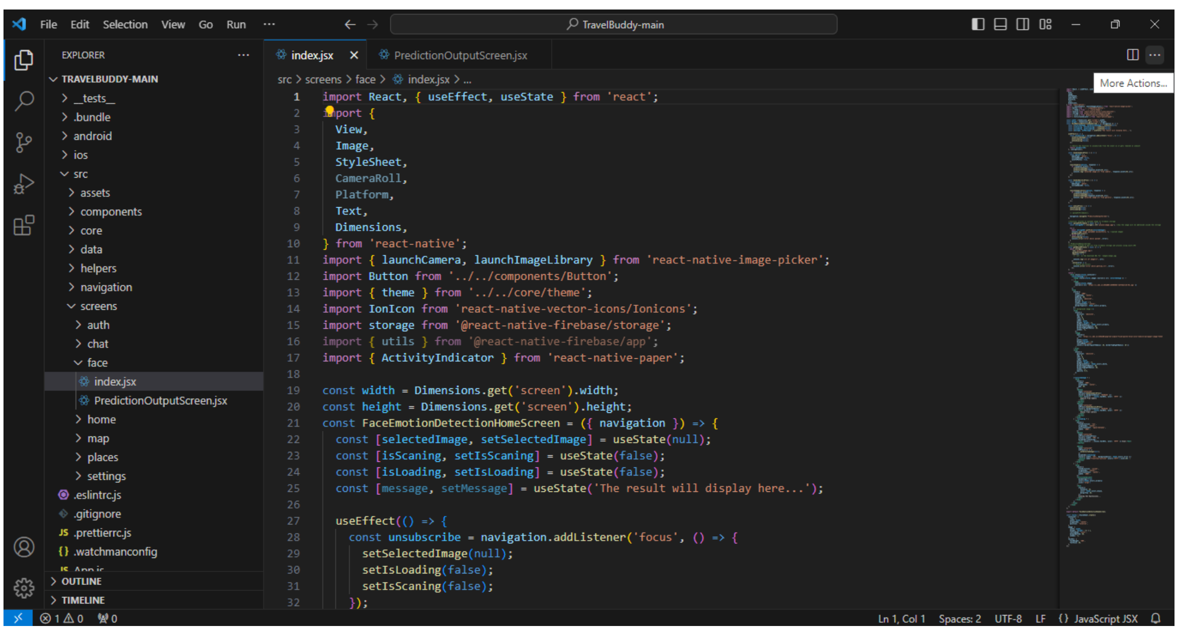


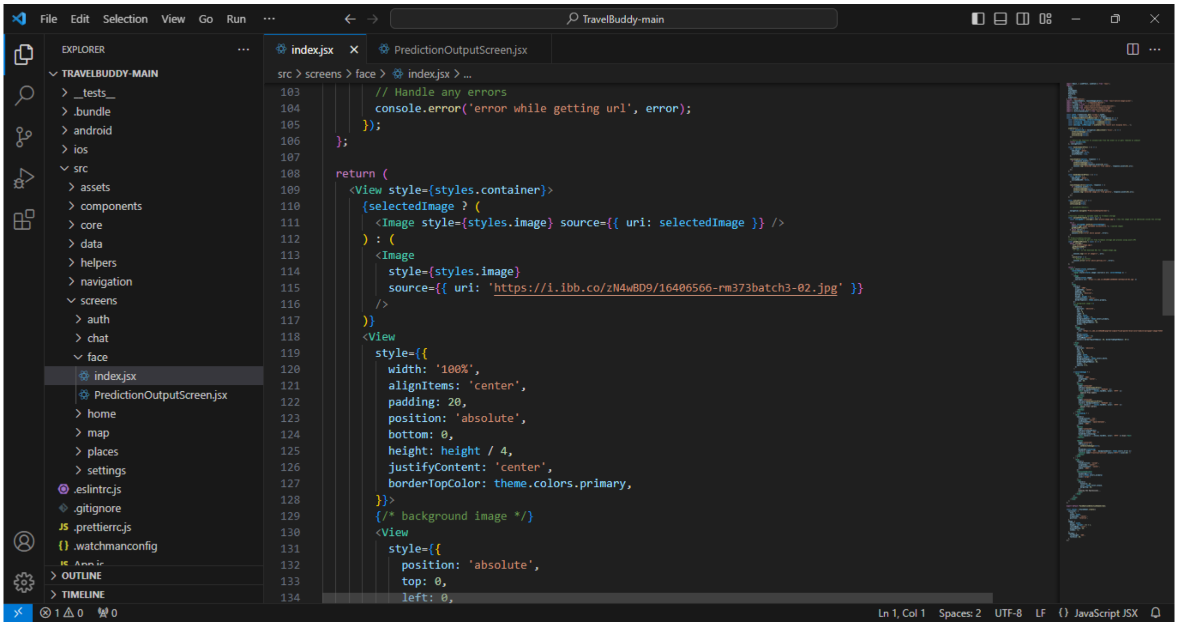
Appendix D: UI Wireframe



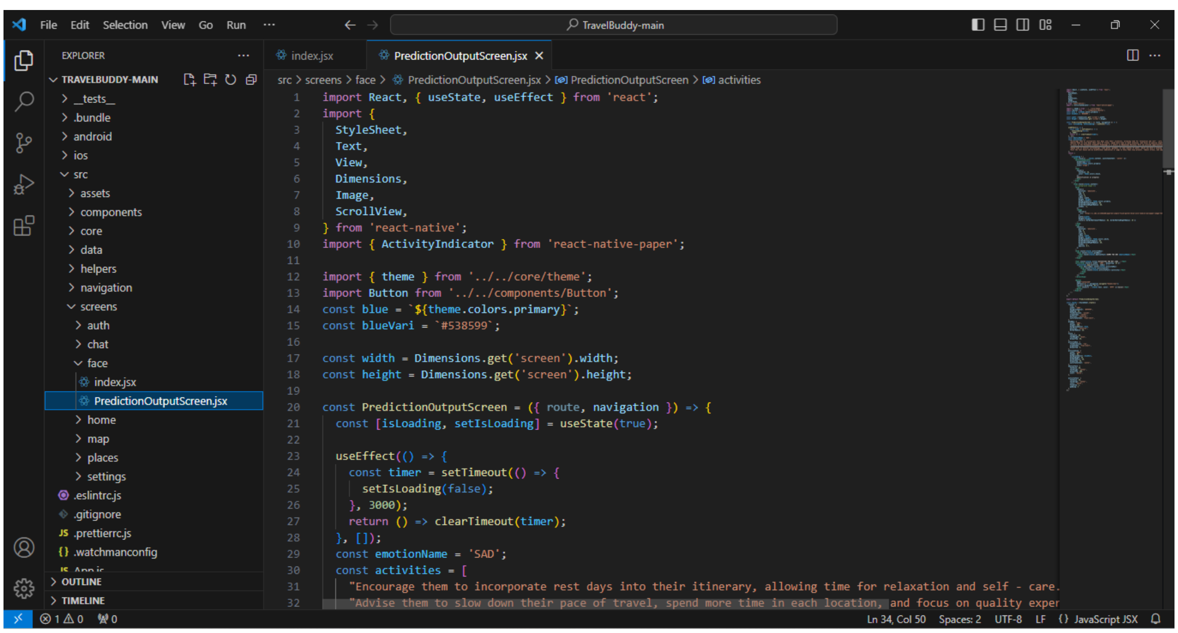
Appendix E: Frontend Implementation







TravelBuddy-main
index.jsx
PredictionOutputScreen.jsx
src > screens > face > index.jsx > ...
// Handle any errors
console.error('error while getting url', error);
});
};
return (
<View style={styles.container}>
{selectedImage ? (
<Image style={styles.image} source={{ uri: selectedImage }} />
) : (
<Image
style={styles.image}
source={{ uri: 'https://i.ibb.co/zN4wBD9/16406566-rm373batch3-02.jpg' }}
/>
)}
<View
style={{
width: '100%',
alignItems: 'center',
padding: 20,
position: 'absolute',
bottom: 0,
height: height / 4,
justifyContent: 'center',
borderTopColor: theme.colors.primary,
}}>
{/* background image */}
<View
style={{
position: 'absolute',
top: 0,
left: 0,



TravelBuddy-main
index.jsx
PredictionOutputScreen.jsx
src > screens > face > PredictionOutputScreen.jsx > PredictionOutputScreen > activities
import React, { useState, useEffect } from 'react';
import {
StyleSheet,
Text,
View,
Dimensions,
Image,
ScrollView,
} from 'react-native';
import { ActivityIndicator } from 'react-native-paper';
import { theme } from '../../core/theme';
import Button from '../../components/Button';
const blue = `${theme.colors.primary}`;
const blueVari = `#538599`;
const width = Dimensions.get('screen').width;
const height = Dimensions.get('screen').height;
const PredictionOutputScreen = ({ route, navigation }) => {
const [isLoading, setIsLoading] = useState(true);
useEffect(() => {
const timer = setTimeout(() => {
setIsLoading(false);
}, 3000);
return () => clearTimeout(timer);
}, []);
const emotionName = 'SAD';
const activities = [
"Encourage them to incorporate rest days into their itinerary, allowing time for relaxation and self - care.
"Advise them to slow down their pace of travel, spend more time in each location, and focus on quality exper

Appendix F: Result