Visvesvaraya Technological University, Belagavi – 590010

PROJECT REPORT
ON

Pixort: An Application for Photo Album Clustering

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

Cheryl Lina Mathias	4SO16CS030
Crystal Fay D'Souza	4SO16CS031
Job Alexander	4SO16CS044
Mariah Sneha Hudson	4SO16CS058

Under the guidance of
Ms Renuka Tantry
Assistant Professor, CS&E Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ST JOSEPH ENGINEERING COLLEGE
(Affiliated to VTU-Belagavi, Recognized by AICTE, NBA Accredited)

Vamanjoor, Mangaluru-575028, Karnataka

2019-2020

Visvesvaraya Technological University, Belagavi – 590010

PROJECT REPORT
ON

Pixort: An Application for Photo Album Clustering

Submitted in partial fulfillment of the requirements for the degree

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted by

Cheryl Lina Mathias	4SO16CS030
Crystal Fay D'Souza	4SO16CS031
Job Alexander	4SO16CS044
Mariah Sneha Hudson	4SO16CS058

Under the guidance of
Ms Renuka Tantry
Assistant Professor, CS&E Department

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
(Affiliated to VTU-Belagavi, Recognized by AICTE, NBA Accredited)

ST JOSEPH ENGINEERING COLLEGE
Vamanjoor, Mangaluru-575028, Karnataka

2019-2020

ST JOSEPH ENGINEERING COLLEGE

(Affiliated to VTU-Belagavi, Recognized by AICTE, B. E. (CSE) - NBA Accredited)

Vamanjoor, Mangaluru-575028

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled **"Pixort: An Application for Photo Album Clustering"** *carried out by*

Cheryl Lina Mathias	**4SO16CS030**
Crystal Fay D'Souza	**4SO16CS031**
Job Alexander	**4SO16CS044**
Mariah Sneha Hudson	**4SO16CS058**

bonafide students of VIII semester (Computer Science & Engineering) in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belagavi during the year 2019-2020. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Ms Renuka Tantry	**Dr Sridevi Saralaya**	**Dr Rio D'Souza**
Project Guide	**HOD - CSE**	**Principal**

External Viva

Name of the Examiner **Signature with date**

1

2

DECLARATION

I **Cheryl Lina Mathias** bearing USN **4SO16CS030,** student of 8th semester B.E. in Computer Science and Engineering, St Joseph Engineering College, Mangaluru hereby declare that project work entitled **"Pixort: An Application for Photo Album Clustering"** has been duly executed by me under the guidance of **Ms Renuka Tantry**, Assistant Professor, Department of Computer Science and Engineering, St Joseph Engineering College, Mangaluru and submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Engineering in Computer Science and Engineering by Visvesvaraya Technological University, Belagavi for the year 2019-20.**

Date : 03-06-2020
Name : Cheryl Lina Mathias
USN : 4SO16CS030
Place : Mangaluru

DECLARATION

I **Crystal Fay D'Souza**, bearing USN **4SO16CS031**, student of 8th semester B.E. in Computer Science and Engineering, St Joseph Engineering College, Mangaluru hereby declare that project work entitled **"Pixort: An Application for Photo Album Clustering"** has been duly executed by me under the guidance of **Ms Renuka Tantry**, Assistant Professor, Department of Computer Science and Engineering, St Joseph Engineering College, Mangaluru and submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Engineering in Computer Science and Engineering by Visvesvaraya Technological University, Belagavi for the year 2019-20.**

Date : 03-06-2020
Name : Crystal Fay D'Souza
USN : 4SO16CS031
Place : Mangaluru

DECLARATION

I **Job Alexander**, bearing USN **4SO16CS044**, student of 8th semester B.E. in Computer Science and Engineering, St Joseph Engineering College, Mangaluru hereby declare that project work entitled **"Pixort: An Application for Photo Album Clustering"** has been duly executed by me under the guidance of **Ms Renuka Tantry**, Assistant Professor, Department of Computer Science and Engineering, St Joseph Engineering College, Mangaluru and submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Engineering in Computer Science and Engineering by Visvesvaraya Technological University, Belagavi for the year 2019-20.**

Date : 03-06-2020
Name : Job Alexander
USN : 4SO16CS044
Place : Mangaluru

DECLARATION

I **Mariah Sneha Hudson,** bearing USN **4SO16CS058**, student of 8th semester B.E. in Computer Science and Engineering, St Joseph Engineering College, Mangaluru hereby declare that project work entitled **"Pixort: An Application for Photo Album Clustering"** has been duly executed by me under the guidance of **Ms Renuka Tantry**, Assistant Professor, Department of Computer Science and Engineering, St Joseph Engineering College, Mangaluru and submitted in partial fulfillment of the requirements for the award of degree of **Bachelor of Engineering in Computer Science and Engineering by Visvesvaraya Technological University, Belagavi for the year 2019-20.**

Date : 03-06-2020
Name : Mariah Sneha Hudson
USN : 4SO16CS058
Place : Mangaluru

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible, whose constant guidance and encouragement crowned our efforts with success. We take this opportunity to thank those who have helped and motivated us throughout the completion of this project.

We would like to express our deep and sincere gratitude to our project guide, **Ms Renuka Tantry**, Assistant Professor, Department of Computer Science and Engineering, for her constant guidance and support, without which this project wouldn't have been completed successfully.

We would like to express our heartfelt thanks to our project coordinator, **Ms Lavina Jean D'Silva,** Assistant Professor, Department of Computer Science and Engineering, who has always extended her wholehearted support, guidance and assistance. We owe our great debt to **Dr Sridevi Saralaya,** Head of the Department of Computer Science and Engineering, for her support and encouragement during the course of development of this project.

We are immensely grateful to our Director, **Rev. Fr Wilfred Prakash D'Souza**, Assistant Directors, **Rev. Fr Rohit D'Costa** and **Rev. Fr Alwyn Richard D'Souza** and our Principal, **Dr Rio D'Souza** for their meticulous support and encouragement.

We extend our gratitude to all the faculty and the staff of the Department of Computer Science and Engineering, SJEC, for their advice, kind co-operation and assistance throughout the academic year.

We also thank all the laboratory administrators and assistants who have helped us make this project a successful one. Lastly, we would like to express our heartfelt appreciation towards our classmates and seniors for their guidance and suggestions.

ABSTRACT

Retrieving digital records on the internet and capturing, uploading images has become such a notable part of our daily existence. One may often encounter an instance when we only have an image as reference and want to find the link to a web page consisting of similar images from the internet, which is not possible due to the lack of search techniques based on content. Retrieval of images based on perusal of their visual content is therefore a compelling and gravitating topic of research posing great challenges.

An effective image clustering application should be capable of accurately extracting images and providing functionalities like Facial Detection and Recognition for customized usage and retrieval. Numerous attempts have been made in the past to implement Emotion Detection that have largely relied on textual Sentiment Analysis but no application has integrated both CBIR and image-based Emotion Detection with decent accuracy. Pixort, implements CBIR and Facial Detection, Recognition, Selection and allows the user to retrieve and segregate the pictures based on facial emotions namely – happy and sad.

The application has been designed using the computer vision tool box offered by MATLAB. Feature Extraction is carried out on the image which is then subjected to Local Binary Pattern algorithm to extract texture related features. These features are then given to a neural network for classification. Facial Detection is implemented using the Viola-Jones algorithm followed by Facial Recognition and Emotion Detection that has been achieved by establishing a relationship on the basis of vivid features using correlation.

Pixort showcases an overall accuracy of about 81 per cent for CBIR and is able to detect faces and segregate images based on emotion with a precision of 0.8675 and recall of 0.972. Using MATLAB for development is extremely advantageous because of its inherent toolboxes and built in functions. Higher levels of accuracy can be visualized if the network is trained more extensively with enhanced filters for advanced feature extraction.

Pixort is currently capable of recognizing nine faces and segregating images based on two emotions. There is definitely scope for improvement with further development and augmentation. Pixort or similar applications can be replicated for videos in addition to images and can uncover novel and contemporary techniques for search and retrieval.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PROBLEM STATEMENT

Digitization of the photography industry has resulted in a massive growth in the amount of visual information available. In the current scenario, images are retrieved by a manual search exhausting valuable time and resources. Thus, the need to store and retrieve images in an efficient manner has led to the demand for Content Based Image Retrieval (CBIR) systems that are capable of extracting images containing desired content. However, these systems do not take into account high level features like human emotions identified within the images. To overcome this, Facial Detection, Facial Recognition and Emotion Detection must be performed to identify images that contain a specific person exhibiting a specific emotion.

OBJECTIVES

To develop an application that:

- The user can use to upload a query image
- Performs feature extraction on the query image and retrieves similar images using content based image retrieval
- Performs facial detection and detects all the faces in the image
- The user can use to select a face among the detected faces
- Performs facial recognition to identify the selected face
- Retrieves images of the person based on query image and segregates it based on emotion

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

Digitization of the photography industry has led to an incredible ease in the acquisition of a camera, either as a standalone device or incorporated into mobile phones which has led to the generation of a huge collection of images that are stored in a digital format. Automatic image retrieval has become an important research problem considering its usage in handling the gigabytes of unlabeled image data that are generated and stored digitally in huge repositories, containing visual information both on the web as well as in the network computing system. Information storage and image acquisition technologies have also progressed to facilitate the creation of large image datasets. Due to the massive growth in the amount of visual information available, information systems have to be upgraded to be able to manage this data.

Manual browsing has been found to be cumbersome even with private collections. With the growth in computational power and the decrease in the cost of storage media, the access and retrieval of the information embedded in these collections using techniques that can present the information efficiently and accurately is imperative.

The main objective of this project is to build an image retrieval application to analyze the query image and retrieve images that are relevant to it.

1.1 Problem Definition

The usage of digital devices has been rapidly increasing, due to various advancements in the field of technology. Thus, the quantity of multimedia data stored is also increasing making the retrieval of an image from an archive a very challenging problem, widely known as the image retrieval problem. The fundamental requirement of an image retrieval model is to identify and organize the images that are in a visual semantic relationship with the query image provided by the user. Most of the search engines on the Internet retrieve

the images on the basis of text that require captions as input. Manually labeling the images in existing image archives containing millions of images is nearly impossible.

Content Based Image Retrieval is the solution to the above mentioned problems. The main requirement is to provide a query image whose visual contents will be analyzed and matched with the images in the archive in terms of an image feature vector.

In private collections of images, identities of individuals appearing in the images is the most important aspect of photo browsing. Users may search their collection for all their images from a specific event. Typically, they search for images wherein they are smiling. Content Based Image Retrieval is unable to identify high level features like human faces and emotions within the images.

To overcome these issues, the application uses Content Based Image Retrieval to retrieve images containing inanimate objects and scenarios. It will perform Facial Detection and Recognition followed by Emotion Detection to identify images of a person exhibiting a specific emotion. This methodology simplifies the search for images in a large database more effectively than text based approaches.

1.2 Scope and Importance

Content Based Image Retrieval is a dominating area of research due to the vast possibilities in both academic and industry ranging from medical and forensic applications to security applications that can be implemented on cloud system, mobile phones and even the World Wide Web. Browsing through a small collection of images to identify an image may be feasible but this technique is not practical for large databases. Thus, Content Based Image Retrieval solves the problem of locating an image in a large and varied collection. It can assist users to search and organize private photo collections. It can improve existing search engines by providing the ability to retrieve data using both text and image features. The fundamental idea can also be extended to the World Wide Web where objectionable web content can be identified, filtered and blocked.

Facial Detection, Recognition and Emotion Detection is gaining attention due to the significant impact it can make in various domains such as gaming, health environments and

education. It can be used to determine the emotional state of a hospital patient by analyzing their facial expression, physiological signals and behavior. Emotions of the students in a classroom can be determined to evaluate the quality of the class. Security agencies can identify people from live feed via cameras using any pre-existing images available in the database. Also, classification of visual information based on emotions can aid the automatic tagging of images with emotional categories and categorizing video sequences into genres.

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

CHAPTER 2

SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Introduction

A software requirements specification is a document that provides a detailed description of the software system being developed. It states the functional and non-functional requirements of the system and includes a set of use cases to describe ideal interactions with the user. It forms the basis of an agreement between clients and corporations on how the software product should perform and what must be done in case of a malfunction. When used appropriately, software requirements specification can help prevent software failure.

2.1.1 Purpose

The purpose of this project is to build a photo album clustering application that will retrieve images featuring an object or person(s) from a collection of images.

2.1.2 Document Conventions

CBIR Content Based Image Retrieval
GUI Graphical User Interface
ANN Artificial Neural Network
LBP Local Binary Pattern

2.1.3 Product Scope

The photo album clustering application is a tool used to locate an image in a large image dataset. It identifies an image by analyzing its content, thus providing a greater accuracy when compared to text based retrieval. In private collections, images featuring a specific person can be retrieved, eliminating the need for manual search. The content based approach for retrieval could be extended to the World Wide Web, where the Internet can be browsed using text as well as features detected from an image. This application could

specifically benefit professional photographers who work with images on a daily basis.

2.2 Overall Description

This section of the document describes the features and functions of the application. It specifies the assumptions and constraints under which the application is to be implemented. It also identifies the class of users that are most likely to benefit from the application.

2.2.1 Product Perspective

In the last few years, the complexity and quantity of multimedia content, especially images, has grown exponentially. Millions of images are clicked and uploaded on a daily basis and stored in one's device. Classification and analysis of these images has become a major challenge. Pixort has been initiated under the pretext of simplifying this issue.

In addition to CBIR, the application will also be capable of analyzing sentiments and fetching images based on a query image which will be an auxiliary in addition to the services provided by the online image management applications.

2.2.2 Product Functions

The major functions of the application are depicted in figure 2.1.

- **GUI Interaction:** Allows a user to upload an image (query image) to the application for processing.
- **Storage and Access Methods:** Access the file system to acquire the query image and to save the resultant images after retrieval.
- **CBIR using Query Image:** Analyze the content of the query image and retrieve relevant images from a collection of images.
- **Facial Detection:** Detect the faces in the image.
- **Face Selection:** Allow the user to select the face whose images have to be retrieved.
- **Facial Recognition and Emotion detection:** Recognize the detected face, retrieve images featuring the face and classify the images depending on the emotion featured

by the person in the images.

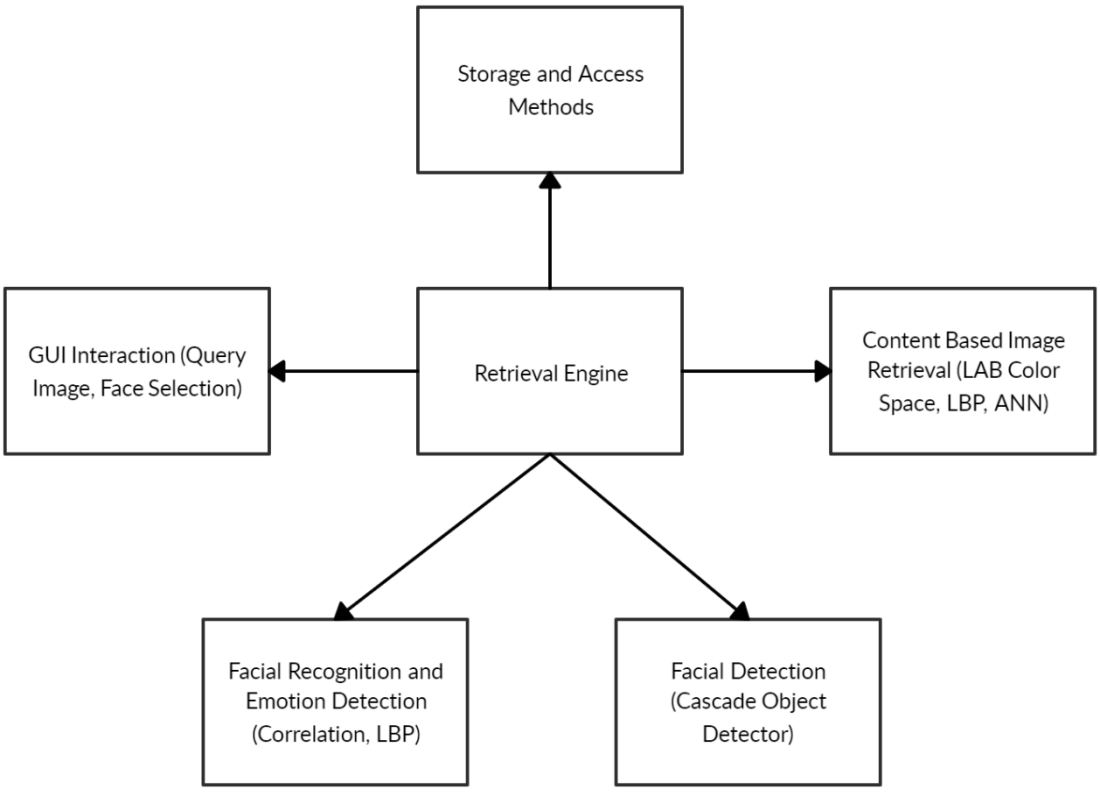

Figure 2.1 Major Functions of Pixort

2.2.3 User Classes and Characteristics

The application is expected to be used by the following kinds of users:

- **Typical Users:** People/photographers who may want to retrieve selected images from an album.
- **Advanced/Professional Users:** Engineers or researchers, who want to use the application for emotion detection and its application in various other branches. They might want to study the accuracy and research the methodology used.
- **Developers/Programmers:** They will work on further development of the application, fix errors and keep training the model with varied training data to increase accuracy.

Table 2.1 Characteristics of User Classes

	TYPICAL USERS	PROFESSIONAL USERS	PROGRAMMERS
Frequency of use	High	Low	Medium
Technical Expertise	Low	High	High
Privilege level	Low	Medium	High
Experience	Low	High	Medium to High

2.2.4 Operating Environment

Requirements for the environment in which the application is to be used:

- **Device:** Laptop/Personal Computer
- **Installed RAM:** 4GB or above
- **Operating System:** Windows 10

2.2.5 Design and Implementation Constraints

The application is bound by the following constraints:

- Memory capacity limits the size of the image folder and number of images stored in it. This leads to reduction in the amount of training data which may impact accuracy of the application.
- Neural Networks require excessively long training time which can make the process of training the model laborious and slow.
- Noisy/poorly tagged training data can limit accuracy or lead to misclassification.
- The developer is also bound by the processing power of the device and constrained to use MATLAB for image processing due to lack of significant choice.

2.2.6 Assumptions and Dependencies

The application is built under the following assumptions:

- The query image uploaded will be clear and noise free.
- The query image must be of the format .png or .jpeg.
- Device will have minimum space of 10GB to store sufficient images for training.

2.3 External Interface Requirements

This section specifies the user, hardware and software interface requirements. It provides the information required to ensure compatibility between the application and the computer system.

2.3.1 User Interfaces

- MATLAB 8.4 - Guide

2.3.2 Hardware Interfaces

- **Device:** Laptop/Personal Computer
- **Processor:** Intel® Core™ i3-7020U CPU @ 2.30GHz 2.30GHz
- **Installed RAM:** 4GB (Minimum), 8GB (Recommended)
- **Disk:** 10GB for MATLAB, 1GB per 1000 images

2.3.3 Software Interfaces

- **Operating System:** Windows 10
- **IDE:** MATLAB 8.4
- **Programming Language:** MATLAB

2.4 System Features

2.4.1 Content Based Image Retrieval

- **Description**

  This feature enables the user to retrieve similar images of objects from the image folder. The user must provide a query image that contains the object and the gallery folder from which images have to be retrieved. The application will then retrieve all the images that feature the object using CBIR and save them into a folder in the file system. This simplifies the process of searching for related images across the image folder and easily fetches all the linked images with a click of a button.

- **Functional Requirements**

  1. The user must provide a query image and gallery folder.
  2. Based on the query image, feature extraction should be carried out on the query image.
  3. Distance and similarity measures have to be calculated on the extracted features to fetch the required images.
  4. The images must be saved in the file system.

2.4.2 Facial Detection, Facial Recognition and Emotion Detection

- **Description**

  This feature allows the user to retrieve all the images featuring a specific person. The application performs facial detection followed by recognition to identify the person in the image. It will then retrieve all the images featuring the person and classifies the images into two emotions namely - happy and sad. These images will then be saved into separate folders in the file system.

- **Functional Requirements**

  1. The user must provide a query image and gallery folder.
  2. Facial detection is carried out and the detected faces will be highlighted.

3. The user must select the face whose images have to be retrieved.
4. Facial recognition will be carried out and all images featuring the person will be retrieved.
5. The images will be classified into ‘Happy’ and ‘Sad’.
6. The images will be saved in the file system.

2.5 Other Non-Functional Requirements

Non-functional requirements are used to specify criteria that evaluate the operation of the system. They define the characteristics of the system as a whole. This section describes the non-functional requirements of the application.

2.5.1 Performance Requirements

- **Standard Compliance**
  The graphical user interface should have a consistent look and feel. The application must be interactive and the delays involved must be minimum. In case of opening windows, popping error messages and saving the images, the delay must be less than two seconds.

- **Availability**
  The services of the application shall be available at all times as long as it is in proper working condition.

- **Reliability**
  The application should be reliable in all its services. It should be optimized enough to provide error-free services.

2.5.2 Safety Requirements

Bugs in the application should raise an alert so that they can be fixed. The application should not terminate without a warning.

2.5.3 Security Requirements

The application shall not store the images or the metadata associated with the images after it has completed its operation. It shall not share the user data with any third party services or applications. Any type of user can use it without any additional privileges. However, only authorized personnel shall be allowed to access and modify the code for debugging or enhancement.

CHAPTER 3
DESIGN

CHAPTER 3

DESIGN

Software design is a stage in the software engineering process at which an executable software system is developed. It is an activity involving identification of software components based on the needs of the client. It deals with representing the client's requirement, as stated in the software requirements specification document, into a format that can be easily implemented using a programming language. The software design phase is the first step in the Software Design Life Cycle (SDLC), which shifts the focus from the problem to the solution.

3.1 Abstract Design

3.1.1 Architectural Diagram

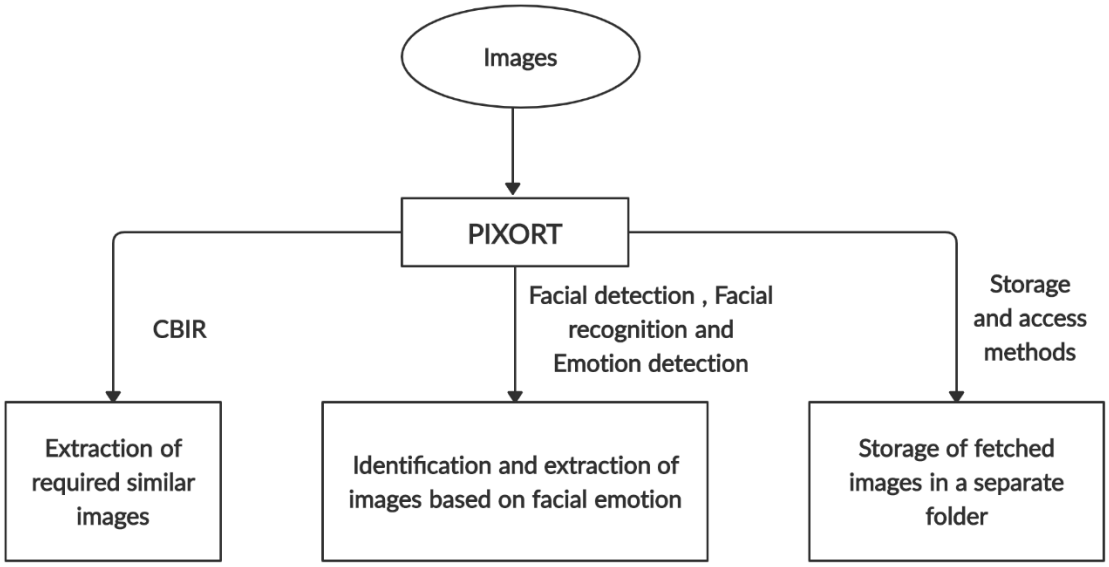

Figure 3.1 Architectural Diagram

Architectural diagrams provide a high-level visualization of the overall structure of a system to ensure that its functionalities meet the requirements of the stakeholders. The architectural diagram of Pixort is shown in figure 3.1.

The application consists of three basic functionalities:

1. **Content Based Image Retrieval**
   Here, the user will pass a query image and feature extraction will be carried out on that image. A similar process will be carried out on all the images in the gallery and a comparison shall yield the images closest to the query image.

2. **Facial Detection, Facial Recognition, Emotion Detection**
   Viola-Jones Algorithm along with Correlation Algorithm is used to carry out Emotion Detection on the images in the gallery but before the emotion is detected, Facial Detection and Facial Recognition must be carried out to detect the face in the image and then fetch relative images.

3. **Storage and Access Methods**
   The application should enable the user to upload images in a separate folder and store fetched images in a separate folder which can be accessed later. The GUI should be user friendly and convenient to use.

3.1.2 Use Case Diagram

Use case diagrams are used to model the interactions of a system with its environment. Each use case identifies different types of users and their interaction with the system.

Figure 3.2 depicts the use case diagram of Pixort. The actors are photographers or just regular users who may want to organize their gallery or retrieve selected images for an album.

The user will be capable of performing the following tasks:

- Uploading a query image as a reference for CBIR and Emotion Detection for fetching relative images.
- Uploading his images into a gallery folder. CBIR and Emotion Detection will be carried out on these uploaded images.
- Fetch images relative to the query image based on CBIR.
- Fetch images relative to the query image based on Emotion Detection.

- View fetched images in a separate folder.
- Save and access this folder of fetched images whenever required.

