A PROJECT REPORT

ON

"IMAGE SORTING USING OBJECT DETECTION AND FACE RECOGNITION"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

SHAIKH REHAN	**16CO52**
SHAIKH ARBAZ	**16CO54**
SHAIKH SOHAIL	**16CO57**

UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2019-2020

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"IMAGE SORTING USING OBJECT DETECTION AND FACE RECOGNITION"

Submitted to
UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER ENGINEERING

BY

SHAIKH REHAN	16CO52
SHAIKH ARBAZ	16CO54
SHAIKH SOHAIL	16CO57

UNDER THE GUIDANCE OF
PROF. MUBASHIR KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2019-2020
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering

SCHOOL OF ENGINEERING & TECHNOLOGY

Plot No. 2 3, Sector - 16, Near Thana Naka,

Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"IMAGE SORTING USING OBJECT DETECTION AND FACE RECOGNITION"

submitted by

SHAIKH REHAN	**16CO52**
SHAIKH ARBAZ	**16CO54**
SHAIKH SOHAIL	**16CO57**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2019-2020, under our guidance.

Date: / /

(Prof. MUBASHIR KHAN)
Project Supervisor

(Prof. KALPANA BODKE)
Project Coordinator

(Prof. TABREZ KHAN)
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express our sincere thanks to our guide **Mubashir Khan**, Assistant Professor, Department of Computer Engineering, AIKTC, School of Engineering, Panvel for his invaluable support and guidance throughout our project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped us track and schedule the process effectively. His time, ideas and encouragement that he gave us help us to complete our project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. HOD NAME**, Head of Department of Computer Engineering and **Prof. MUBASHIR KHAN**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped us directly or indirectly during this course of work.

SHAIKH REHAN
SHAIKH ARBAZ
SHAIKH SOHAIL

Project I Approval for Bachelor of Engineering

This project entitled ***ÏMAGE SORTING USING OBJECT DETECTION AND FACE RECOGNITION¨*** by ***Shaikh Rehan, Shaikh Arbaz, Shaikh Sohail*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents our ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Shaikh Rehan
16CO52

Shaikh Arbaz
16CO54

Shaikh Sohail
16CO57

ABSTRACT

Title: Image sorting using Object detection, Face detection and recognition.

The requirement of the user is to quickly organize their images while doing little work. To achieve this we have implemented image sorting and grouping using three techniques; Object detection, face recognition and face detection. Face detection is acheived using Haar cascade classifier, whereas face recognition is done using LBPH algorithm. Object detection uses a deep learning method called as YOLO algorithm.

Keywords: Adaboost, Machine learning, Deep learning, Categorize, Object detection, Face detection, Face recognition, Convolution Neural Network(CNN).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image is one of the most used multimedia files. Since smartphones are developed cameras as well, everyone takes images and the trillions of images stored. Since these are not going to slow down, the digital image continues to grow. 1.2 trillion images are generated by the end of 2017. 4.7 trillion photos will be stored [**?**]. 182 photos per month are taken by the average iOS user, whereas 111 photos are being taken by average Android user [**?**]. Since there are bulk amounts of photos the average user does not bother to organize the images. It takes a considerable amount of time to organize these images.

This application helps you to quickly organize the unsorted image. The basic idea is to give images as input to the program, and objects or faces are detected in the images. The image will then be moved or copied to a folder to their respective class. This allows you to rapidly go through and organize your large amounts of pictures.

The input image is fed to the application, it will detect whether there is the face or not, if the face is detected in the image then face recognition is applied to the image. For face detection "Haar-Cascade algorithm" is used. After the face is recognized it will create a directory by the name of the given class label and for face recognition "Linear Binary Pattern Histogram" (LBPH) algorithm is used. If a face is not found then object detection will be used on the image. For object detection YOLO algorithm is used. If any object is detected in the image, it will be moved to the respective directory by the name of the class label.

Additionally, the search by face option allows users to search images with similar faces in the directory. All the images having the same face will be displayed.

1.1 Purpose

The purpose of this project is to make use of emerging new technologies of machine learning, deep learning to build an application that can help save people valuable time. The application would organize the images on a user's system in bulk. The images are sorted according to the visual content of the images. This operation happens locally and there is no requirement of the internet.