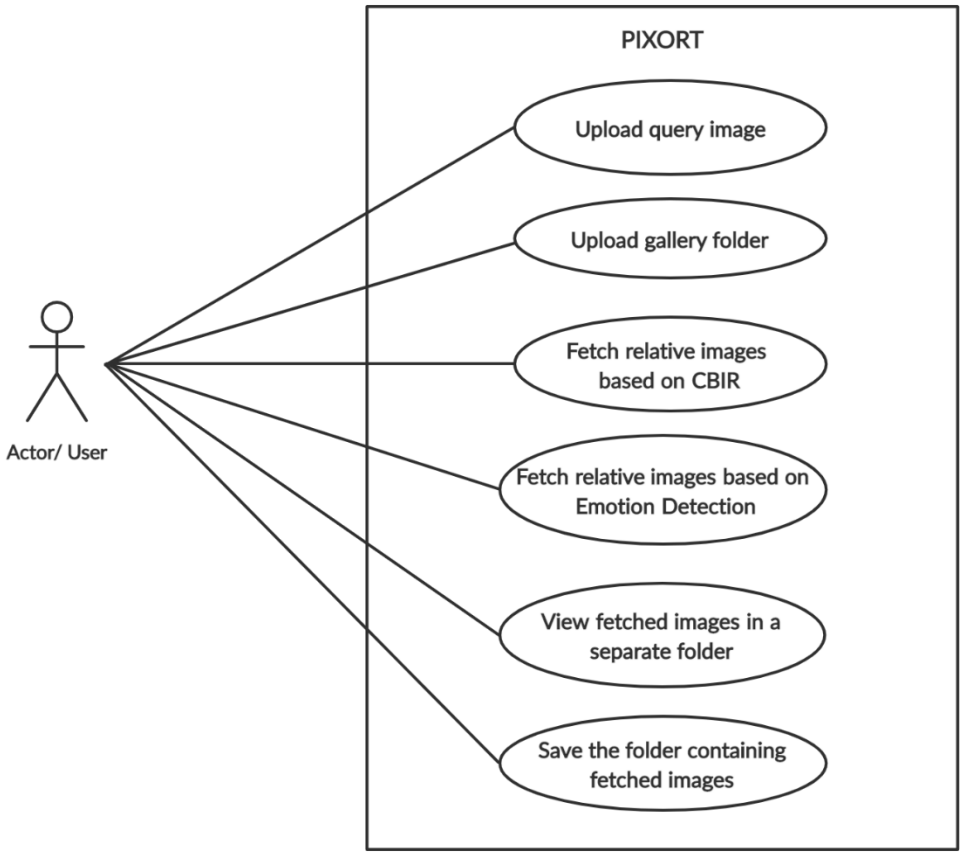

Figure 3.2 Use Case Diagram

3.2 Functional Design

3.2.1 Modular Design Diagram

A module is defined as a unique and addressable component of the software that can be modified and solved independently without interruption from other modules of the software. Modularization is defined as the process of breaking down software into multiple independent modules where each module is developed separately. Figure 3.3 depicts the modular design diagram for Pixort.

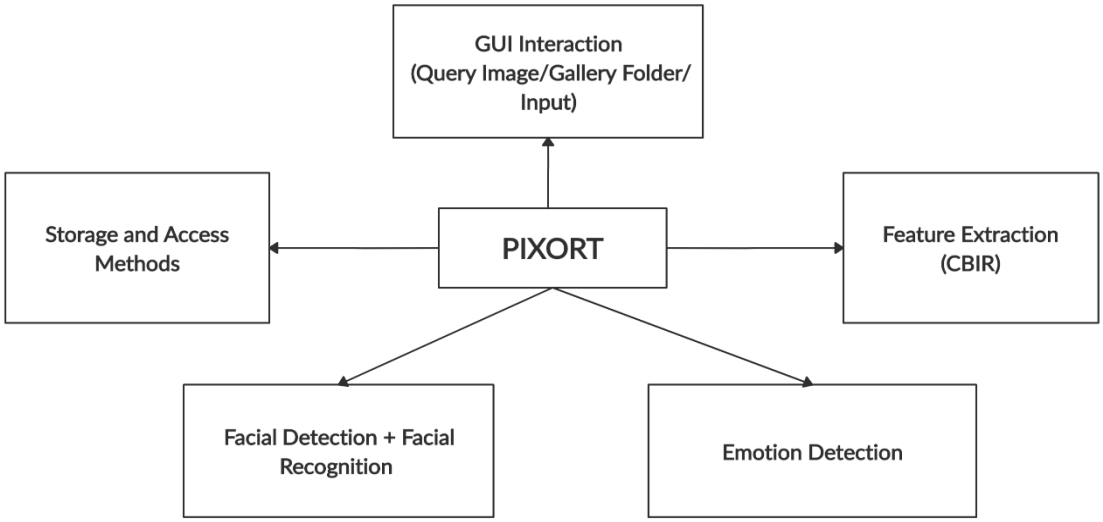

Figure 3.3 Modular Design Diagram

Pixort application will consist of the following modules as shown in figure 3.3.

1. **GUI Interaction**

   The GUI will enable the user to efficiently use the application and its functionalities. It must be user friendly and neat.

2. **Feature Extraction**

   This module forms the basis for CBIR. Different features like texture and color have to be extracted in order to find images relative to the query image.

3. **Facial Detection and Facial Recognition**

   This forms the basis for Emotion Detection. After a query image has been provided as input, the application first has to detect the presence of faces and subsequently recognize them. Only after this the emotion can be detected.

4. **Emotion Detection**

   This module is capable of fetching the relative sad and happy images from the folder of images uploaded by the user, based on the face detected in the query image.

5. **Storage and Access Methods**

   This module should allow the user to store and access the fetched images in separate

folders efficiently.

3.2.2 Sequence Diagram

Sequence diagrams are models that describe the sequence of interactions that take place for each significant interaction. They visually model the flow of logic within the system, enabling the documentation and validation of the logic.

Sequence diagrams are used primarily to design, document and validate the architecture, interfaces and the logic of the system by describing the chain of actions that have to be performed to complete the task.

3.2.2.1 Sequence Diagram for Content Based Image Retrieval

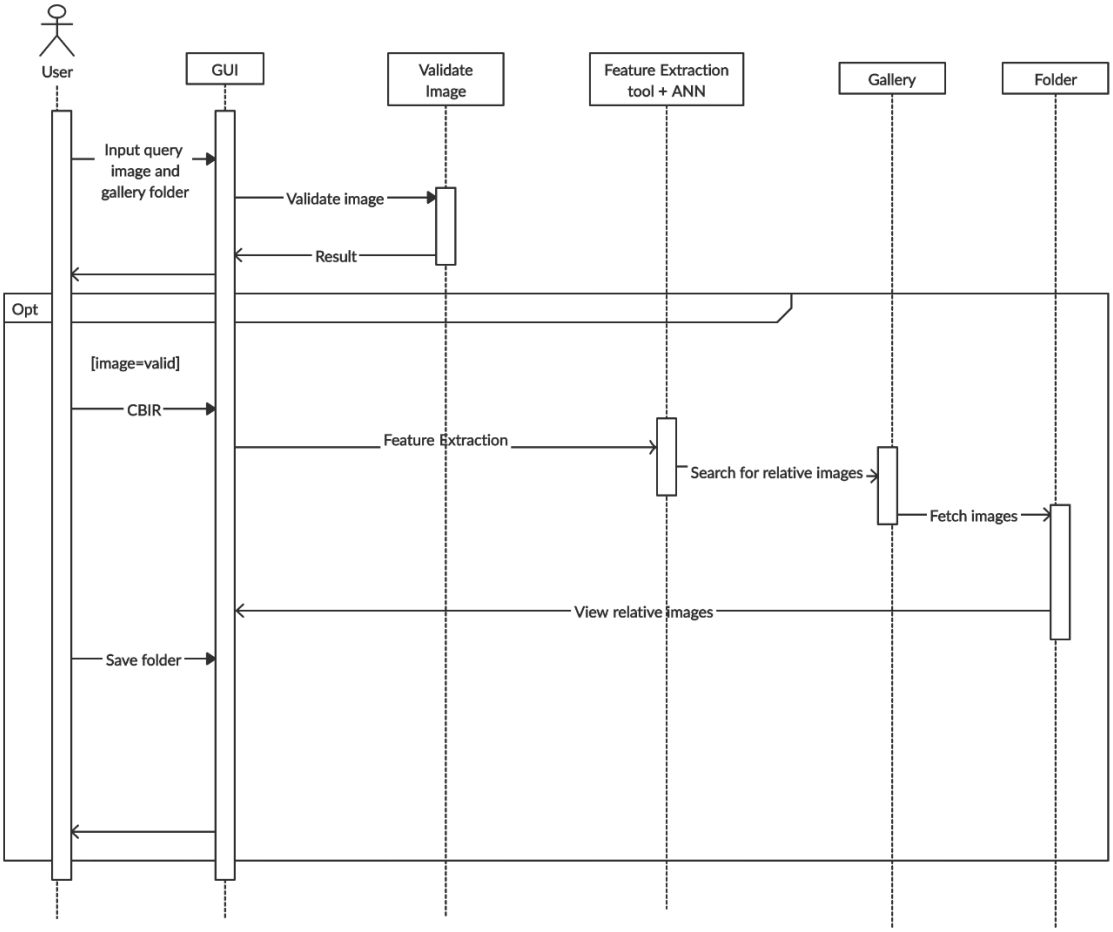

Figure 3.4 Sequence Diagram for CBIR

In figure 3.4, upon input of the query image and gallery folder by the user, the image is validated and the result is notified to the user. If the image is valid, features are extracted from the image and compared with the gallery of uploaded images to fetch relative images. These fetched images can then be viewed by the user and saved in a folder for later access.

3.2.2.2 Sequence Diagram for Emotion Detection

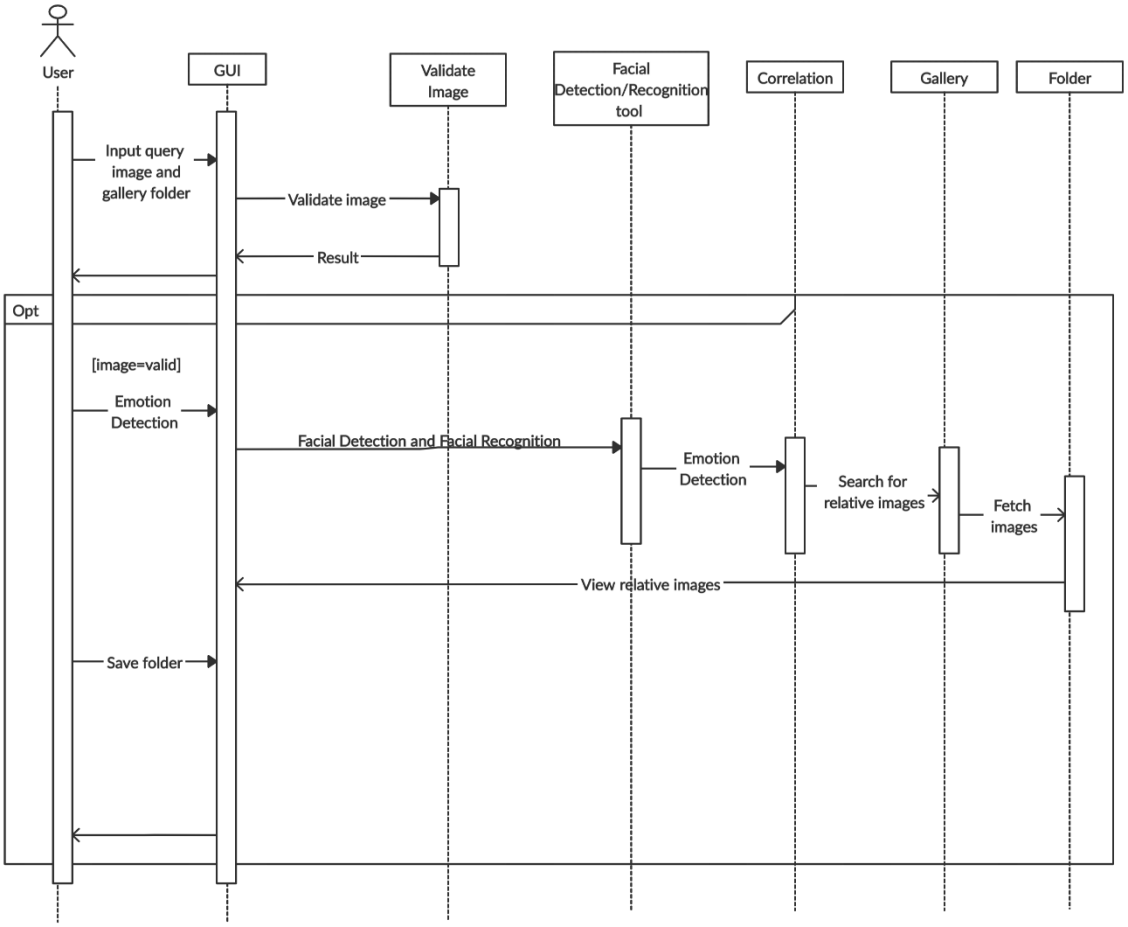

Figure 3.5 Sequence Diagram for Emotion Detection

In figure 3.5, upon input of the query image and gallery folder by the user, the image is validated and the result is notified to the user. If the image is valid, faces are detected and recognized followed by Emotion Detection. Once the face is recognized, the image is compared with the gallery of uploaded images based on similar faces and emotions to fetch relative happy and sad images. These fetched images can then be viewed by the user and saved in a folder for later access.

3.3 Control Flow Design

3.3.1 Complete System Flow Diagram

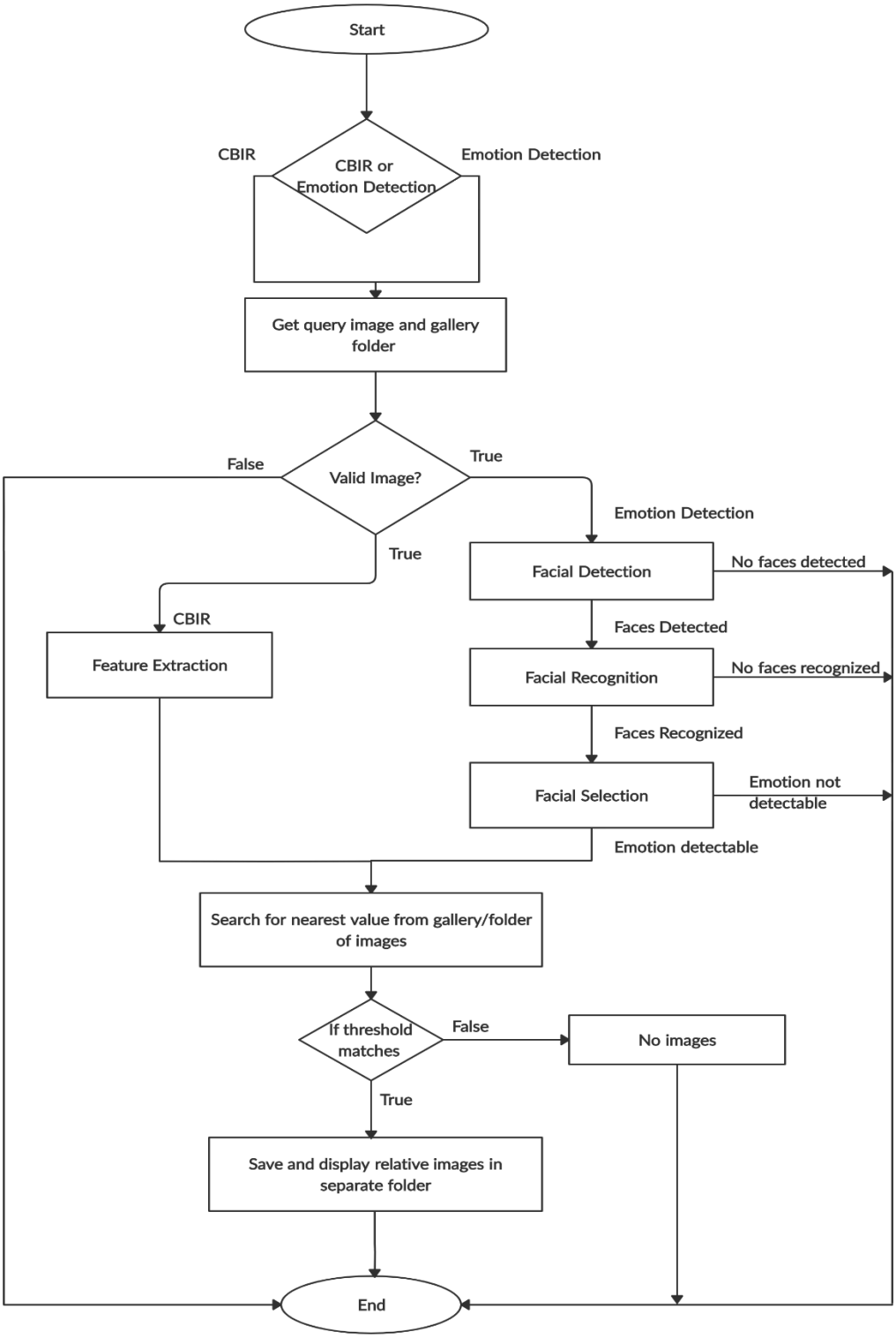

Figure 3.6 System Flow Diagram

As shown in figure 3.6, the user is first asked to choose between CBIR and Emotion Detection. In both the cases, he has to upload a query image and gallery folder. After uploading the image and folder, it is validated and if the query image is valid, the user can then proceed with CBIR or Emotion Detection.

If the user had selected CBIR, feature extraction is performed on the query image using different algorithms for parameters based on color, texture, etc. and once feature extraction is complete, the application will look for nearest values from the folder of uploaded images.

If the user selects Emotion Detection, Facial Detection and Recognition is carried out in the query image and then the emotion is predicted using Correlation algorithm. In case any of the above steps fail, an error message will be displayed and not proceed further. If the face is recognizable, the application will look for nearest values from the folder of uploaded images.

In both the cases, if the degree of similarity crosses a particular threshold, those particular images are fetched and displayed in a separate folder which can be accessed by the user later. In the case of Emotion Detection, the sad and happy images are fetched separately in two folders which can be viewed and saved. If no image is found, a message will be displayed thereby leading to the end.

3.3.2 Algorithms for Logic Implementation

3.3.2.1 Algorithms in Feature Extraction

Feature extraction is a type of dimensionality reduction technique that represents significant parts of an image as a feature vector. In Pixort, the color and texture features of the images are extracted and stored. The image is first converted from sRGB color space to L*a*b* color space, where L* denotes lightness, a* denotes the red/green coordinate and b* denotes the blue/yellow coordinate. The texture of the resultant image is then extracted using Local Binary Pattern (LBP) and the pixel information is displayed. Then, the mean LBP value of all the pixels is calculated and stored. This process is depicted in figure 3.7.

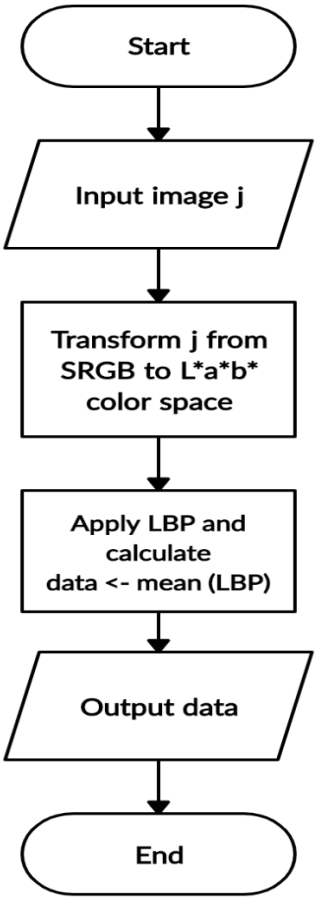

Figure 3.7 Algorithms in Feature Extraction

3.3.2.2 Algorithms in Content Based Image Retrieval

The user must first provide a query image, the gallery folder from which the relevant images have to be retrieved and a folder name for the result folder as in figure 3.8. Post validation, feature extraction will be carried out on the query image to store the mean LBP value in the variable ‘data’. A pre-trained feed forward back propagation neural network will then be loaded and simulated to identify the value of X associated with ‘data’. The function Identify_Query_Object() is then invoked to identify the object in the query image that is associated with X. Once the object has been identified, the same operation is carried out on all the images in the gallery folder which yields a value Y for every image. If the value of X equals the value of Y, the image will be saved in the result folder, and the next image from the gallery will be processed. This process continues until all images in the gallery have been processed.

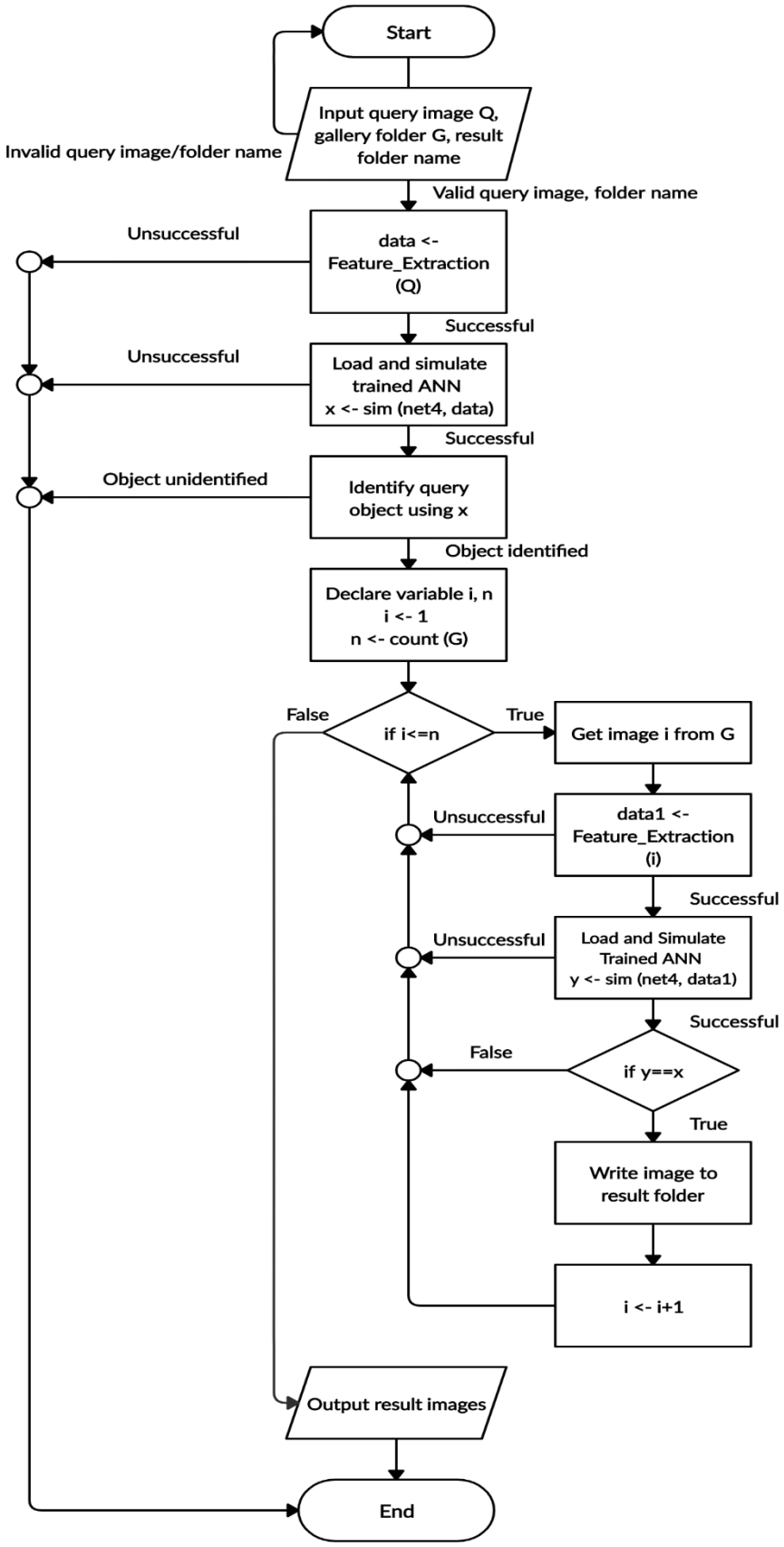

Figure 3.8 Algorithms in CBIR

3.3.2.3 Algorithms in Emotion Detection

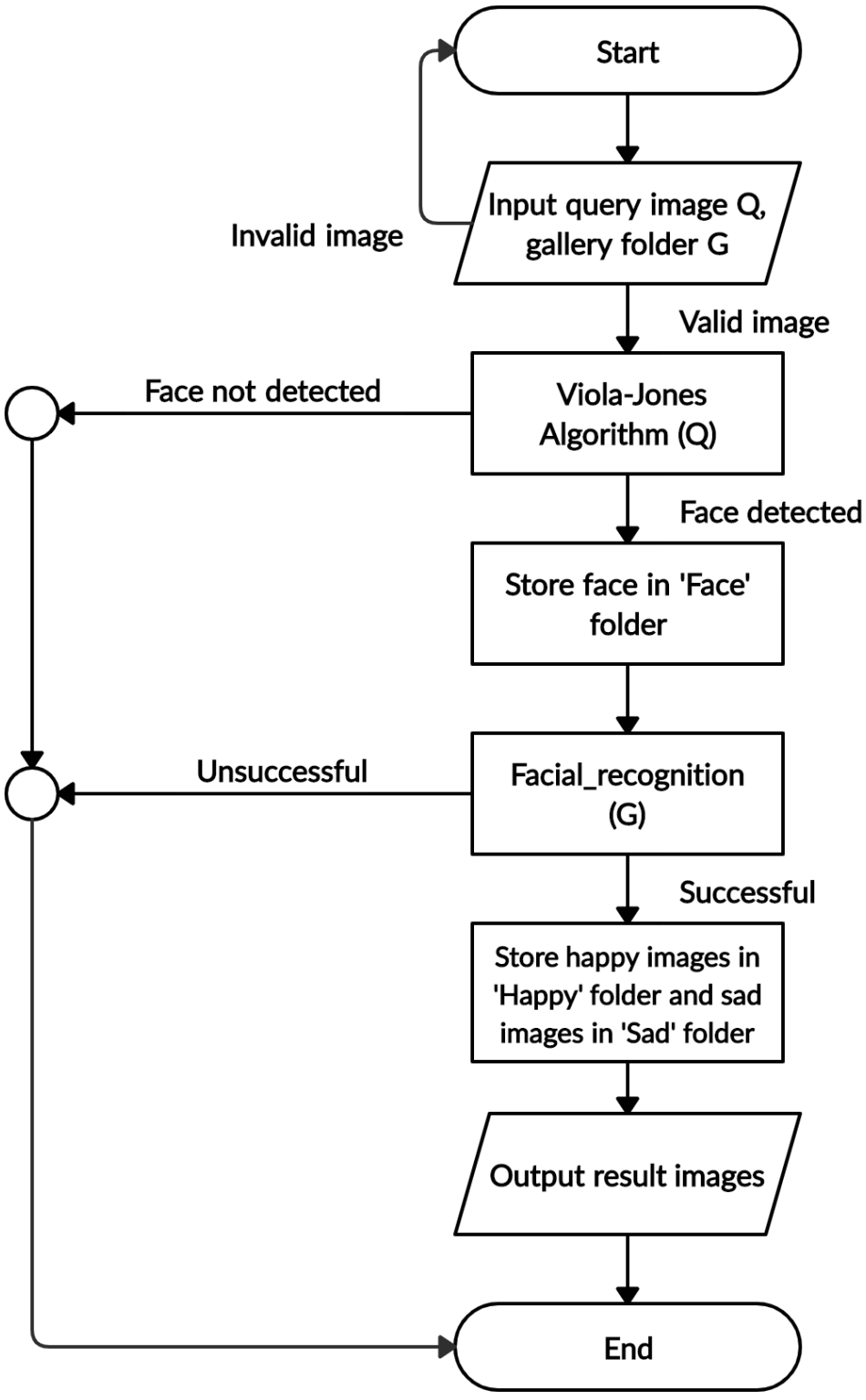

Figure 3.9 Algorithms in Emotion Detection

The user must provide a query image as input as shown in figure 3.9. After the image has been validated, Viola-Jones algorithm will be carried out on the query image. Viola-Jones algorithm is used to detect faces in images and highlights the faces by outlining a box around it. After all the faces in the query image have been detected, these faces will be cropped and stored in the 'Face' folder. The Facial_recognition() function will then be invoked to perform facial recognition and emotion recognition. The function will then categorize the images in the gallery folder as 'Happy' and 'Sad' and display those images to the user. The user may then save those images and view them in the folders 'Happy' and 'Sad' respectively.

3.3.2.4 Algorithms in Facial Recognition

In figure 3.10, the user must provide a folder name for the result folder and select the gallery folder from which the images are to be retrieved. After the folder name has been found to be valid, the user must select a face from the 'Face' folder. Feature extraction is then carried out on the selected face and the resultant value that is stored in 'data' is converted to unsigned int8. The binary file 'templates' is then loaded. This file contains the unsigned int8 LBP values of all the faces that are found after pre-processing the training dataset. The correlation between the selected face and all the faces in 'templates' is then carried out to yield the value 'vd' which is the index corresponding to the highest matched character.

The Match_faces() function is then invoked which recognizes the person associated with the value 'vd'. If a match is found, the same process is carried out on all the images in the gallery folder to retrieve the images of the identified person. The binary file 'templates2' is then loaded. This file contains the unsigned int8 LBP values of all the faces arranged according to the facial emotion found after pre-processing the training dataset. The function Emotion_Recognition() is then invoked.

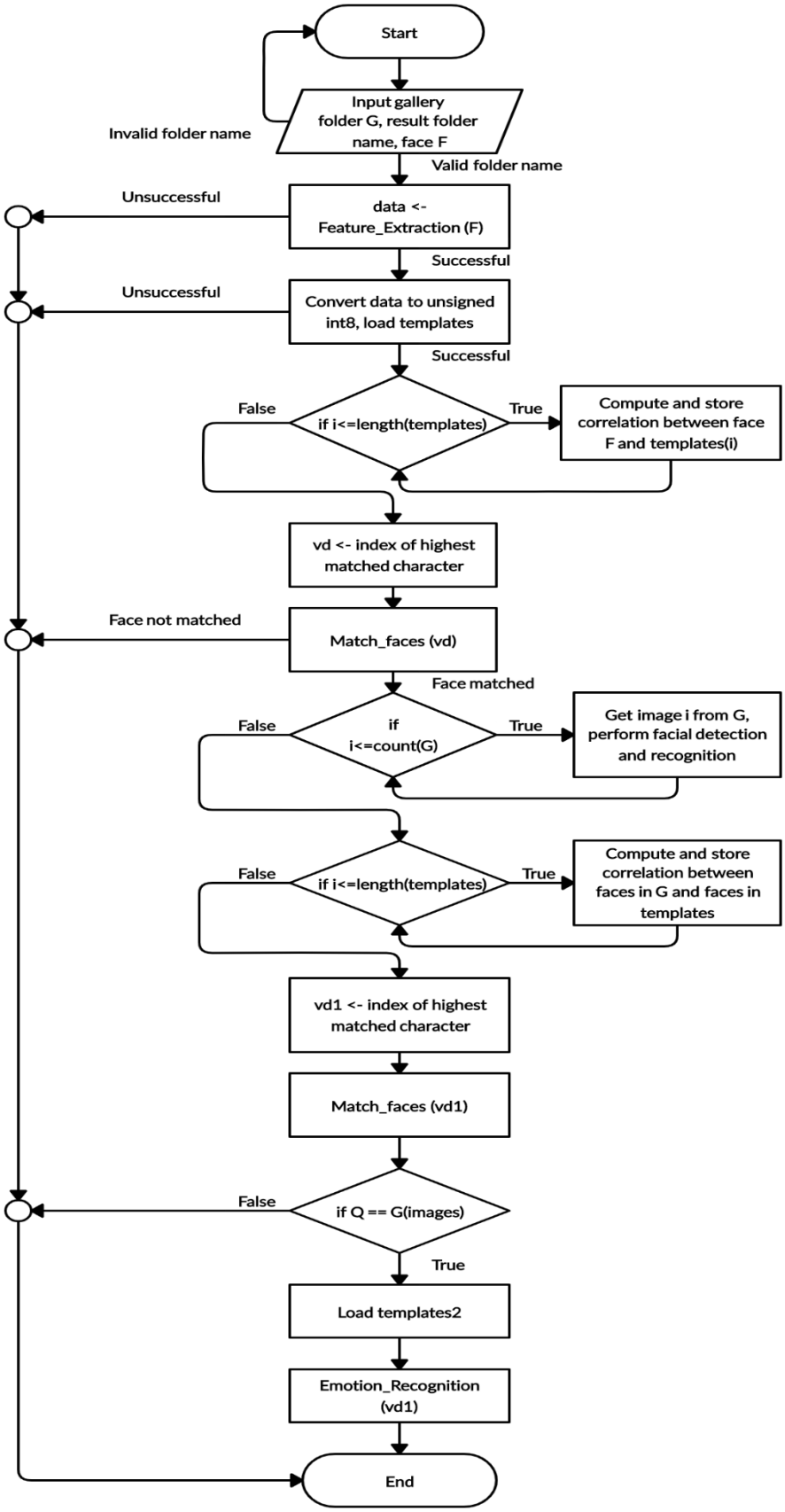

Figure 3.10 Algorithms in Facial Recognition

3.3.2.5 Algorithms in Emotion Recognition

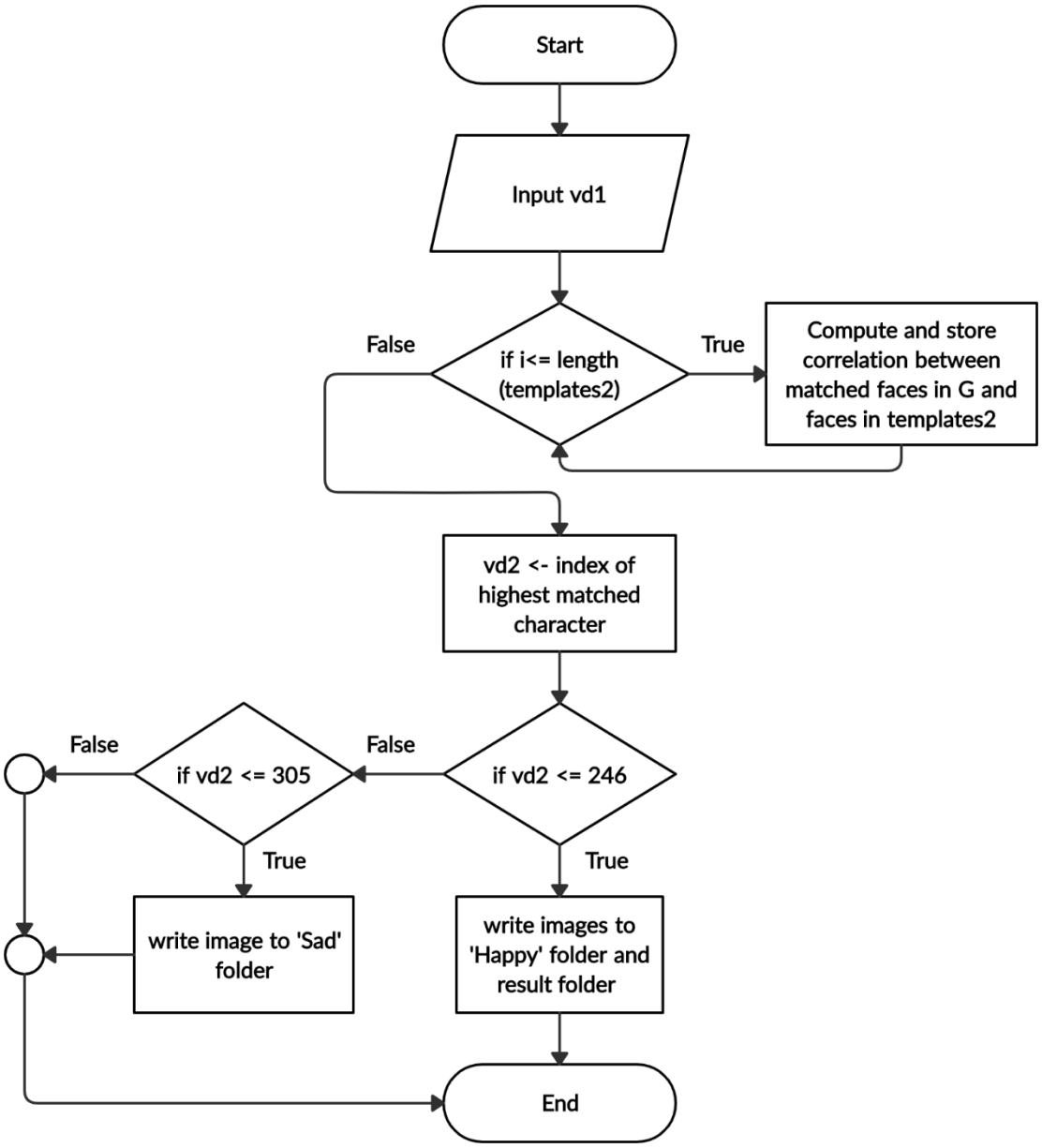

Figure 3.11 Algorithms in Emotion Recognition

The Emotion_Recognition() function computes and stores the correlation between the matched faces in the gallery folder and the faces in ‘templates2’ to yield the value ‘vd2’ which is the index corresponding to the highest matched character. Depending on the value of ‘vd2’ the images will be classified as ‘Happy’ and ‘Sad’.

3.3.3 Activity Diagram for Use Cases

UML activity diagrams describe the activities of a process and visually depict the flow of control from one activity to another. These activities may be parallel or conditional. The main objective of these diagrams is to model the workflow behind the system being designed. It may also be drawn to supplement the use case.

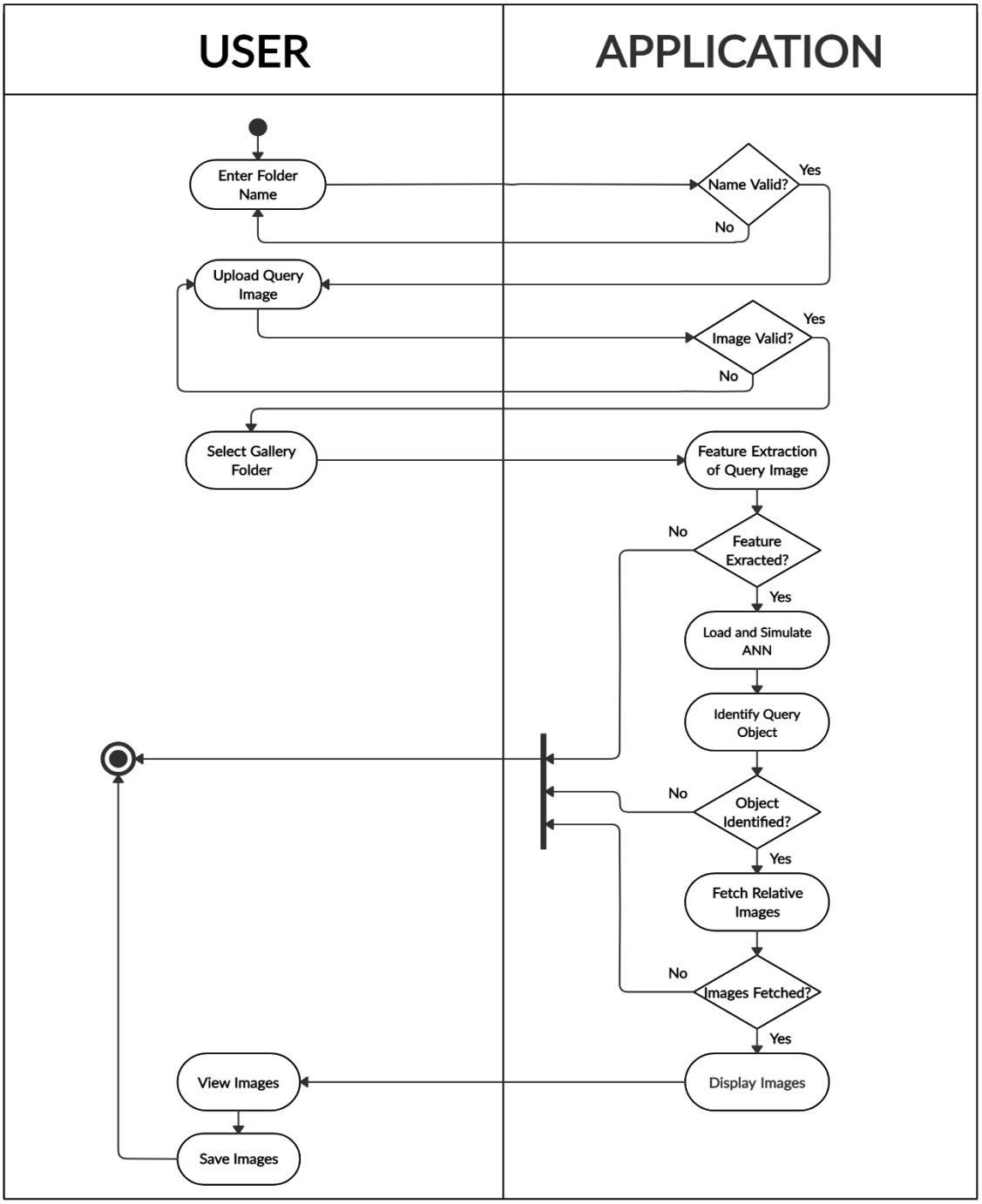

Figure 3.12 Activity Diagram for CBIR

The activity diagram for Content Based Image Retrieval and Emotion detection is shown in figure 3.12 and figure 3.13 respectively.

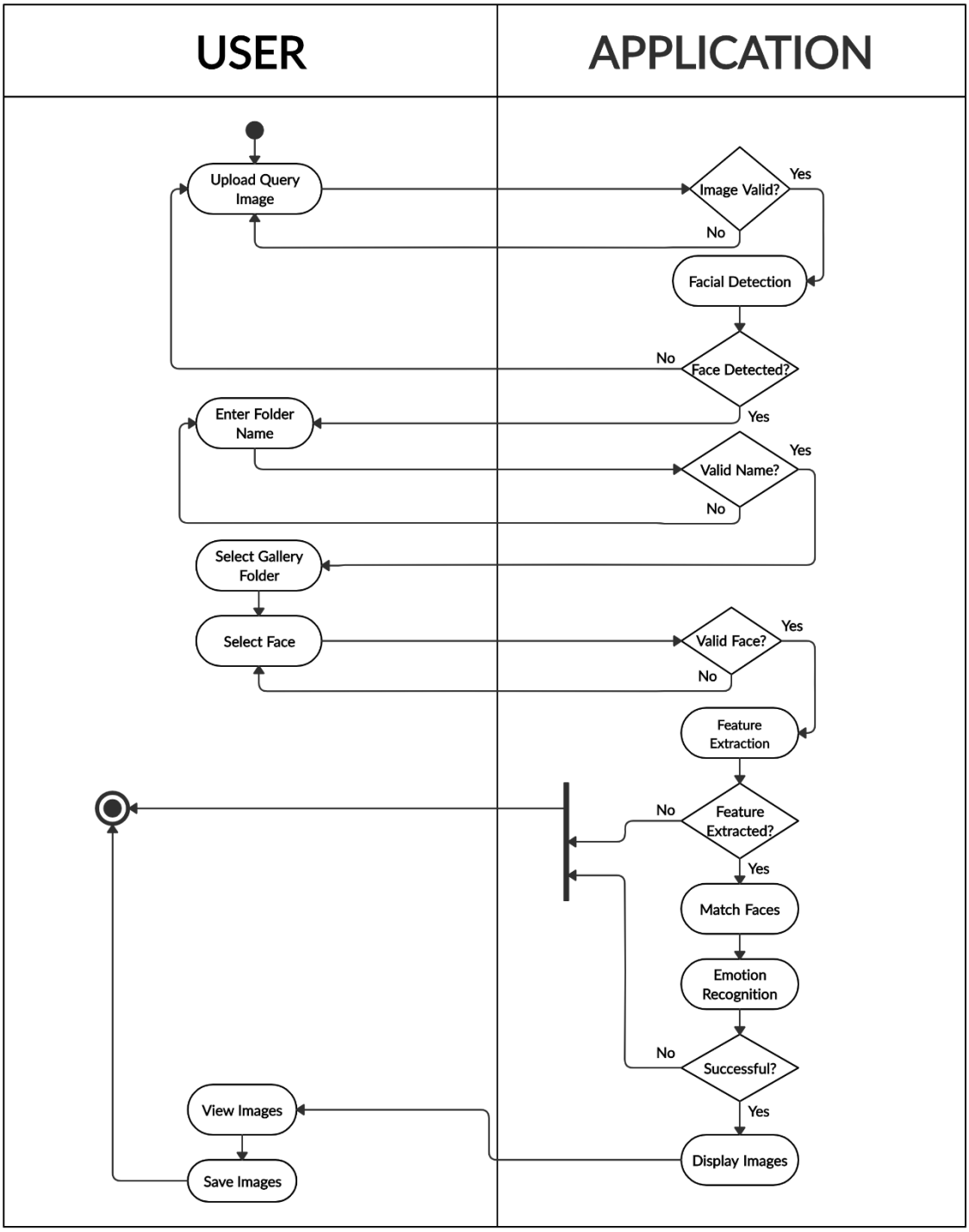

Figure 3.13 Activity Diagram for Emotion Detection

3.4 Presentation Layer Design

3.4.1 User Interface Form Design

The application requires the user to make a choice between CBIR and Emotion Detection as shown in figure 3.14.

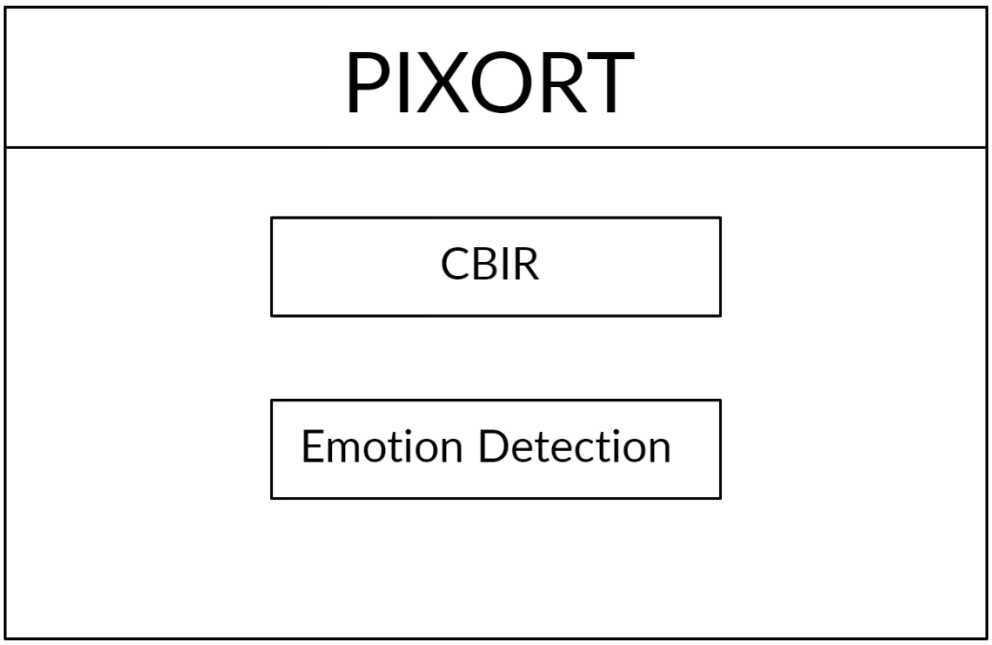

Figure 3.14 Main Interface

If CBIR is selected, the user will be redirected to the interface in figure 3.15. Here, the user must provide a result folder name, upload a query image and provide the gallery folder. The application will then fetch all the relevant images and display them. The user may then save the images and view them in the result folder.

If Emotion Detection is selected, the user will be redirected to the interface in figure 3.16. Here, the user must provide a query image for facial detection and recognition after which the user will be redirected to the interface in figure 3.17. The user must then provide a result folder name, the gallery folder and select the face whose images have to be retrieved. The application will then retrieve all the relevant images and display them. The user may then save the images and view them in the result folder.

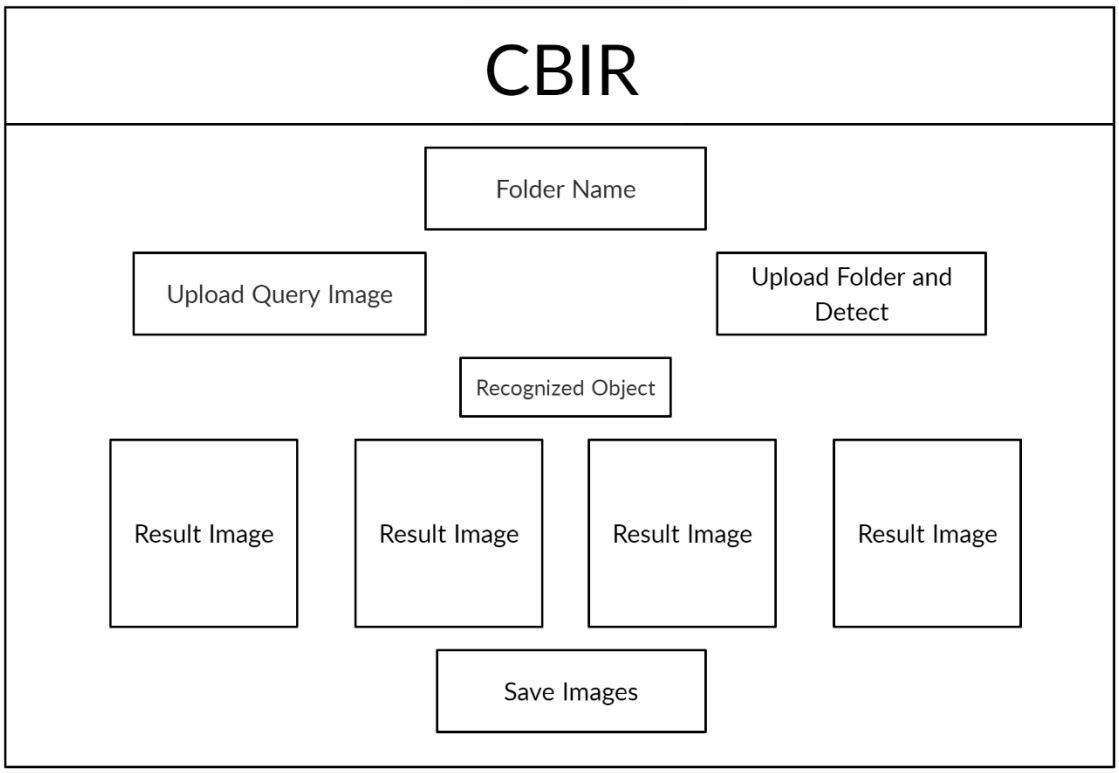

Figure 3.15 Content Based Image Retrieval Interface

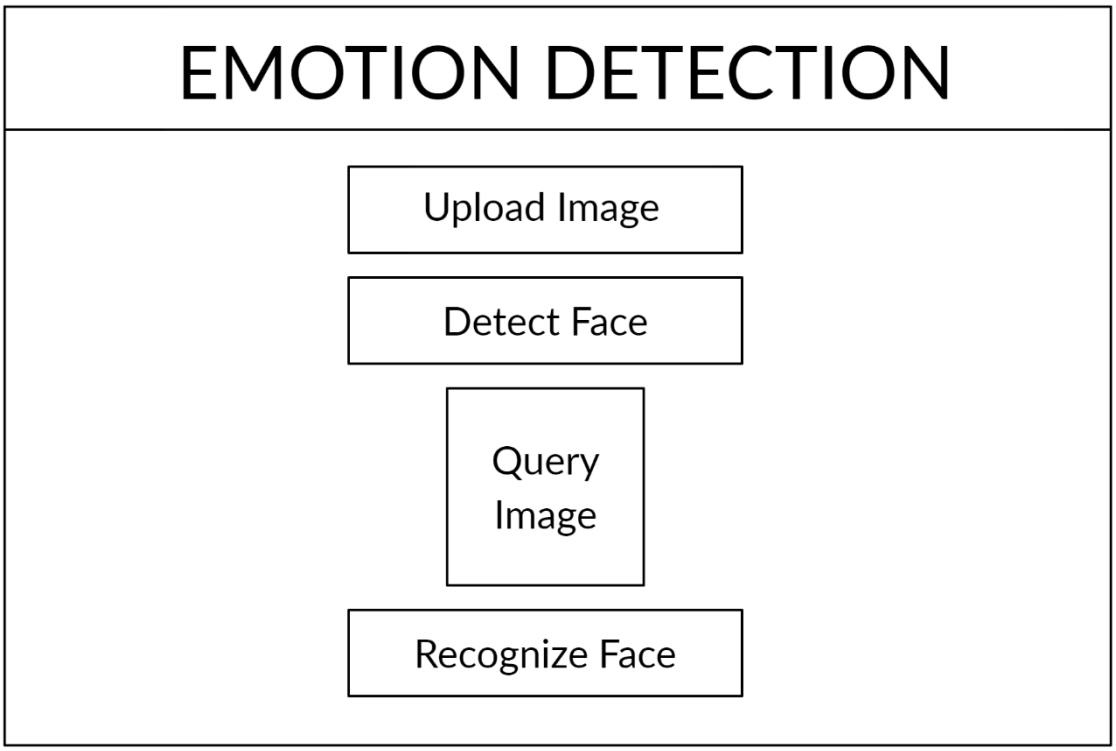

Figure 3.16 Emotion Detection (1) Interface

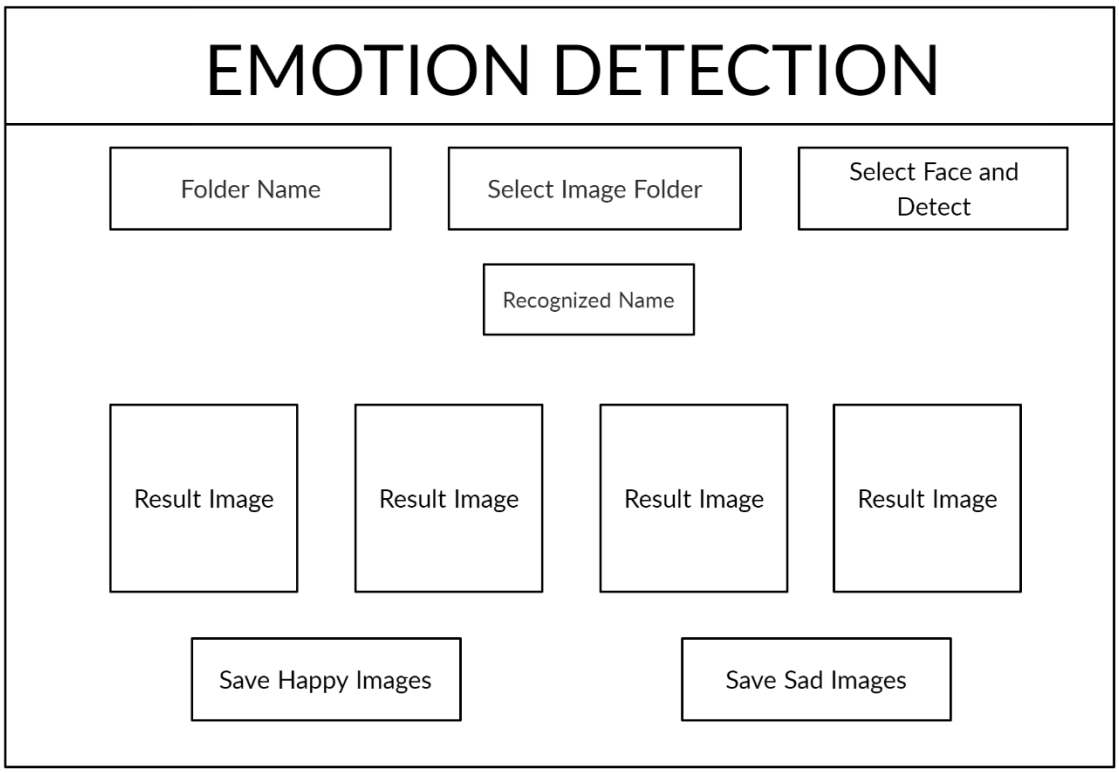

Figure 3.17 Emotion Detection (2) Interface

CHAPTER 4
IMPLEMENTATION

CHAPTER 4

IMPLEMENTATION

4.1 Hardware Used

Device Name	LAPTOP-8L6EGJ88
Processor	Intel® Core™ i3-7020U CPU @ 2.30GHz 2.30 GHz
Installed Ram	4.00 GB
System Type	64-bit operating system, x64-based processor

4.2 Software Used

Operating System	Windows 10
Graphical User Interface	MATLAB 8.4 – Guide
IDE	MATLAB 8.4
Programming Language	MATLAB

4.2.1 MATLAB

MATLAB is a multi-paradigm arithmetic computing software package and proprietary programming language developed by MathWorks for plotting of data and functions, matrix manipulations, creation of user interfaces and implementation of algorithms.

Though MATLAB is slower than some languages, it provides many built-in functions for technical computation, computer vision, neural networks, and ensures a suitable environment for image processing and deep learning.

In this project, MATLAB version 8.4 is used. In MATLAB, Graphical User Interfaces are generated either programmatically or using visual design environments such as GUIDE and App Designer.

4.2.2 Operating System

An Operating System (OS) is a system software that acts as an interface between the user

and hardware, manages computer hardware and software resources and provides necessary services for computer programs.

4.2.2.1 Windows 10

Microsoft Windows 10 is a series of operating systems developed by technology giant Microsoft co-operation and released as part of its Windows NT family of operating systems. It is one of the most widely used operating system on personal computers and is compatible with MATLAB 8.4.

4.3 Algorithms Used

4.3.1 Local Binary Pattern (LBP)

It is an efficient image operator which is used to transform an image into an array or image consisting of integer labels describing textures and pixel intensity of the image. Local binary pattern operator is generally applied on a 3×3 pixel.

$$\mathrm{LBP}(x_c, y_c) = \sum_{i=0}^{7} s(g_i - g_c)2^i \quad \text{……...……. (4.1)}$$

$$s(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

where,

centre pixel value g_c determines the threshold for each neighbourhood pixel g_i in each block

i ←sampling points (p = 0, 1, …, 7 for a 3x3 cell, where $y_c = 8$)

x_c← radius (for 3x3 cell, it is 1).

4.3.2 Feed Forward Backpropagation Neural Network

In a multilayer feed-forward neural network, Backpropagation algorithm which is a supervised learning method is used for training weights.

In the backpropagation approach, internal weightings of input signals are modified to produce an expected output signal thereby modeling a given function. The system is trained using a supervised learning method, where the difference between the system's output and the known expected output, that is, the error, is presented to the system and used to alter its internal state.

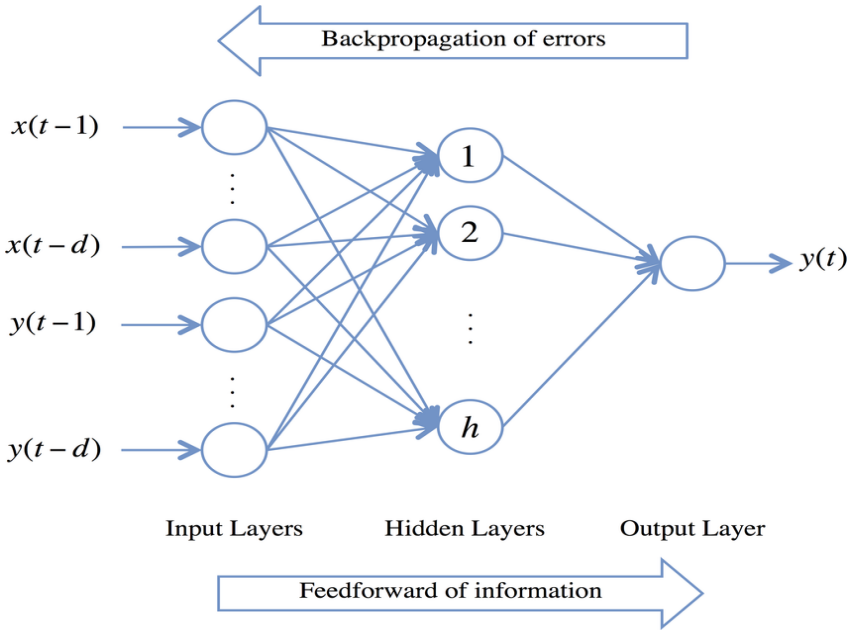

Figure 4.1 Feed Forward Backpropagation Neural Network

4.3.3 Viola-Jones Algorithm

The Viola-Jones algorithm is the most commonly used mechanism for Facial Detection. In this algorithm, training is slow however, detection is fast. The algorithm does not use multiplications. It uses Haar basis feature filters. Detection is carried out inside a detection window. A minimum and maximum window size is chosen, and for each size a sliding step size is chosen. Then the detection window is moved across the image as follows:

1. A minimum window size and corresponding sliding step is set.
2. For the window size which is chosen, slide the window horizontally and vertically with the same step. At each step, a set of *N* facial recognition filters are applied. If any filter gives a positive response, the face is detected in the current window.

3. When the window reaches the maximum size, stop the procedure. If not, increase the size of the window and the corresponding slide step to the next chosen size and go to step 2.

4.3.4 Correlation Algorithm

Correlation is a term used to measure the strength of a linear relationship between two quantitative variables or arrays.

In MATLAB, corr2 function computes the correlation coefficient using,

$$r = \frac{\sum_m \sum_n (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_m \sum_n (A_{mn} - \bar{A})^2\right)\left(\sum_m \sum_n (B_{mn} - \bar{B})^2\right)}} \quad \text{..... (4.2)}$$

where $\bar{A}$ = mean2(A), and $\bar{B}$ = mean2(B), mean2 is used to compute the mean of the elements of a matrix.

4.4 Pseudocode

4.4.1 Content Based Image Retrieval

```
CBIR (folder name, query image: Q, gallery folder)
BEGIN
    if  "Upload Query Image" button is pressed then
        Delete contents of identified folder
        Redirect users to the file system
        if Q is invalid then
            return "Please select valid image"
        else
            Resize Q
        end if
        if "Upload Gallery and Detect" button is pressed then
            Get folder name
```

if folder name is existing **then**
return "Enter another folder name"
end if
if folder name is not entered or invalid **then**
return "Please enter valid folder name"
end if
mkdir (folder name)
Feature_Extraction(Q)
Load and simulate Trained Neural Network:net4
x = round (sim (net4, data))
Identify_Query_Object (net4, data)
Begin
Fetch_Relative_Images (x)
End
Display fetched images on GUI
if "Save in folder" button is pressed **then**
Redirect user to identified folder
end if
end if
end if
END

4.4.2 Feature Extraction

Feature_Extraction (Q)
BEGIN
Transform Q from sRGB to L*a*b* colour space
Apply LBP and display pixel information
Calculate mean LBP=data
END

4.4.3 Identifying the Query Object

```
Identify_Query_Object (x, net4, data)
BEGIN
        if x==1
                return msgbox('Burger')
        elseif x==2
                 return msgbox('Cat')
        elseif x==3
                return msgbox('Dog')
        elseif x==4
                return msgbox('Pizza')
        else
                return msgbox(‘Unidentified’)
END
```

4.4.4 Fetching the Relative Images

```
Fetch_Relative_Images (x)
BEGIN
        Redirect user to file system to upload gallery folder
        for i=1:(number of images in folder) do
                Get image=Ii from dataset
                Resize Ii
                Feature_Extraction (Ii)
                y = round (sim (net4, data1))
                if y==x then
                        identified folder ← Ii
                        folder name ← Ii
                end if
        end for
END
```

4.4.5 Pre-processing of Training Images

BEGIN

Invoke Cascade Object Detector

Input image dataset for training

for each image in image dataset **do**

Apply Viola-Jones Algorithm

Return bounding box values

if face is detected **then**

for every step in bounding box **do**

Mark face using rectangle

end for

for every step in bounding box **do**

Crop face and resize

Write face to destination folder

end for

end if

end for

END

4.4.6 Creating templates for Facial Recognition using Correlation

The folder ‘faces’ consists of all the faces that are found after pre-processing the training dataset.

Create_Templates (faces folder: F)

BEGIN

Invoke Cascade Object Detector

for each image in F **do**

Resize image

if dimension(image)==3 **then**

rgb2gray(image)

end if

Convert image from sRGB to L*a*b* space

Apply LBP
Convert LBP value to unsigned int8
Store corresponding value in array
end for
for each image in F **do**
Create templates using mat2cell() function
end for
END

4.4.7 Creating templates2 for Emotion Detection using Correlation

The folder ‘emotion’ consists of all the faces arranged according to the facial emotion found after pre-processing the training dataset.

Create_Templates2 (emotion folder: E)
BEGIN
Invoke Cascade Object Detector
for each image in E **do**
Resize image
if dimension(image)==3 **then**
rgb2gray(image)
end if
Convert image from sRGB to L*a*b* space
Apply LBP
Convert LBP value to unsigned int8
Store corresponding value in array
end for
for each image in E **do**
Create templates2 using mat2cell() function
end for
END

4.4.8 Emotion Detection

Emotion Detection (query image:Q, folder name, gallery folder:G)
BEGIN
if "Upload Image" button is pressed **then**
Delete face folder
Redirect to file system
Input Q
if "Detect Face" button is pressed **then**
Apply Viola-Jones algorithm
if face is detected **then**
Move face to faces folder for selection
else
return msgbox('Cant detect face')
end if
if "Recognize Face" button is pressed **then**
Facial_Recognition(G)
end if
if "View happy images in folder" button is pressed **then**
Redirect user to happy folder
end if
if "View sad images in folder" button is pressed **then**
Redirect user to sad folder
end if
end if
END

4.4.9 Facial Recognition

Facial_Recognition (folder name, G)
BEGIN
if "Select Folder" button is pressed **then**

```
Redirect user to file system to select gallery folder:G
if "Select a face" button is pressed then
        Delete contents of happy and sad folders
        Get folder name to store happy images
        if folder name is existing then
                return "Enter another folder name"
        end if
        if folder name is not entered or invalid then
                return "Please enter valid folder name"
        end if
        mkdir(folder name)
        Redirect user to file system to select a face
        Resize face
        if dimensions(face)==3 then
                rgb2gray(face)
        end if
        Convert image from sRGB to L*a*b* space
        Apply LBP
        Convert LBP value to unsigned int8
        Load templates
        for values in templates do
                Compute and store correlation between selected face and
                faces in template in array
        end for
        vd← index which corresponds to the highest matched character
        Match_faces(vd)
        if  face is matched then
                for every image in G do
                        Carry out facial detection and recognition
                end for
                for values in templates do
                        Compute and store correlation between faces in G
                        and faces in template in array
```

```
                    end for
                    vd1← index which corresponds to the highest matched
                    character
                    Match_faces(vd1)
                    if Q==G(images) then
                        Load templates2
                        Emotion_Recognition(vd1)
                    end if
                end if
            end if
        end if
END
```

4.4.10 Match Faces

```
Match_faces(vd)
BEGIN
    if vd<=21
        disp('Archana');
        x1=1;
    elseif vd<=37
        disp('Chandrakanth');
        x1=2;
    elseif vd<=100
        disp('Cheryl');
        x1=3;
    elseif vd<=134
        disp('Crystal');
         x1=4;
    elseif vd<=159
        disp('Job');
        x1=5;
```

```
elseif vd<=211
        disp('Krithi');
         x1=6;
elseif vd<=234
        disp('Manish');
         x1=7;
elseif vd<=287
        disp('Mariah');
        x1=8;
elseif vd<=305
         disp('Sahil');
         x1=9;
elseif vd<=851
         disp('Unidentified');
        msgbox 'Cant recognize face select another';
        x1=10;
else
        disp('Unidentified');
        msgbox 'Cant recognize face select another';
        x1=10;
END
```

4.4.11 Emotion Recognition

```
Emotion_Recognition(vd1)
BEGIN
    for values in templates2 do
        Compute and store correlation between matched faces in G and faces in
        templates2 in array
    end for
    vd2← index which corresponds to the highest matched character
    if vd2<=246 then
```

```
            image is classified as happy
            write image to happy folder and folder name
        elseif vd2<=305 then
            image is classified as sad
            write image to sad folder
        end
END
```

CHAPTER 5
TESTING

CHAPTER 5

TESTING

Software testing is the process of investigating, verifying and validating software or applications to ensure that they are bug free and providing the stakeholders with information related to the quality of the software or service being tested. It gives the business a wider perspective to appreciate and understand the risks involved in software implementation.

Software testing involves two steps, mainly verification and validation of properties. In general, these properties indicate the limit to which the system under test meets the requirements that lead its design and development, if the system responds accurately to vivid inputs, performs its functions within a justifiable time limit, is easily usable, can be installed and run in intended environments, and on the whole, if it achieves the expected result its stakeholders desire.

Software testing typically aims at executing a program or application with the intention of finding software bugs and also working on improving accuracy and efficiency. Testing is carried out in iterations and is an iterative process because when one bug is resolved, it may expose another.

5.1 Software Testing Levels

5.1.1 Unit Testing

In Unit testing, individual units of a software are tested. Unit testing is carried out to validate each unit of the software and its performance. This type of testing is usually done by developers on the go, to ensure that each unit is working and functioning as anticipated.

5.1.2 Integration Testing

Integration testing is where individual units are merged and tested together as a group. The aim of this level of testing is to expose incompatibilities, faults and irregularities in the

interaction between integrated modules. Integration tests involve lot of code, and may produce traces larger than unit tests.

5.1.3 System Testing

System testing is the process of testing the entire integrated system as a whole. This test is carried out to evaluate the system's end-to end compliance with the stated requirements.

5.1.4 Acceptance Testing

Acceptance testing involves testing a software product for acceptability. It is carried out to evaluate the system's conformance with the clients or stakeholders demands and assess whether it is acceptable for deployment.

5.2 Test Cases

5.2.1 Main Interface

Table 5.1 Test Cases for Main Interface

No.	Test Cases	Expected Output	Observed Output	Result
1.	The CBIR button is clicked	The user must be redirected from home interface to CBIR interface	The user is redirected from the home interface to the CBIR interface	Pass
2.	The Emotion Detection button is clicked	The user must be redirected from home interface to Emotion Detection (1) interface	The user is redirected from the home interface to the Emotion Detection (1) interface	Pass

5.2.2 Content Based Image Retrieval Interface

Table 5.2 Test Cases for CBIR Interface

No.	Test Cases	Expected Output	Observed Output	Result
1.	Invalid folder name is entered	Warning message to instruct user to input a valid folder name	A message box is displayed instructing the user to input a valid folder name	Pass
2.	Existing folder name is entered	Warning message to inform user that the folder already exists and its contents will be overwritten. If the user wishes, he may enter another valid folder name	A message box is displayed informing the user that the folder already exists and its contents will be overwritten. If the user wishes, he may enter another valid folder name	Pass
3.	Query image of format other than .png or .jpg is selected	File system does not allow selection of the image	File system does not allow selection of the image	Pass
4.	Detect button is pressed without a valid input query image or folder name	Warning message to inform user to give valid query image and folder name	A message box is displayed informing users to give valid query image and folder name	Pass
5.	CBIR is carried out without folder name being specified	Warning message to enter a valid folder name	A message box is displayed instructing the user to enter a valid name	Pass

6.	Upload query image button is pressed after entering valid folder name	Redirect the user to the file system to select a valid query image	The user is redirected to the file system and can only select an image file, that is, JPEG or PNG	Pass
7.	Detect button is pressed after entering valid folder name and uploading valid query image	Window with LBP pattern of query image must be displayed and 4 relative images must be fetched on the GUI	Window with LBP pattern of query image is displayed and 4 relative images are displayed on the GUI	Pass
8.	Save images in folder button is pressed after entering valid folder name, uploading valid query image, pressing detect button	Redirect to folder with fetched images	The user is redirected to the folder containing the fetched images	Pass
9.	4 relative images are not fetched	The fetched images should be displayed on GUI irrespective of number	The fetched images are displayed on GUI irrespective of number	Pass

5.2.3 Emotion Detection (1) Interface

Table 5.3 Test Cases for Emotion Detection (1) Interface

No.	Test Cases	Expected Output	Observed Output	Result
1.	Recognize Face button is pressed without uploading query image	Warning message to instruct user to upload query image	A message box is displayed instructing the user to upload a query image	Pass

2.	Detect face button is pressed without uploading query image	Warning message instructing user to input a valid query image	A message box is displayed instructing the user to input a valid query image	Pass
3.	The input image doesn't contain a face	Warning message to inform user that no face is detected	A message box is displayed informing user that faces cannot be detected	Pass
4.	Recognize Face button is pressed without Detect face button being pressed	Warning message to instruct user to press Detect face	A message box to instruct user to detect faces	Pass
5.	Upload Image button is clicked	Redirect the user to the file system to upload a valid query image	The user is redirected to the file system and is able to only select an image file, that is, JPEG or PNG	Pass
6.	Detect Face button is pressed after Upload Image button is pressed	Faces in query image have to be detected	Faces in query image are detected	Pass
7.	Recognize face button is pressed after query image is uploaded and Detect Face button is clicked	The user should be redirected to the next interface with recognized faces in the faces folder	The user is redirected to the next interface with recognized faces in the faces folder	Pass

5.2.4 Emotion Detection (2) Interface

Table 5.4 Test Cases for Emotion Detection (2) Interface

No.	Test Cases	Expected Output	Observed Output	Result
1.	Invalid folder name is entered	Warning message to instruct user to input a valid folder name	A message box is displayed instructing the user to enter a valid folder name	Pass
2.	Existing folder name is entered	Warning message to inform user that the folder already exists and its contents will be overwritten. If the user wishes, he may enter another valid folder name	A message box is displayed informing user that a similar folder already exists and its contents will be overwritten. If the user wishes, he may enter another valid folder name	Pass
3.	Select a face and detect button is pressed after entering a valid folder name and selecting the image folder	Redirect the user to the faces folder to select a valid face	The user is redirected to select a face in the faces folder	Pass
4.	An invalid face is selected in the faces folder	Warning message to inform user that the selected face cannot be recognized and that he needs to select another face	A message box is displayed informing the user that the selected face cannot be recognized and that he needs to select another face	Pass

5.	Save happy / sad images in folder button is pressed without selecting a face	Warning message to instruct user to select a face	A message box is displayed instructing the user to select a face	Pass
6.	Minimum two sad/ happy images are not fetched	The fetched images should be displayed on the GUI irrespective of number	Available fetched images are displayed on the GUI	Pass
7.	View happy/sad images in folder buttons are pressed	User must be redirected to the respective folder containing the fetched happy/sad images	User is redirected to the respective folder containing the fetched happy/sad images	Pass
8.	Select image folder button is pressed	User should be redirected to file system to upload folder of images	User is redirected to file system to upload folder of images	Pass

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

CHAPTER 6

CONCLUSION AND FUTURE SCOPE

The main focus of this project was to demonstrate the significance of content based retrieval in image retrieval systems. In contrast to conventional approaches that are based on text, identifying and fetching images based on its content is an accurate and resourceful method for image retrieval. The application used two features to represent the query image and was able to identify and retrieve images similar to the query image with a training accuracy of 0.88605 and an overall accuracy of 0.81597.

A practical face detector employing Viola-Jones algorithm using cascade object detector in MATLAB was designed. The application was also able to recognize faces and categorize the images based on two emotions by measuring the strength of the relationship between the query image and images in the gallery using correlation with a precision of 0.8675 and recall of 0.972.

Content Based Image Retrieval is presently an area of active research in the deep learning and computer vision community due to its vast potential in both academic and industry ranging from medical and forensic applications to security applications that can be implemented on cloud system, mobile phones and even the World Wide Web. However, the major challenge lies in identifying features that adequately represent an image. This area can be further explored and the techniques can be fine-tuned to increase the retrieval efficiency. The neural network used can be trained extensively to fetch images on a broader spectrum and with an enhanced accuracy.

Facial and emotion detection is an indivisible part of computer vision. Emotions of the students in a classroom can be determined to assess the quality of the class. Classification of the images based on emotions has applications in automatic tagging of images with emotional categories and automatically categorizing video sequences into genres. The future scope of the application lies in the ability to detect and recognize more faces with greater accuracy and further categorize them into six or more emotions. Many tasks and processes can be carried out if one becomes conscious of the intricacies and exploits the endless possibilities offered under this field.

REFERENCES

[1] https://en.wikipedia.org/wiki/Content-based_image_retrieval Visited on: 11/04/2020

[2] Algorithmia, Introduction to Facial Emotion Recognition, 2018, Available: https://algorithmia.com/blog/introduction-to-emotion-recognition Visited on: 15/04/2020

[3] http://www.leeds.ac.uk/educol/documents/00001240.htm Visited on: 18/04/2020

[4] Afshan Latif, Aqsa Rasheed and Umer Sajid, "Content-Based Image Retrieval and Feature Extraction: A Comprehensive Review", *Mathematical Problems in Engineering*, Hindawi, 26 August 2019

[5] https://en.wikipedia.org/wiki/MATLAB Visited: 18/04/2020

[6] https://in.mathworks.com/help/images/ref/rgb2lab.html Visited on: 21/04/2020

[7] https://in.mathworks.com/matlabcentral/answers/15774-local-binary-pattern-lbp Visited on: 21/04/2020

[8] https://in.mathworks.com/help/deeplearning/ref/perform.html Visited: 22/04/2020

[9] https://in.mathworks.com/help/matlab/creating_guis/about-the-simple-guide-gui-example.html Visited on: 22/04/2020

[10] https://in.mathworks.com/help/deeplearning/ref/sim.html Visited on: 24/04/2020

[11] https://in.mathworks.com/help/vision/examples/image-retrieval-using-customized-bag-of-features.html Visited: 27/04/2020

[12] https://in.mathworks.com/help/releases/R2014a/vision/index.html Visited: 28/04/2020

[13] https://in.mathworks.com/help/images/ref/corr2.html Visited: 06/05/2020

[14] https://in.mathworks.com/help/vision/ref/vision.cascadeobjectdetector-system-object.html Visited on: 14/05/2020

[15] https://in.mathworks.com/help/matlab/creating_guis/add-code-for-components-in-callbacks.html#f10-1001464 Visited on: 20/05/2020

[16] https://in.mathworks.com/matlabcentral/answers/15389-how-to-disable-a-button-later-in-the-program-on-gui Visited on: 23/05/2020

[17] https://in.mathworks.com/matlabcentral/answers/215784-most-correct-way-to-pass-shared-data-between-callbacks-in-guide-gui-application Visited on: 25/05/2020

APPENDIX

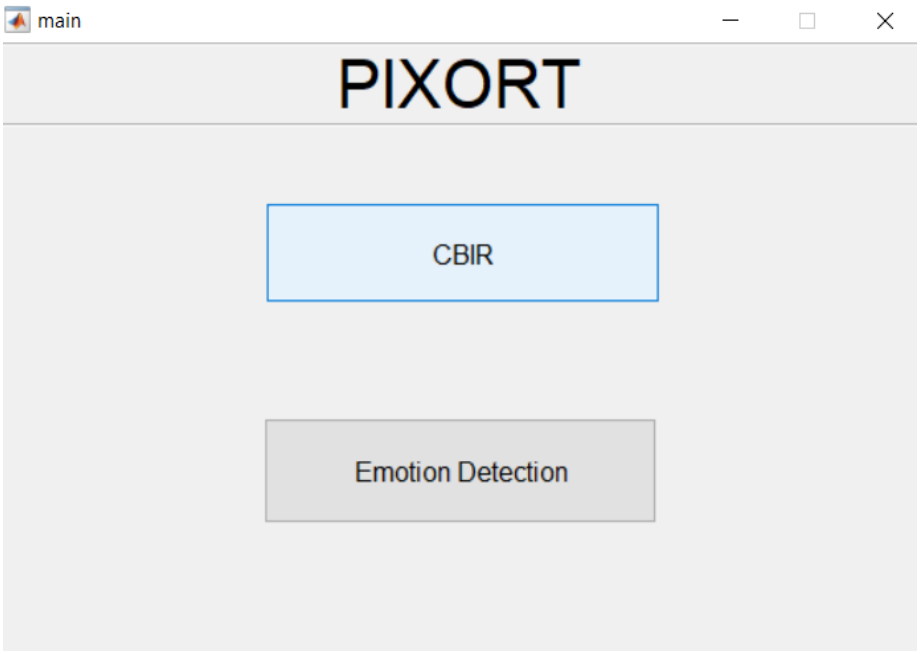

Figure 1. Selection of CBIR in the Main Interface

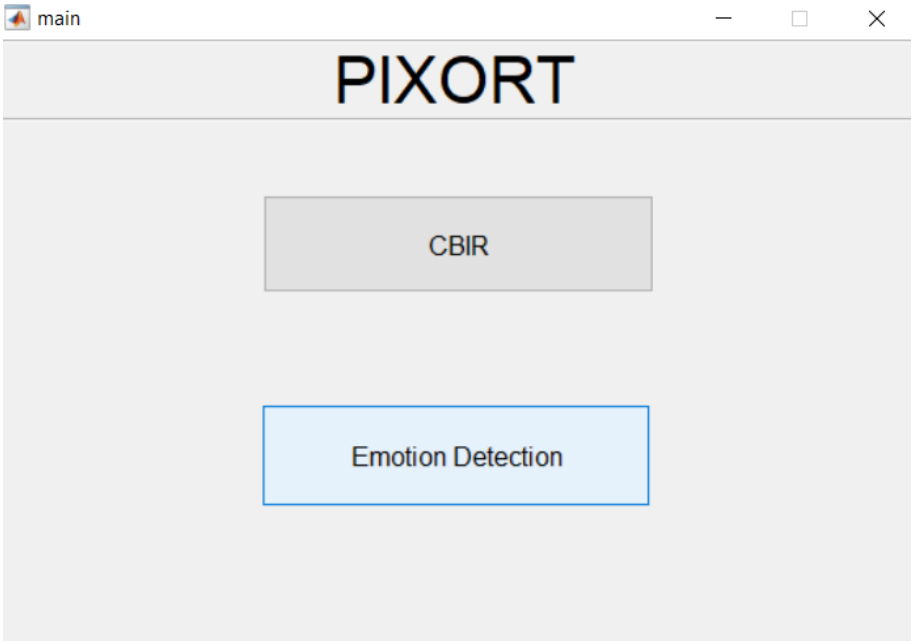

Figure 2. Selection of Emotion Detection in the Main Interface

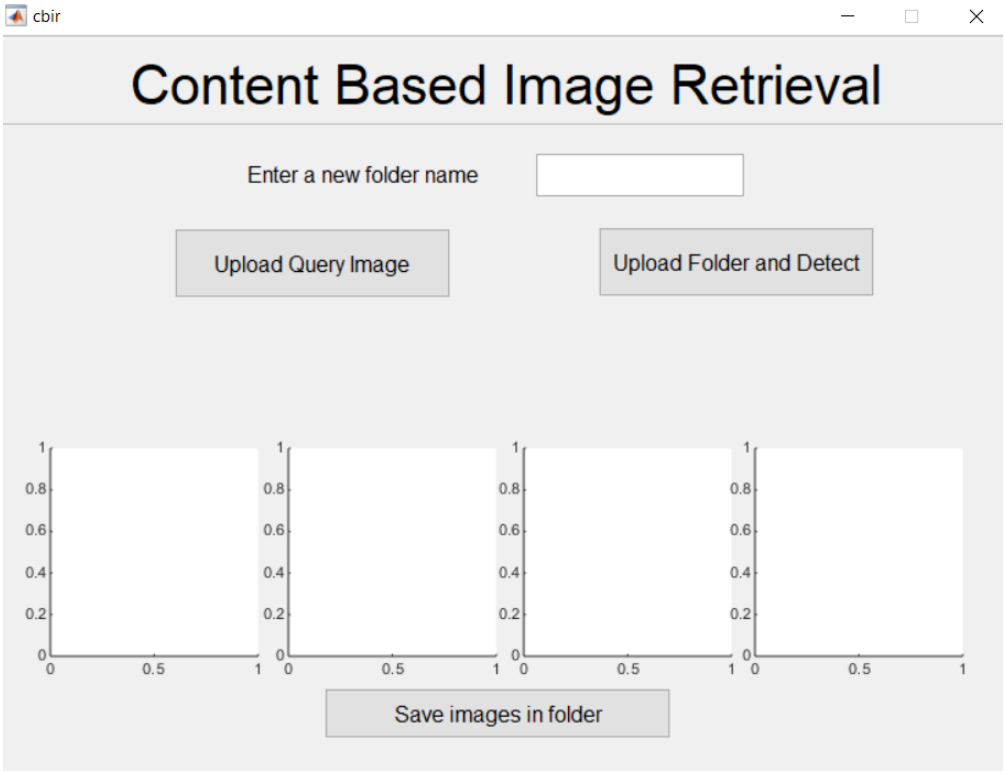

Figure 3. CBIR Interface

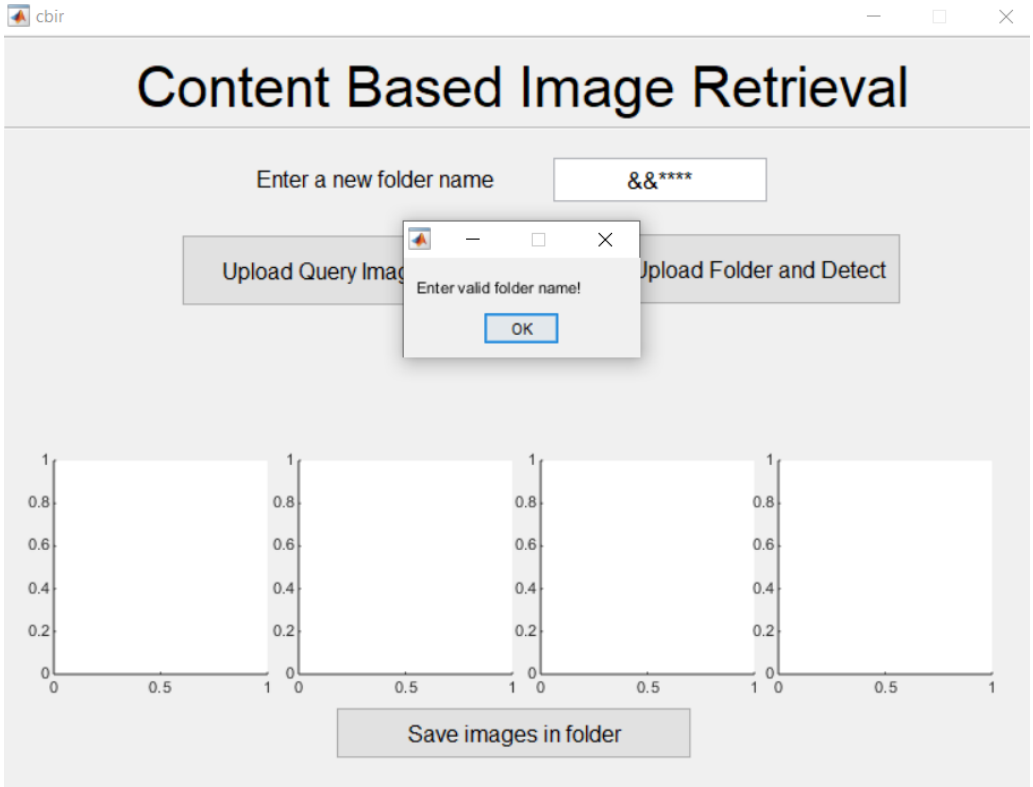

Figure 4. Invalid Folder Name is Entered

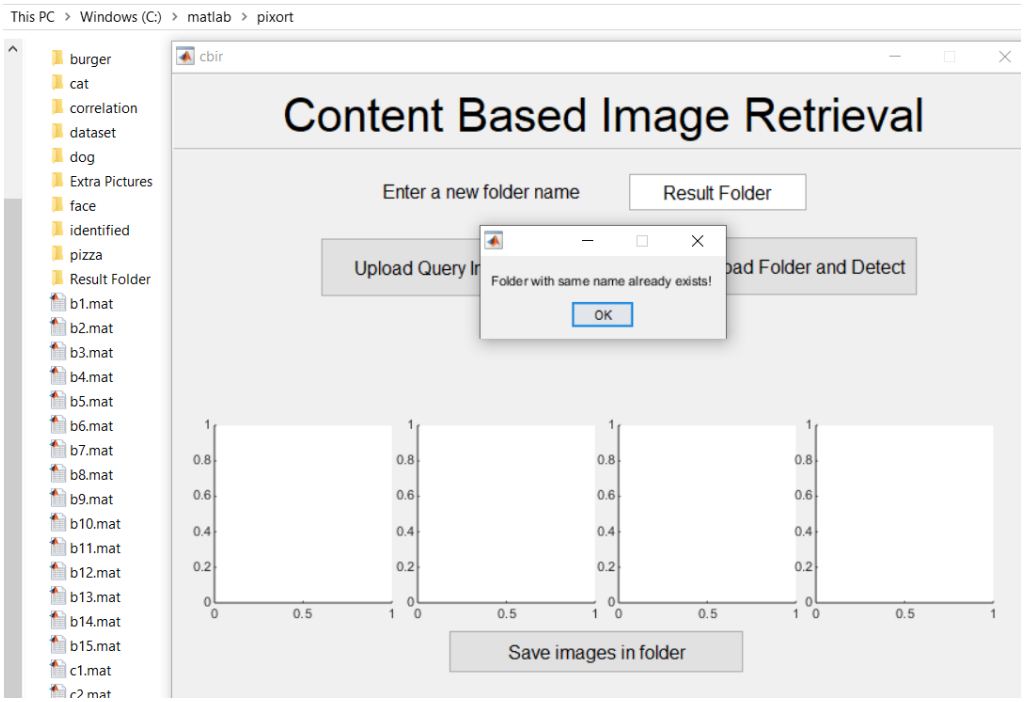

Figure 5. Existing Folder Name is Entered

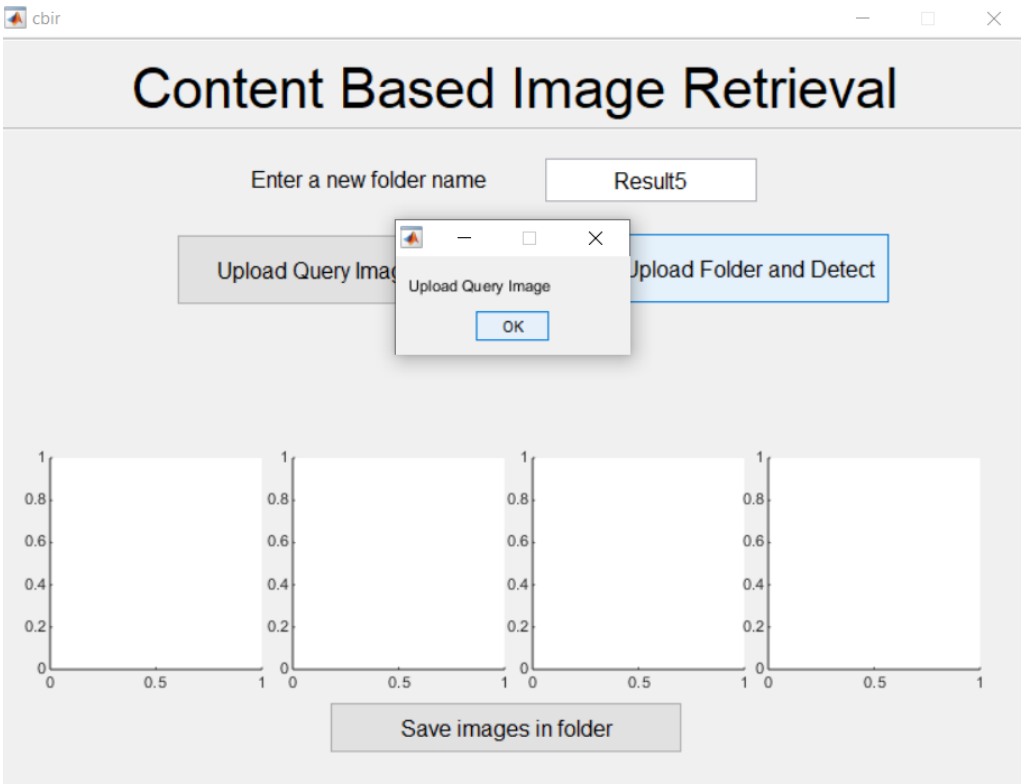

Figure 6. Upload Folder and Detect Button is Pressed without Uploading Query Image

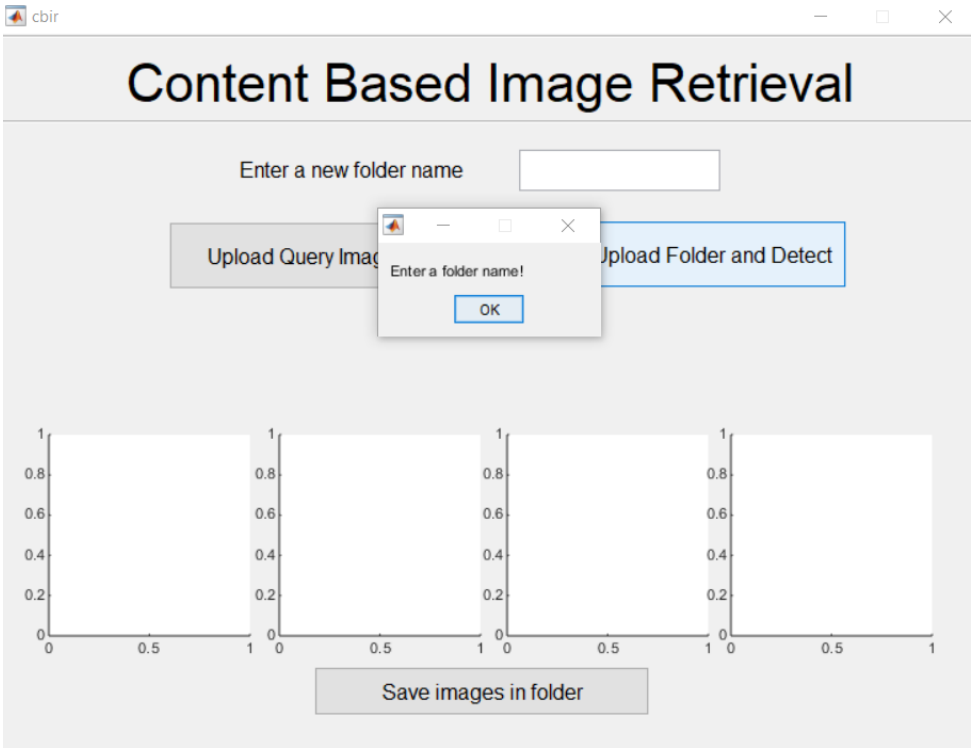

Figure 7. Upload Folder and Detect Button is Pressed without Entering Folder Name

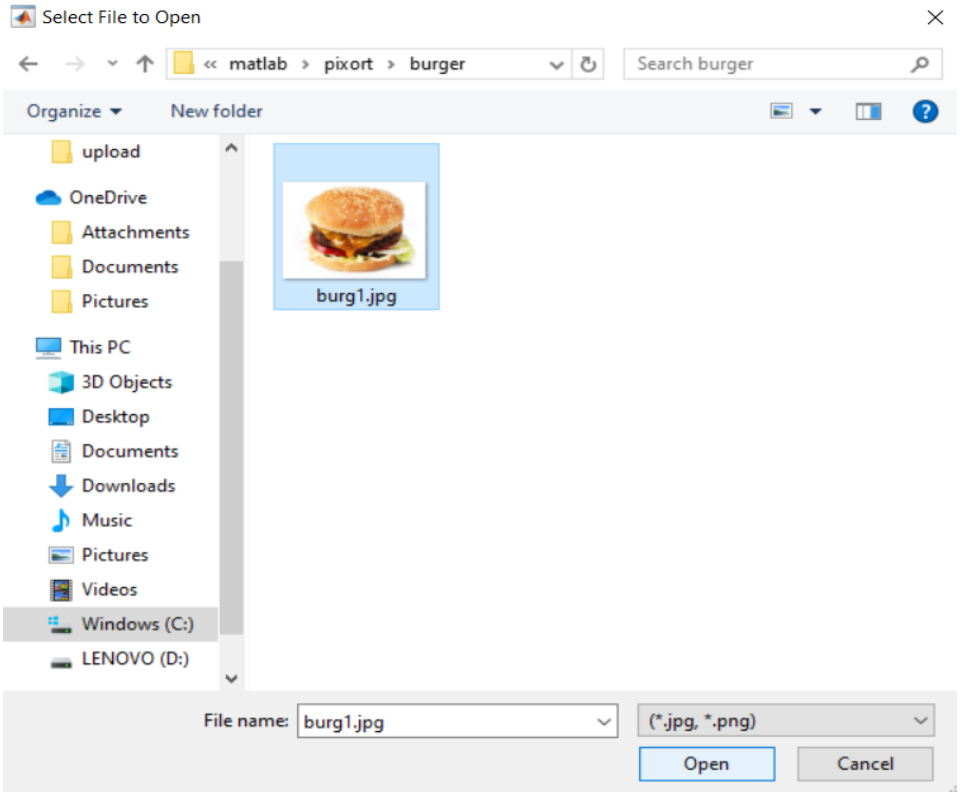

Figure 8. Selection of Query Image

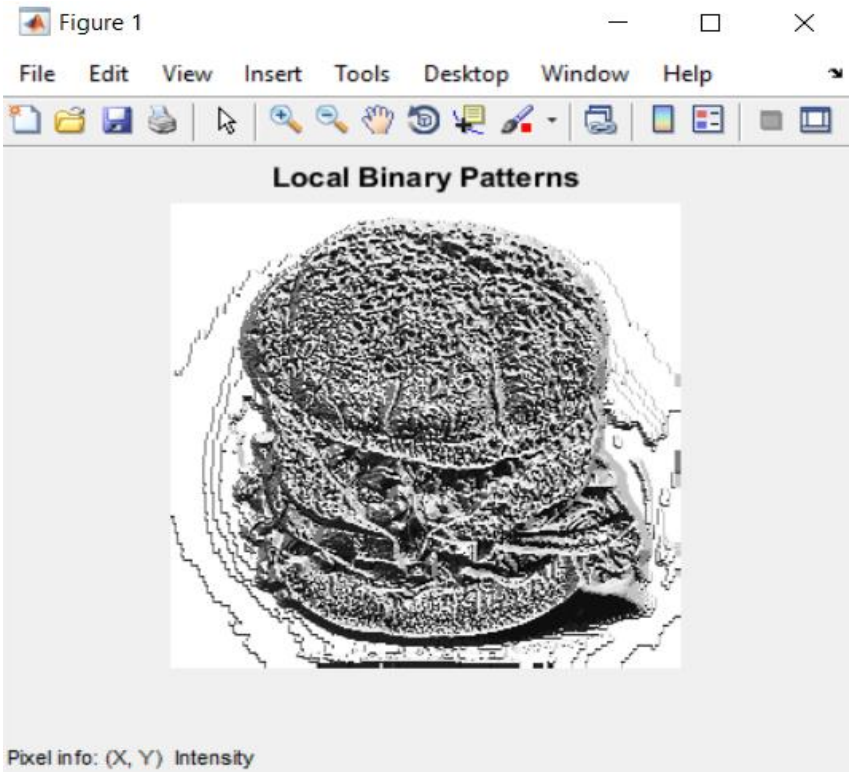

Figure 9. Local Binary Pattern of Query Image

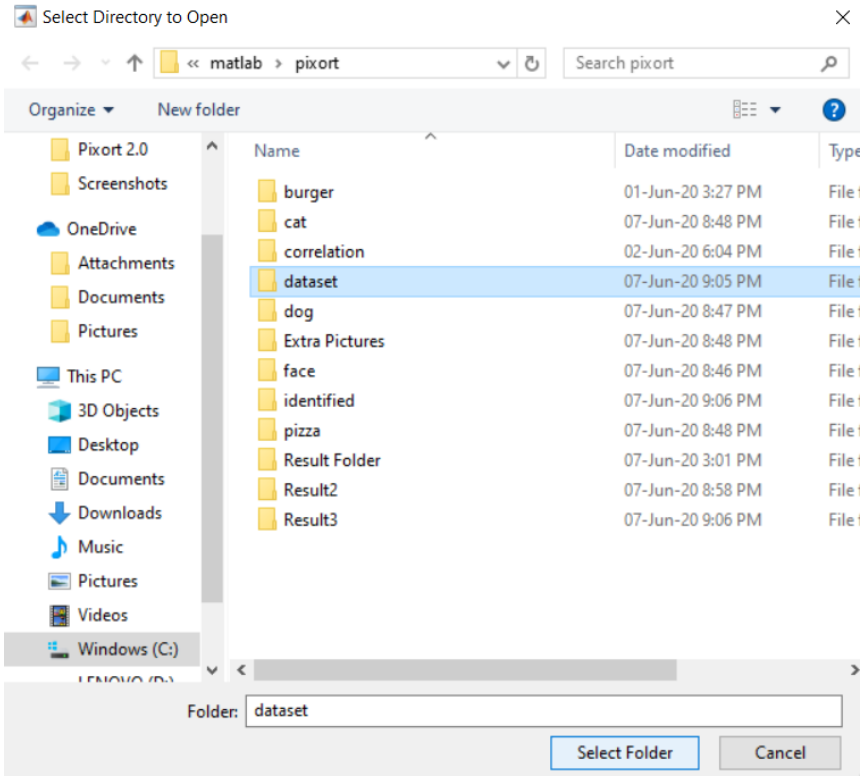

Figure 10. Selection of Gallery Folder

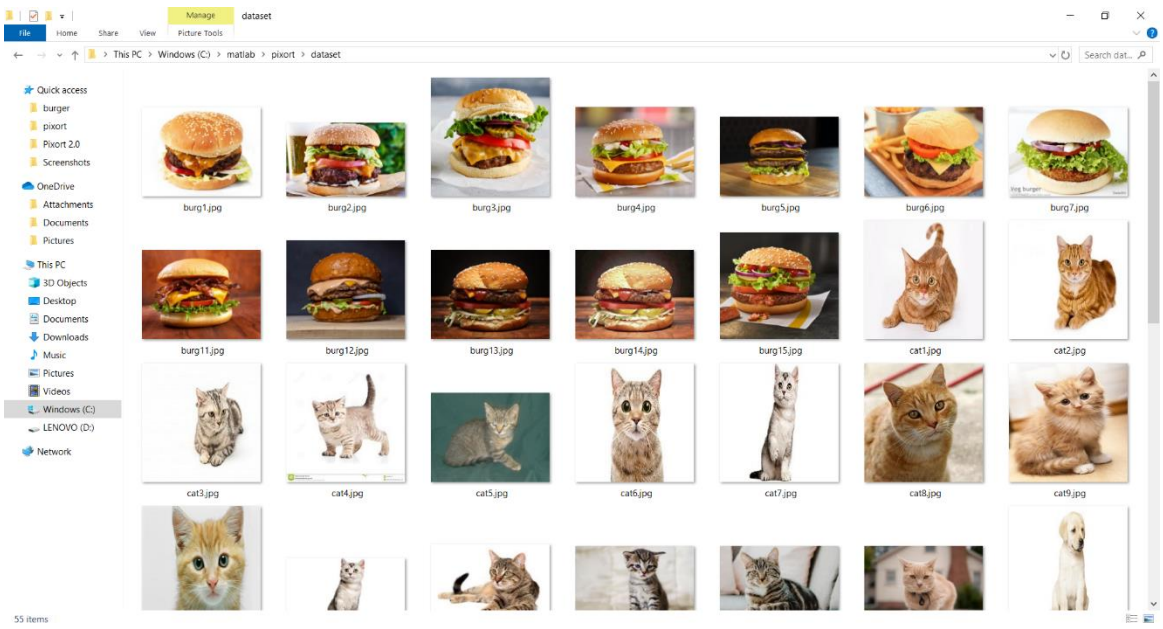

Figure 11. Images in the Gallery Folder

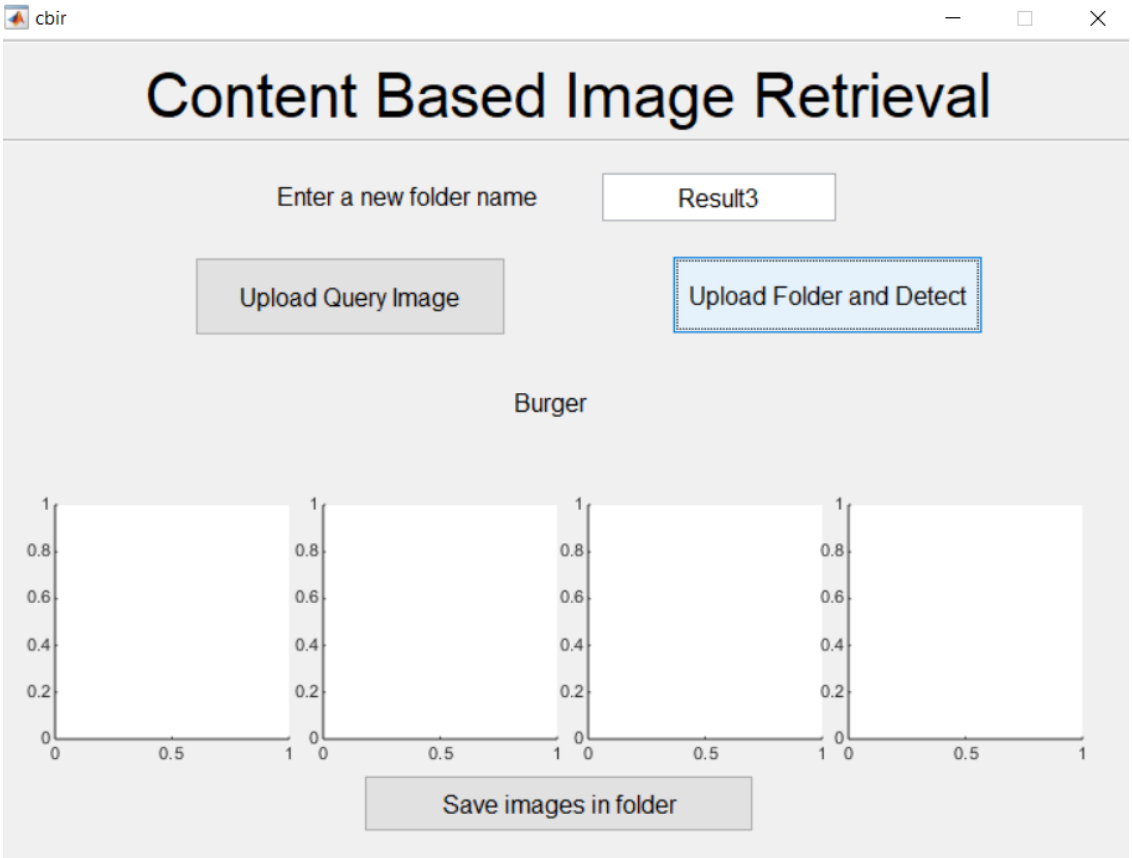

Figure 12. Identification of Content

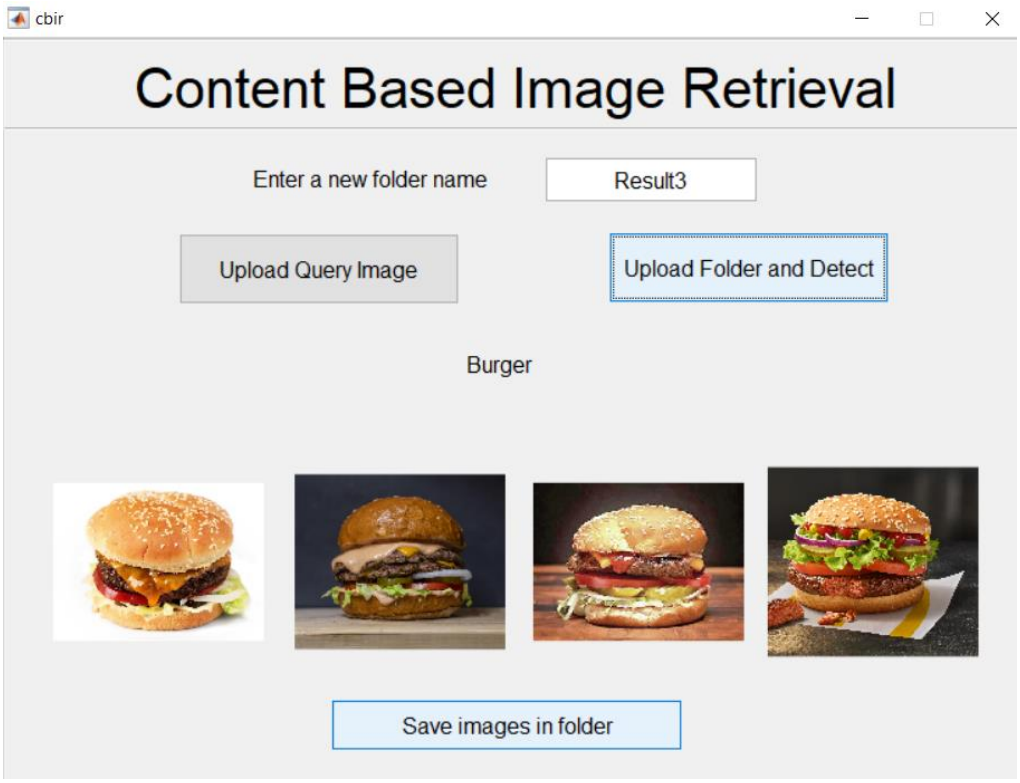

Figure 13. Retrieved Images

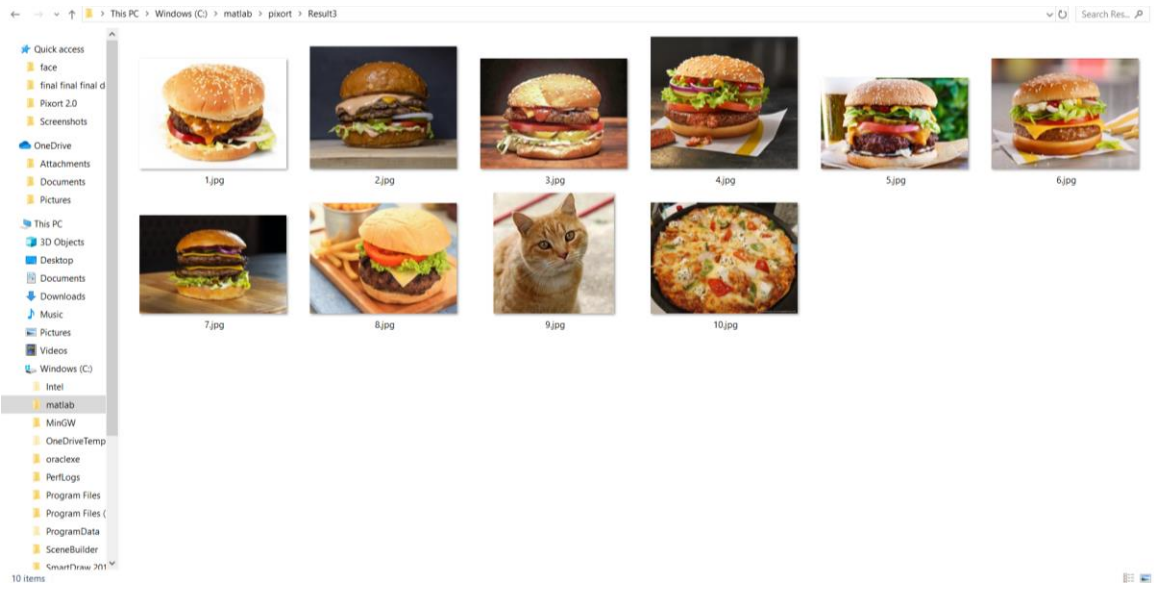

Figure 14. Images Saved in Result Folder

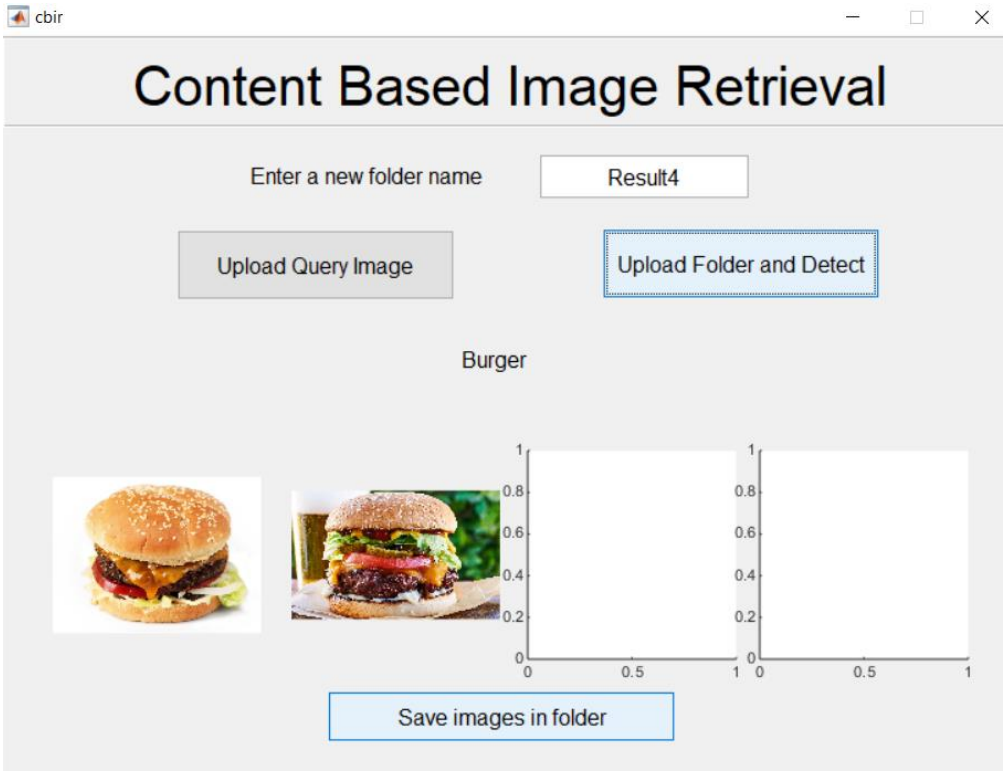

Figure 15. Minimum Four Images are not Retrieved

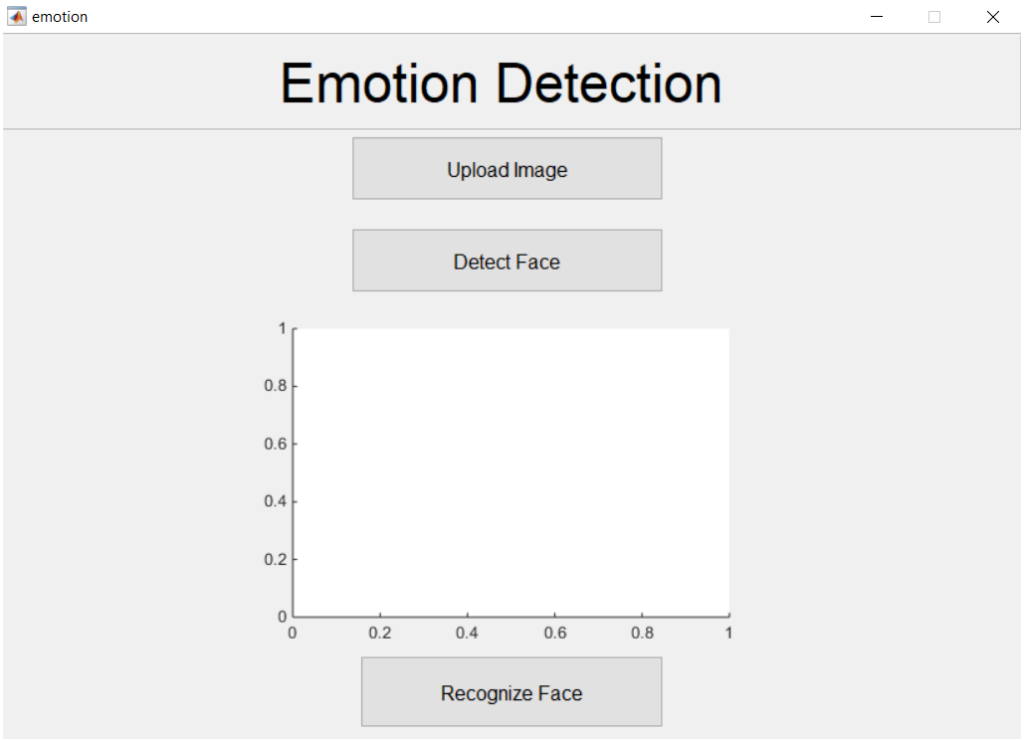

Figure 16. Emotion Detection (1) Interface

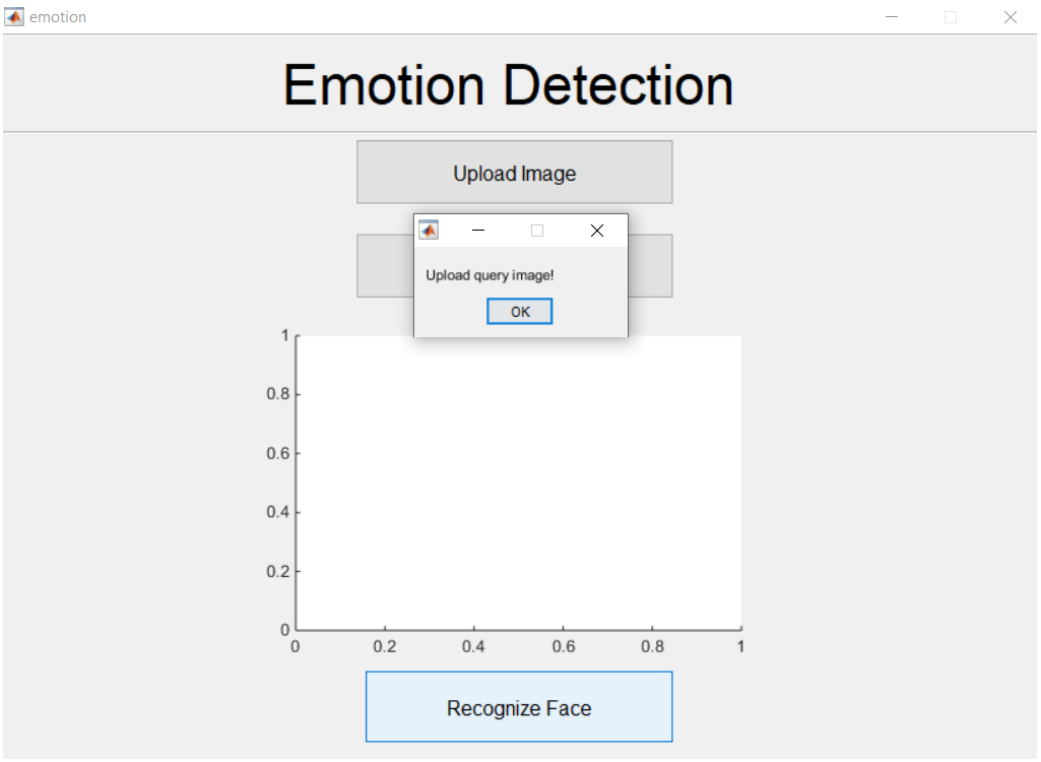

Figure 17. Recognize Face Button is Pressed without Uploading Query Image and Pressing Detect Face Button

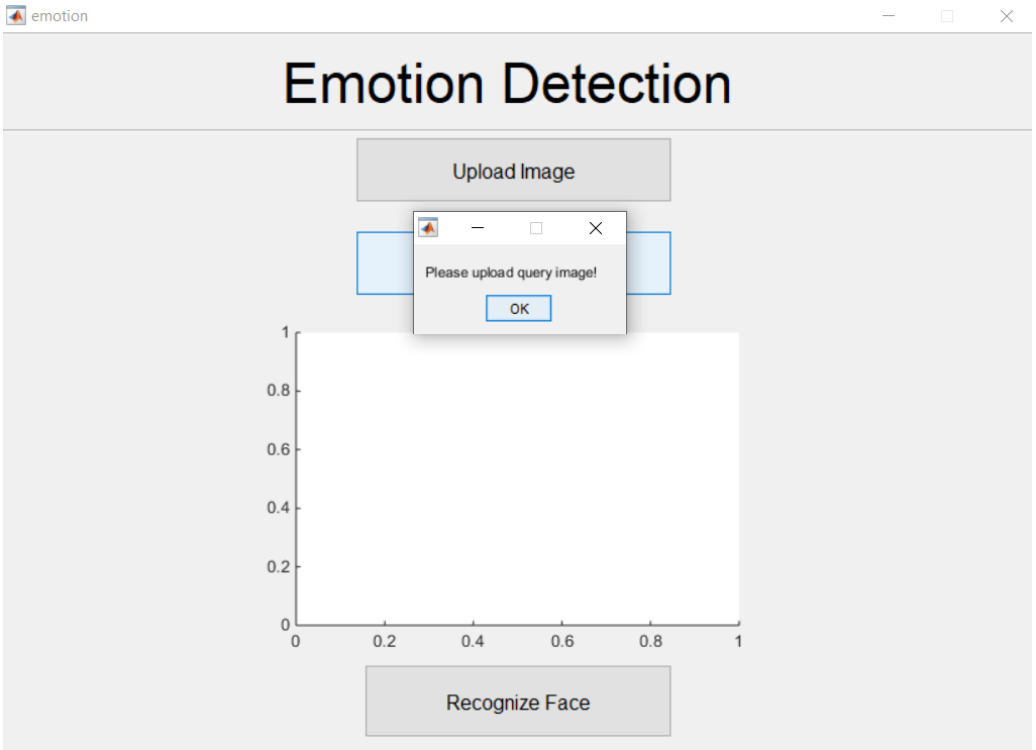

Figure 18. Detect Face Button is Pressed without Uploading Query Image

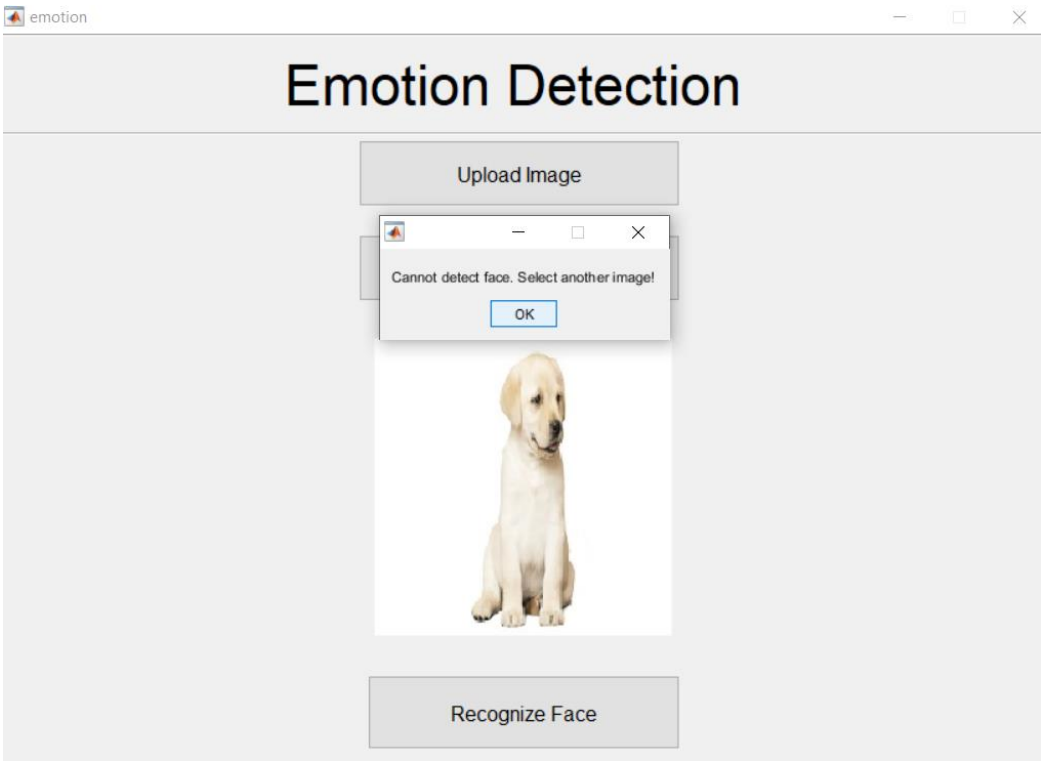

Figure 19. The Query Image does not contain a Face

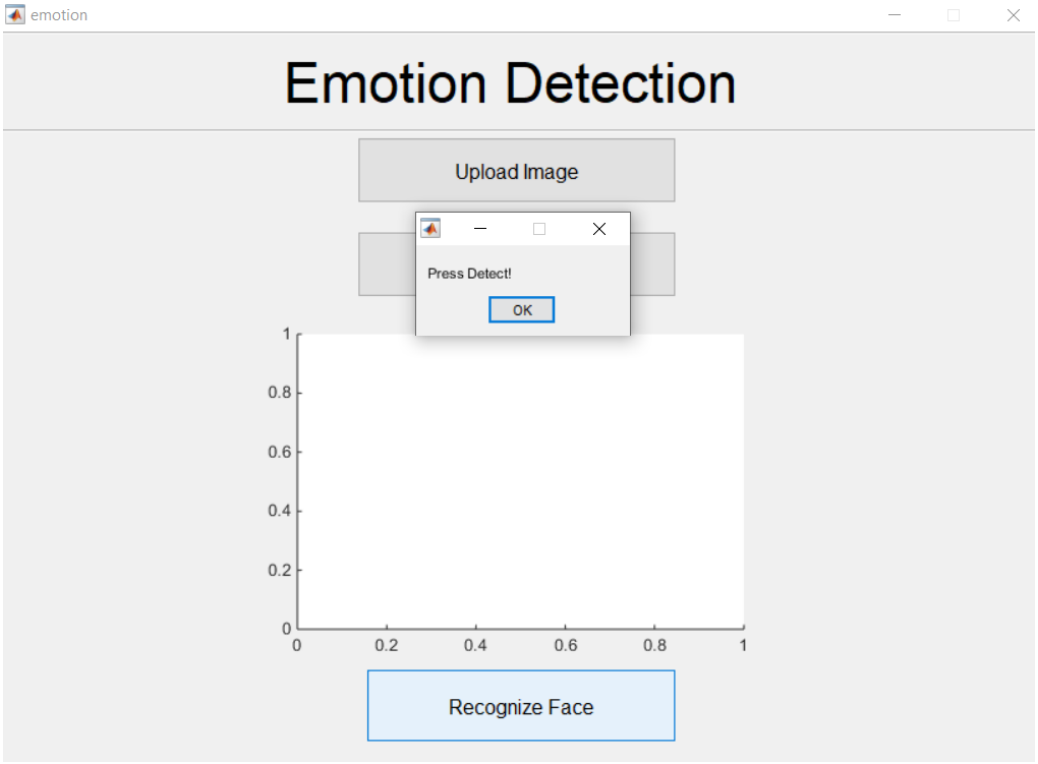

Figure 20. Recognize Face button is Pressed after Uploading Query Image and before Pressing Detect Face Button

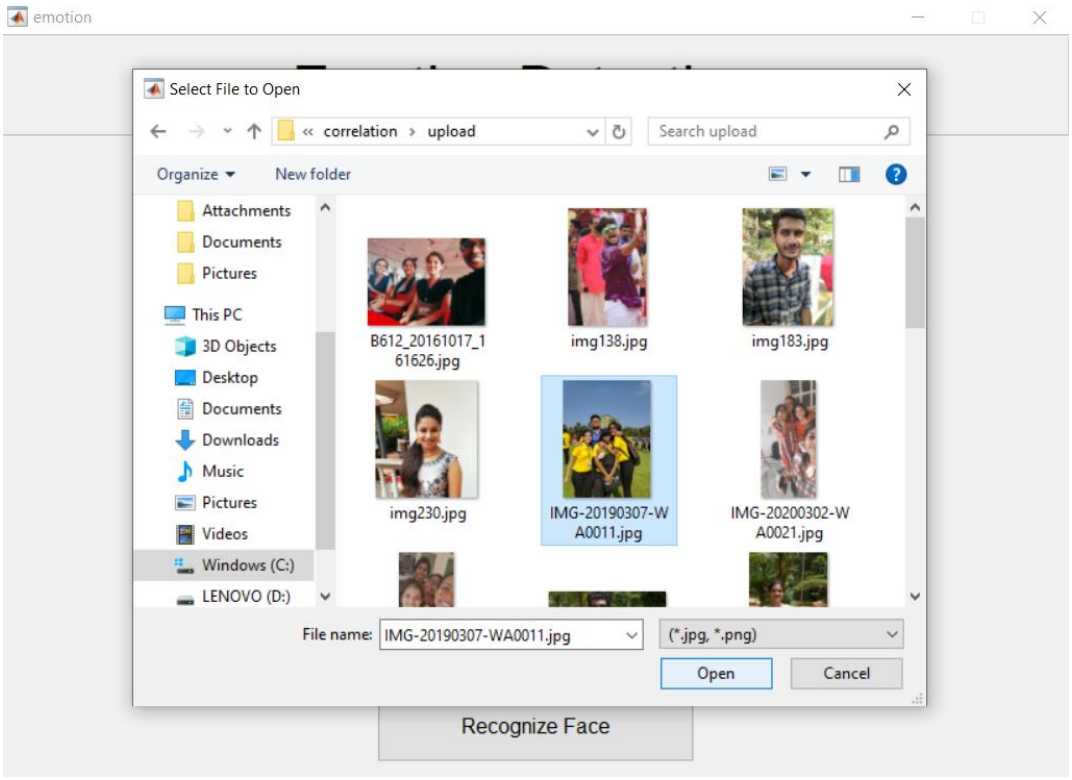

Figure 21. Uploading Query Image (1)

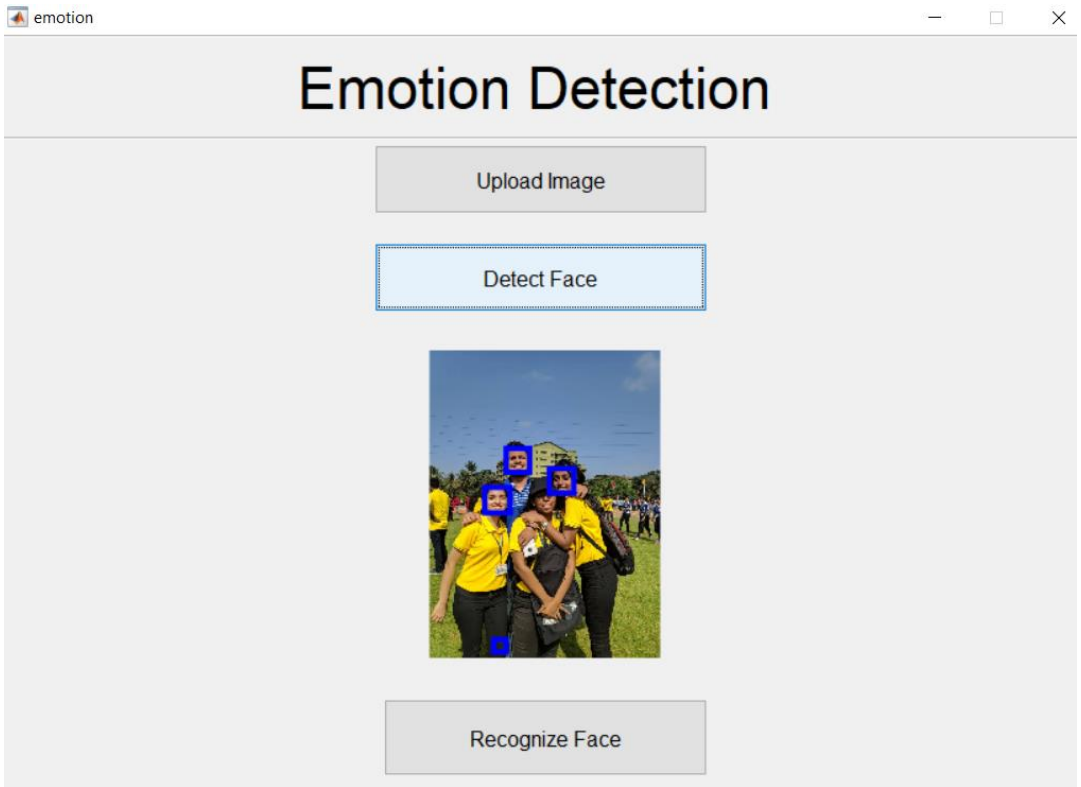

Figure 22. Detected Faces

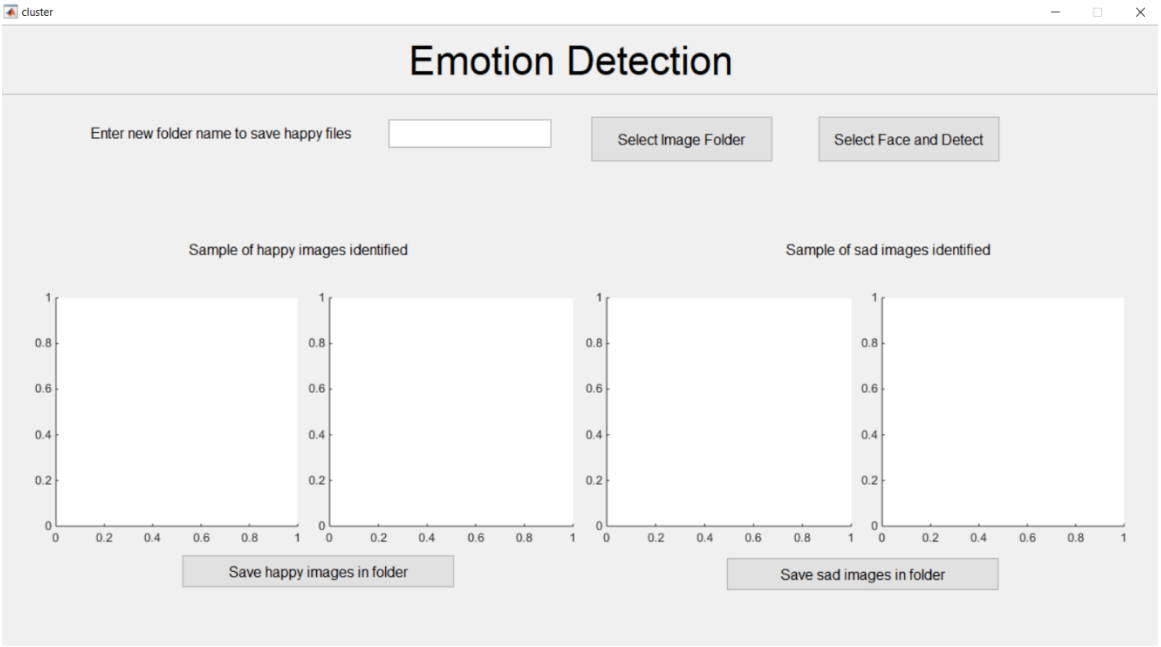

Figure 23. Emotion Detection (2) Interface

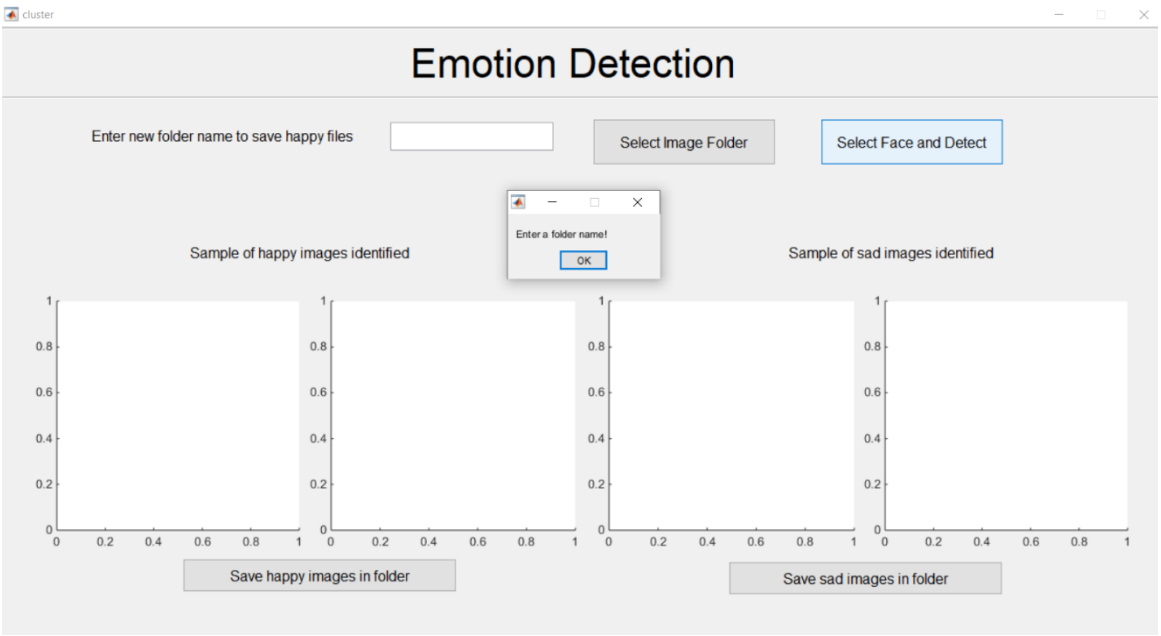

Figure 24. Pressing Select Face and Detect Button without Entering Folder Name

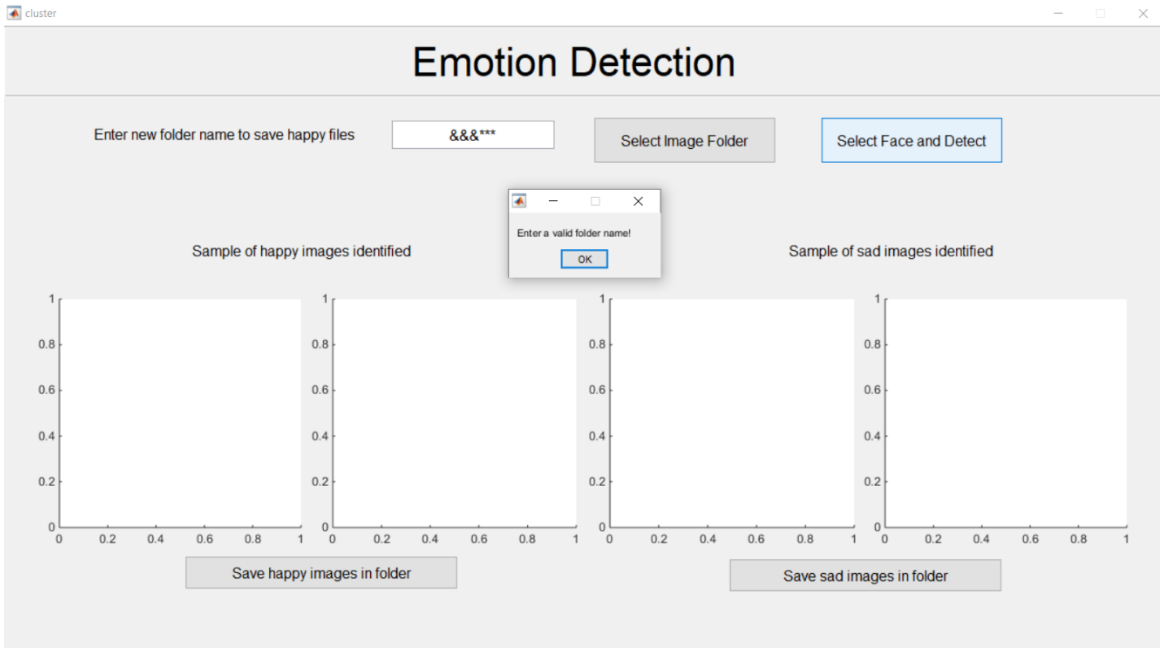

Figure 25. Invalid Folder Name is Entered

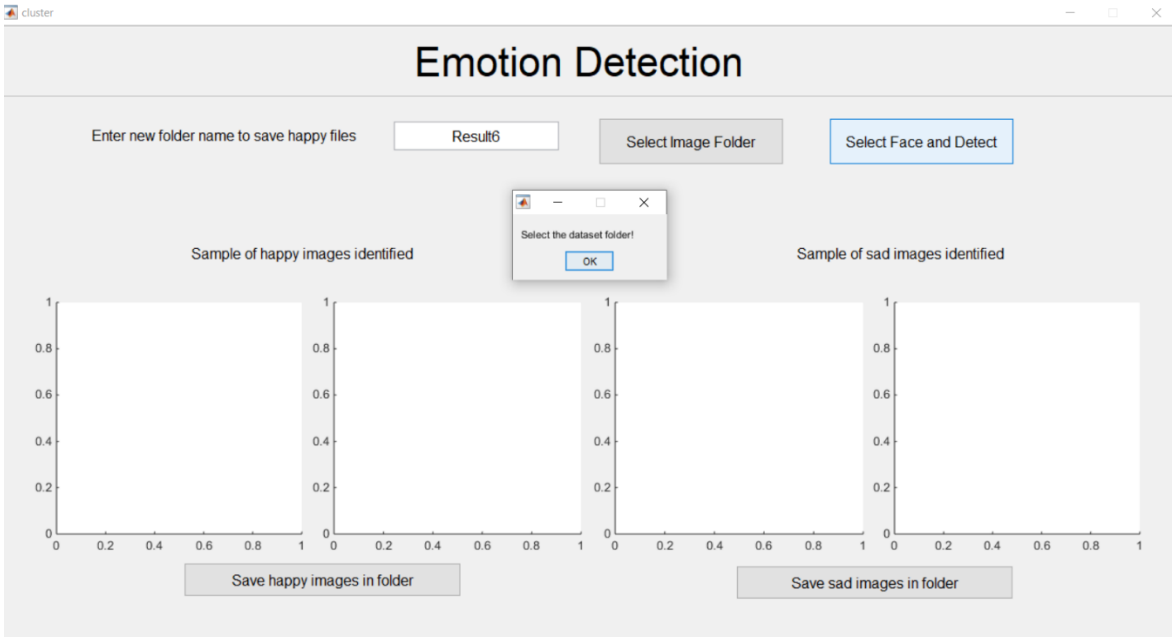

Figure 26. Pressing Select Face and Detect Button without Uploading Gallery Folder

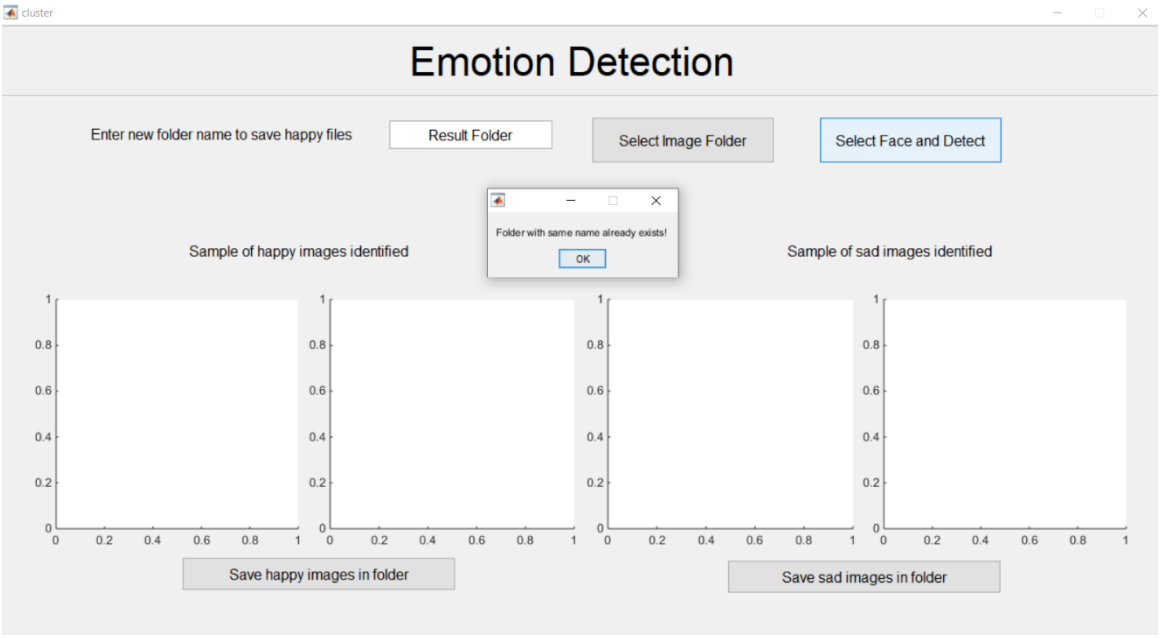

Figure 27. Existing Folder Name is Entered

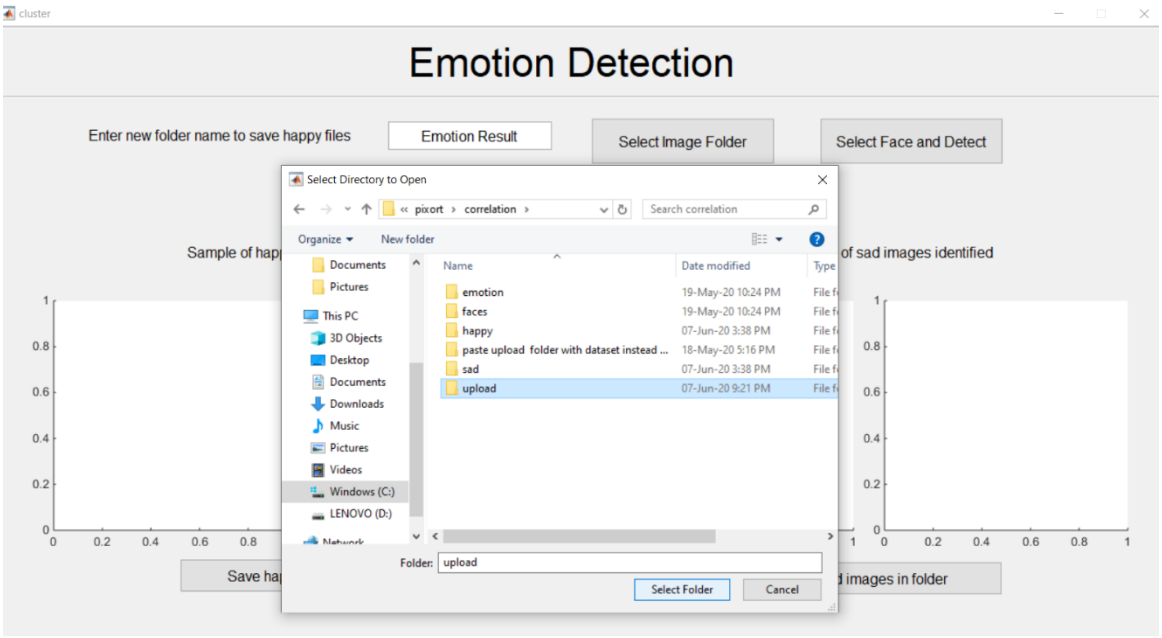

Figure 28. Pressing Select Face and Detect Button without Uploading Gallery Folder

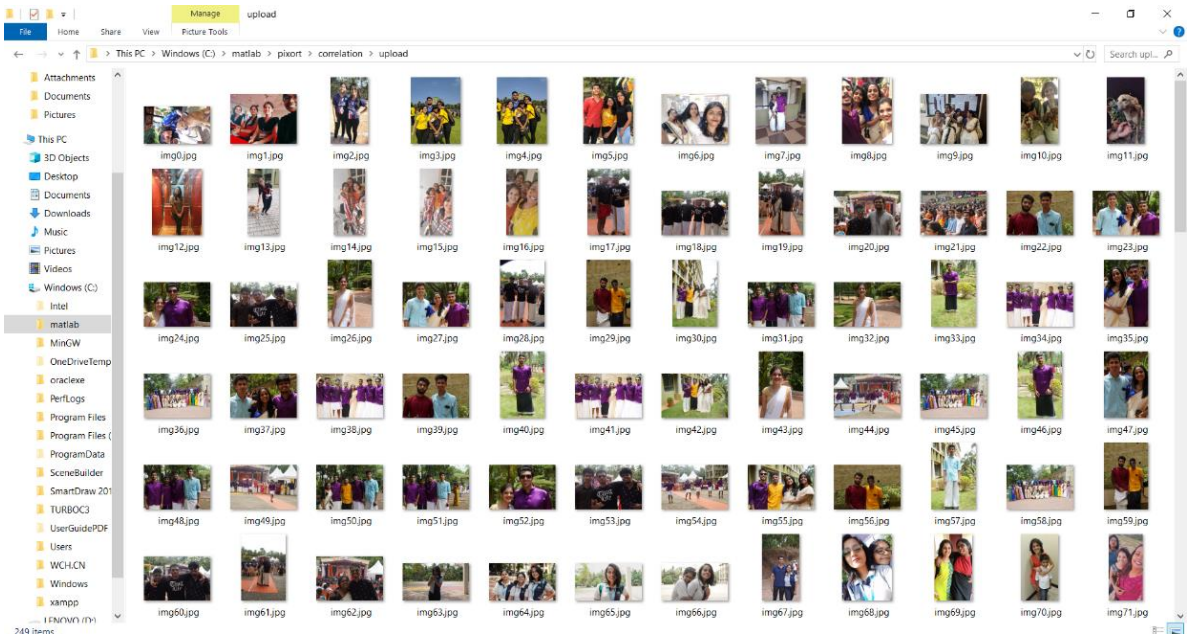

Figure 29. Images in Gallery Folder

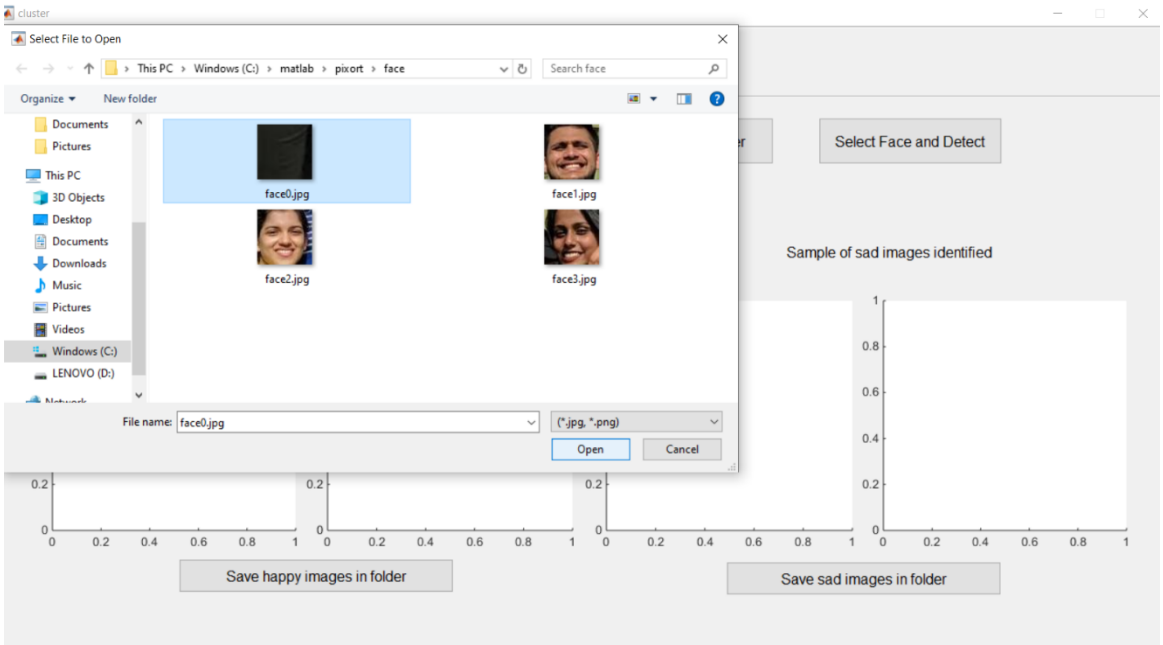

Figure 30. Selection of Invalid Face (1)

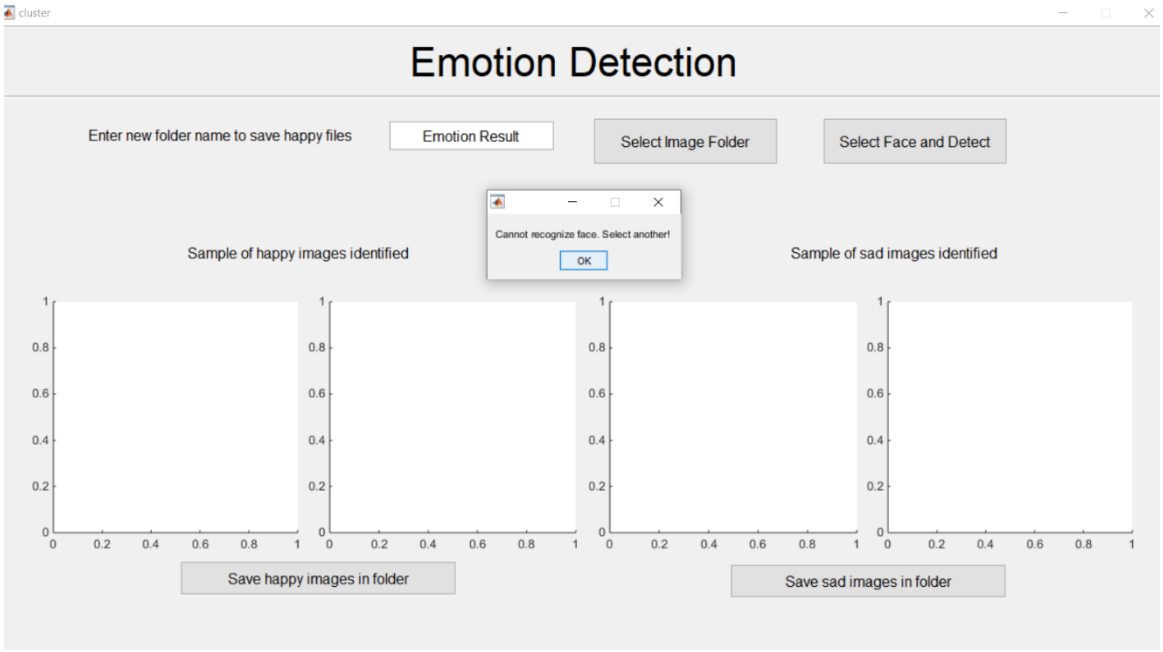

Figure 31. Selection of Invalid Face (2)

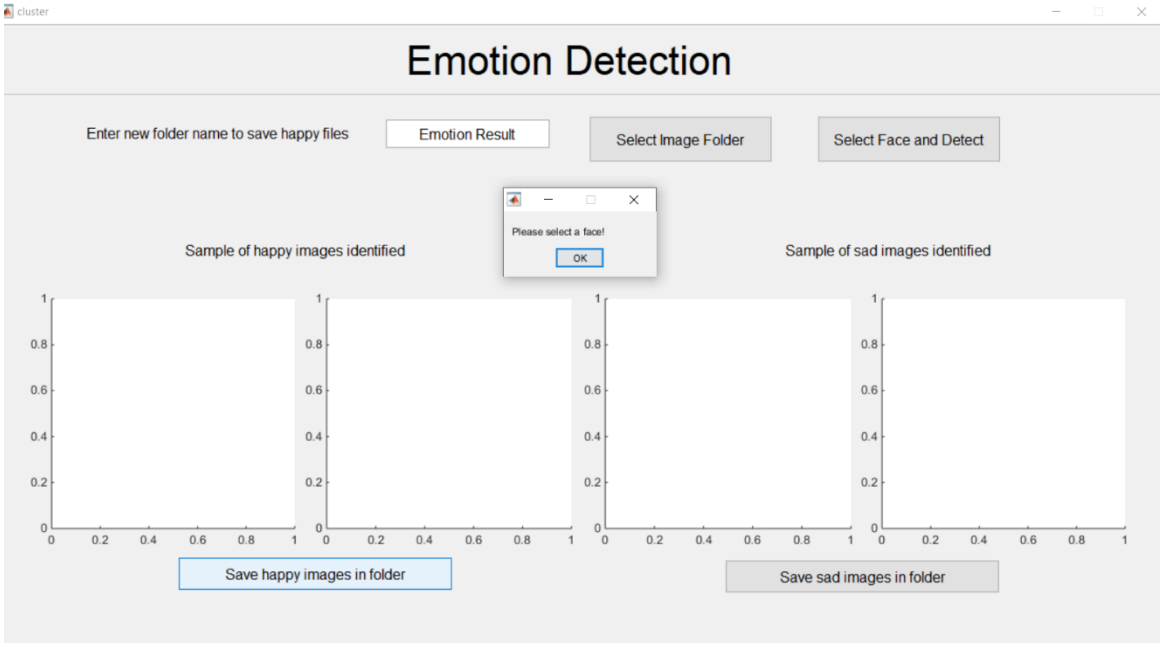

Figure 32. Pressing Save happy images in folder Button without Selecting a Face

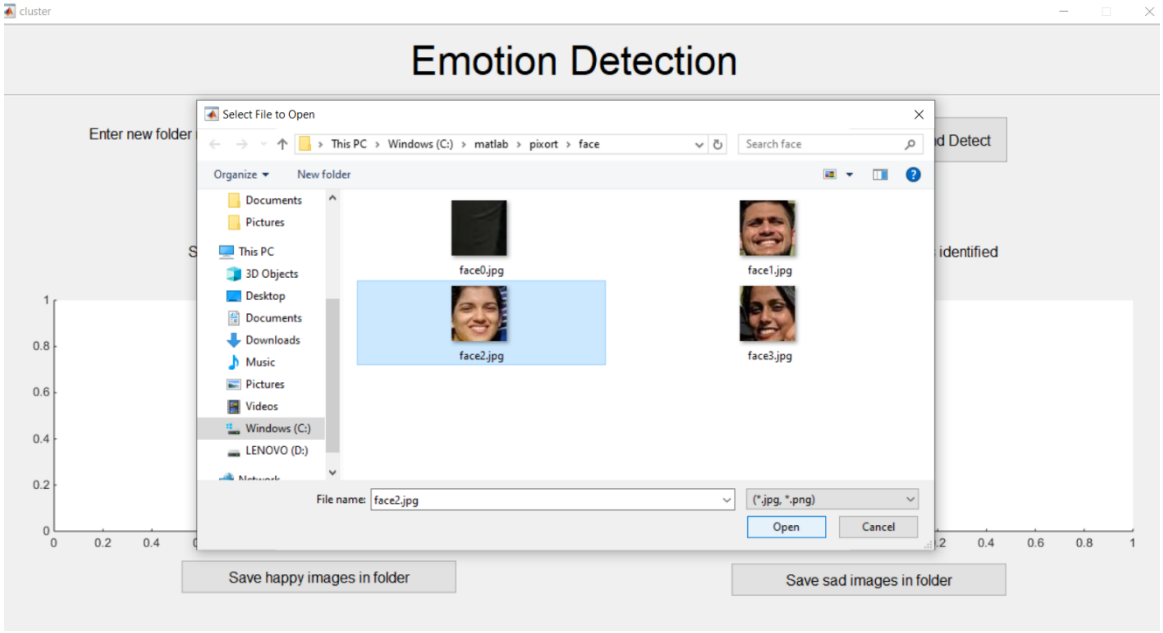

Figure 33. Selection of Face

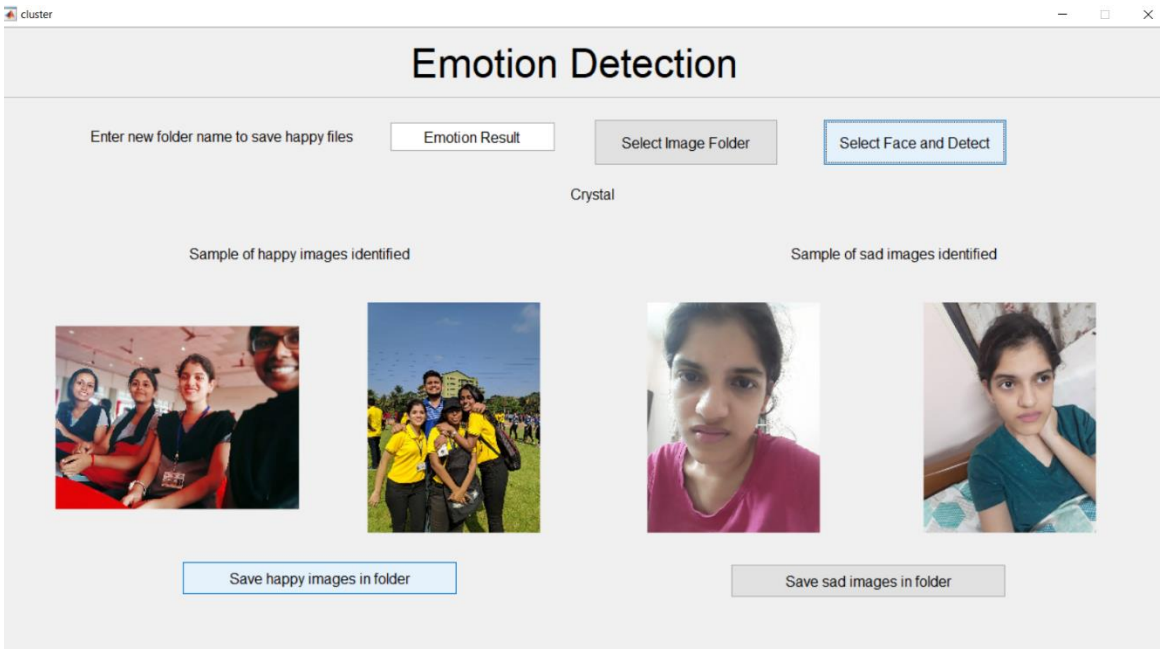

Figure 34. Retrieved Images

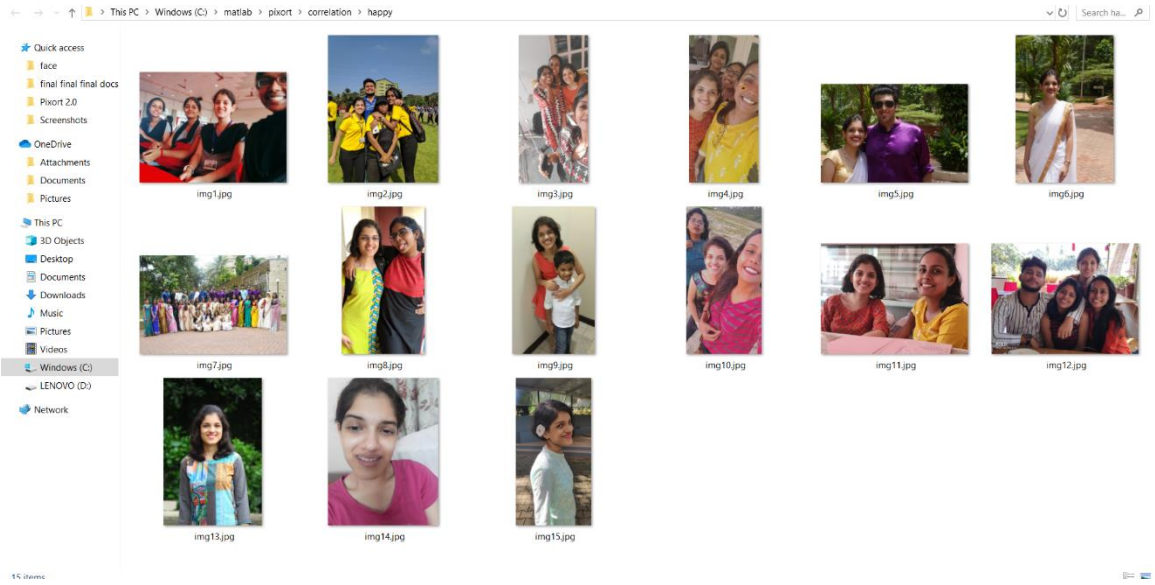

Figure 35. Images in Happy Folder

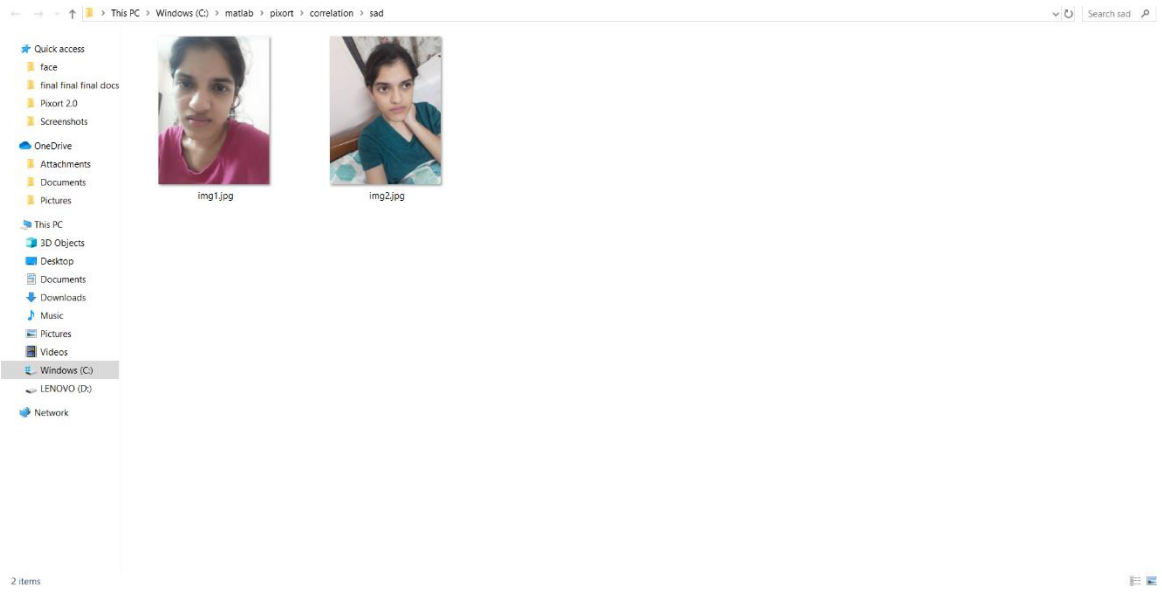

Figure 36. Images in Sad Folder

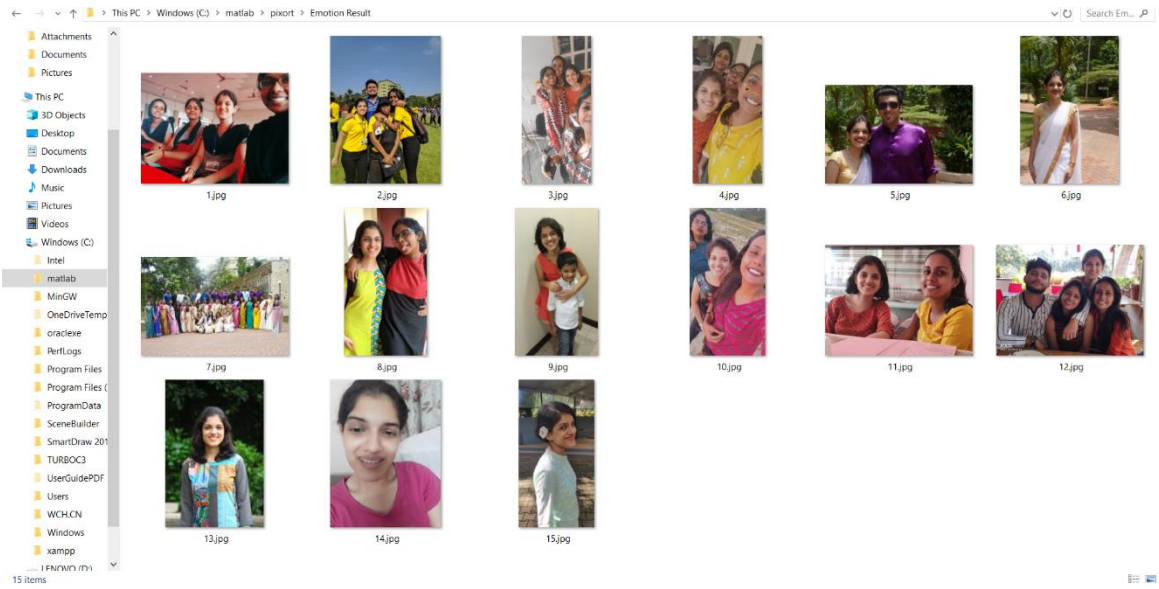

Figure 37. Images in Result Folder

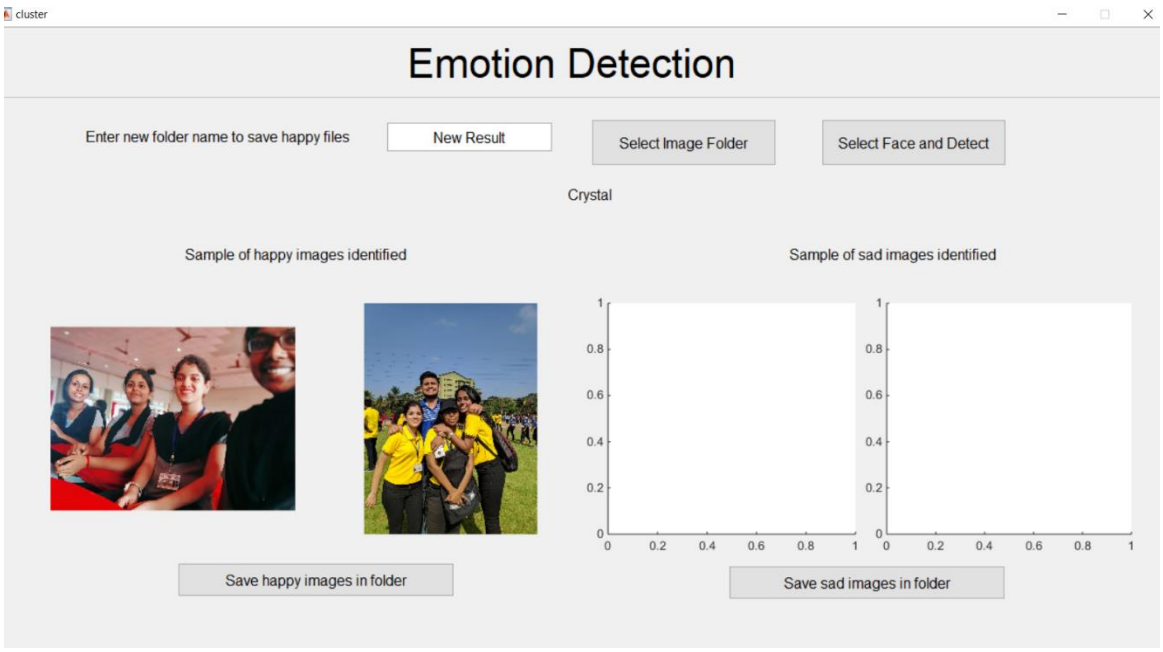

Figure 38. Minimum Two Happy/Sad Images not Fetched

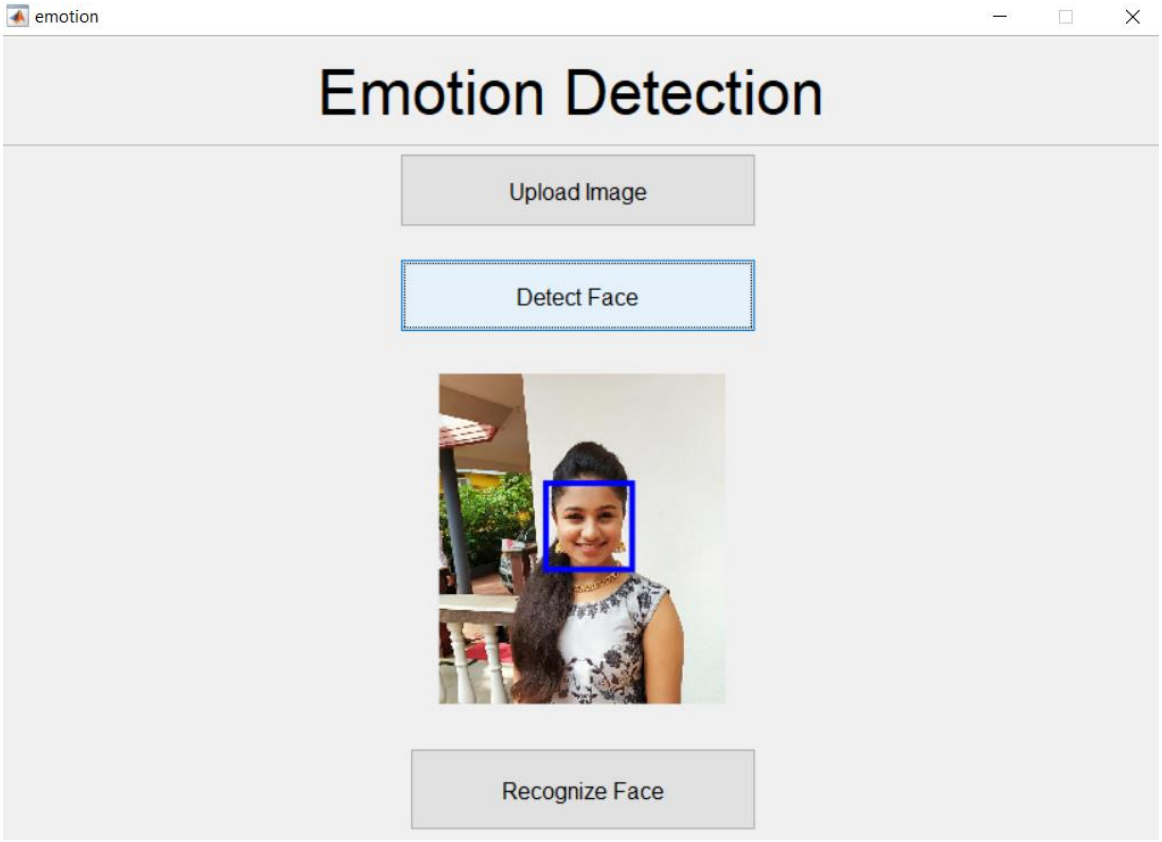

Figure 39. Uploading and Detecting Faces in Query Image (2)

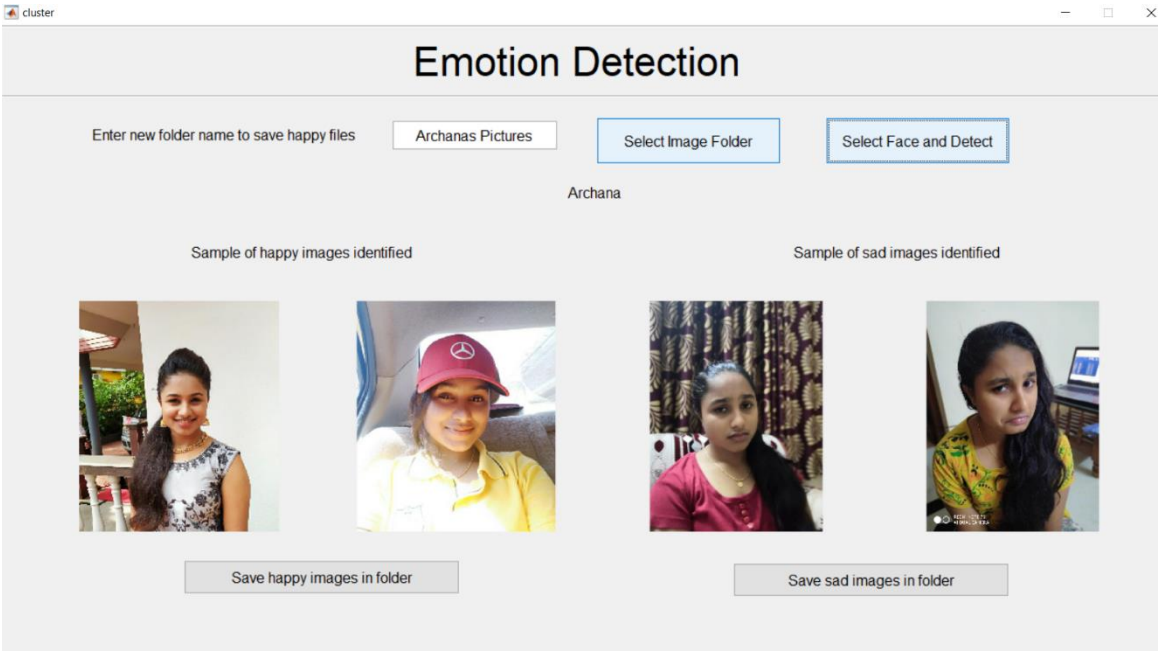

Figure 40. Retrieved Images

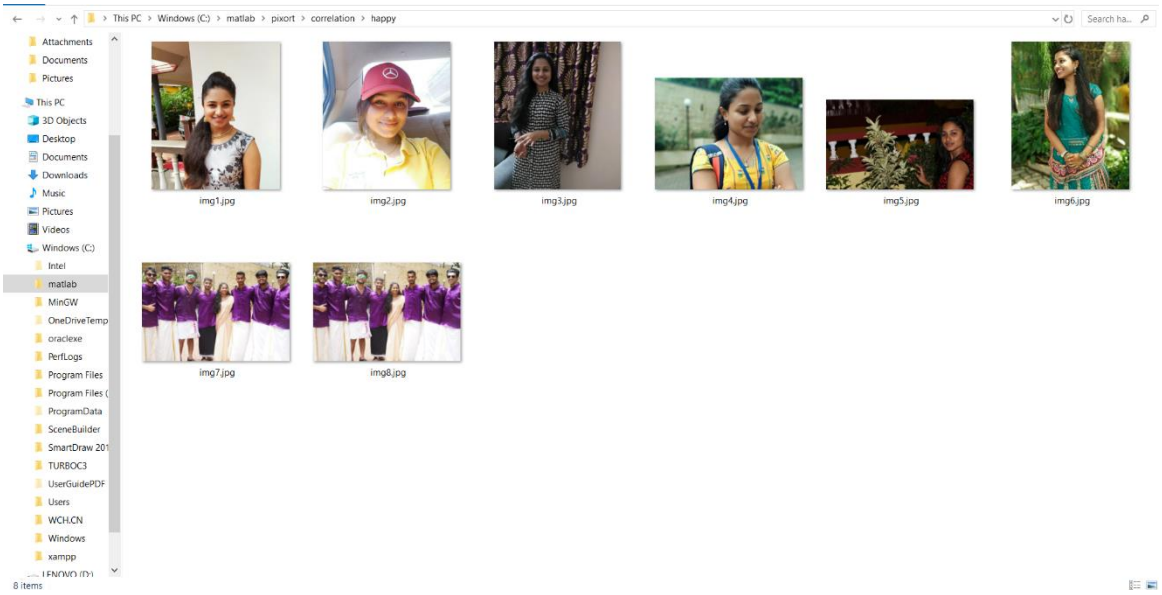

Figure 41. Images Saved in Happy Folder

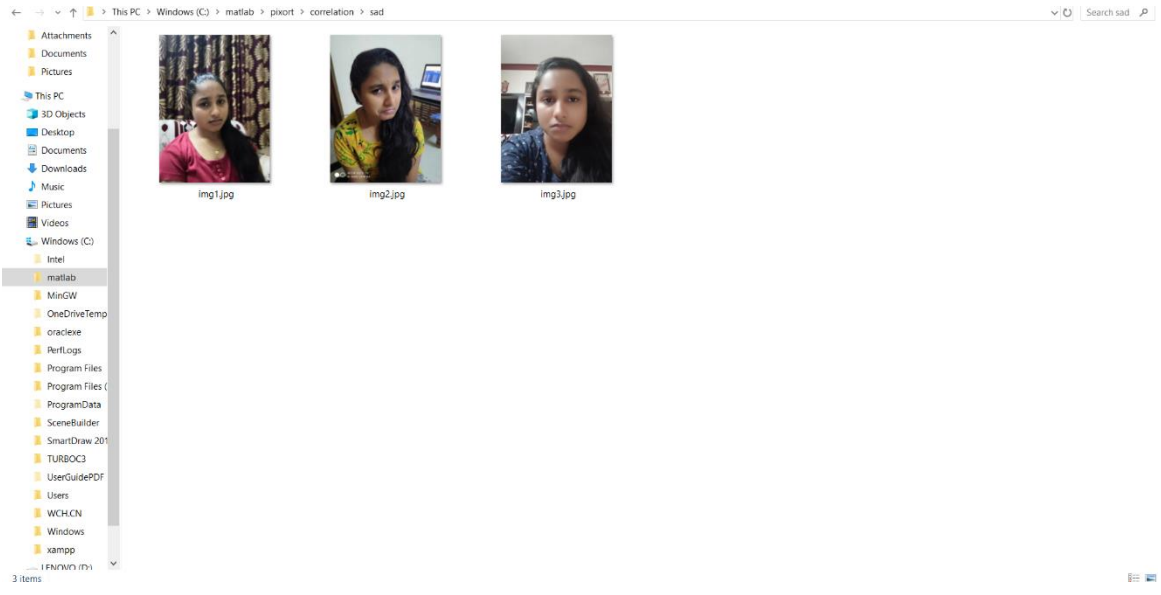

Figure 42. Images Saved in Sad Folder

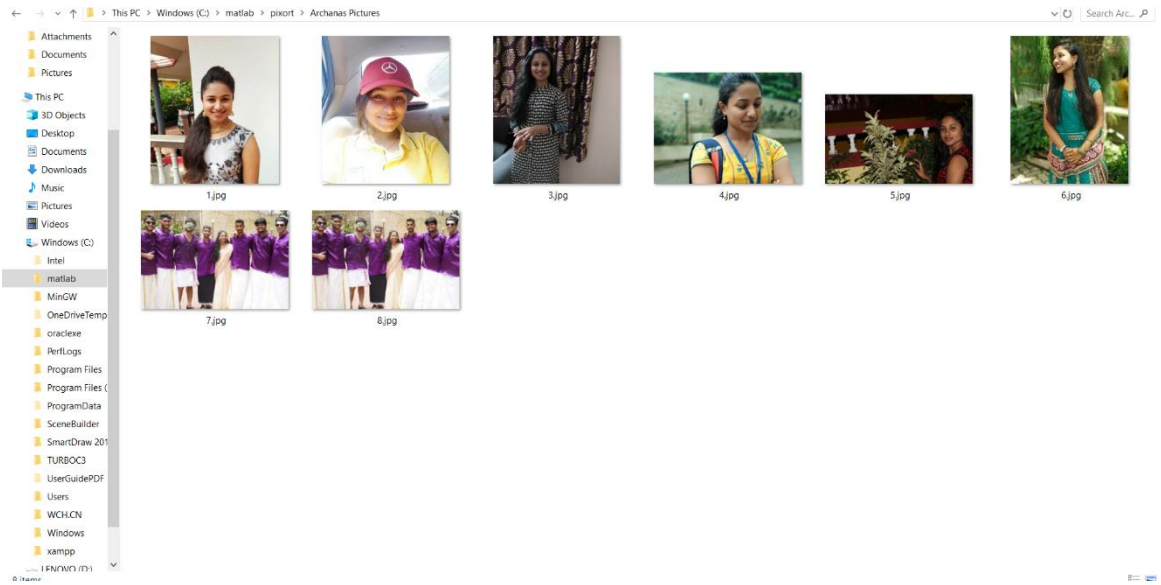

Figure 43. Images Saved in Result Folder