1.2 Project Scope

To implement object detection, face recognition and detection algorithms in the application. To properly categorize the images into appropriate classes and move them to their respective directories. To provide a search by image feature that uses the aforementioned algorithms to give a list of similar images without actually moving the images from their respective file locations. The application would provide the basic file managing capabilities of moving, deleting, and sharing of images from within the application.

1.3 Project Goals and Objectives

To find a way to reduce human burden by bulk sorting of images automatically using emerging technologies without much human intervention. To create an application which automatically sorts and categorizes images using deep learning and machine learning techniques for object detection, face detection and face recognition.

1.3.1 Goals

- To sort the images on the users' system based on user choice of directory.
- To sort the images based on the visual information of the images.
- To sort the images in bulk at once.
- To organize the sorted images in appropriate directories.
- To sort the images by faces and objects.

1.3.2 Objectives

- To make use of deep learning methods for sorting of the images.
- To make use GUI development tools to build a user-friendly interface.
- To make use of system features to organize the images in directories.
- To apply face and object detection and recognition algorithms to achieve object detection, face detection and recognition.
- To build am image organization application using the mentioned technologies.

1.4 Organization of Report

The project report starts with an Abstract and the associated keywords of the project. Then the following that are the table of contents, list of figures and list of tables. Following that the main report starts now with Chapter 1 being the 'Introduction' to the project. The introduction briefly introduces the idea behind the project, outlining the purpose of the project, project's goals and objectives. Then Chapter 2 is about 'Literature Survey'. Here, the technical research papers are studied and documented. Their brief abstract along with their advantages and disadvantages and techniques to overcome the disadvantages are given. Continuing in chapter 2, there is the technical review of the technologies used in the project, and the advantages of technologies and the reasons to use them are outlined. The next chapter that is Chapter 3 is titled as 'Project Planning'. This chapter is all about project management. All the information regarding the members of the group such as their names and roles are mentioned here. Information regarding, assumptions and constraints, project management approach, ground rules, project budget and timeline are discussed in this chapter. Chapter 4 is titled 'Software Requirements Specification'. It details all the project requirements. It gives the overall description of the product. It then details the system features. It details external interface requirements such as hardware, software, communication and user interfaces' requirements. And finally, at the end of the chapter are the details of non-functional requirements. Chapter 5 'System Design' begins with system requirements definition specifying functional and non-functional requirements. All the appropriate diagrams such as Use-case diagram, system architecture diagram, etc are given here. Then, details of all the modules of the system are given. Chapter 6 'Implementation' is where the code of the program is written. Chapter 7 is about testing the product. Chapter 8 has the screenshots of the working product. Chapter 9 has Conclusion of the report and the future scope of the project. References and Achievements finish the report.

Chapter 2

Literature Survey

2.1 Object Recognition in Images

Object Recognition is a field of study in image processing. The process of identifying objects in video or an image is termed as Object Recognition. It has a huge number of applications in the field of activity recognition, robot localization, and automation, etc. Objects appear different when seen from a different perspective. It should be invariant to changed viewpoints, robustness, occlusion and object transformations. This task targets to perform a technique including mainly 2 stages. In the first stage, the input image is categorized using a classifier. In this paper [1] two type of classifiers are used for classifier optimization which are "k-nearest neighbour(kNN)" and "Support Vector Machine(SVM) classifier". SVM classifier uses GIST features and the kNN classifier uses SIFT features. Various kernels are used in SVM such as Gaussian, Linear, and Polynomial. Feature extraction takes place, forming a similarity matrix. It is given to the kNN classifier. The comparison shows that the SVM classifier is more accurate than the kNN classifier. Coil-20P and Eth80 are the datasets used for the processing.

2.1.1 Disadvantages of Paper

a. Hand engineering of features is required before object recognition is applied.

b. Unable to connect Designer and Customers.

c. Collaboration is not possible.

d. Cannot be viewed from Different angles.

2.1.2 How to overcome the problems mentioned in Paper

a. Use feature extraction techniques such as convolutional neural networks.

2.2 The Object Detection Based on Deep Learning

The paper [3] tells about the emergence of the object detection based on deep learning. It reviews the classical method of object detection. The paper[3] states that in deep learning methods the region selection can be achieved using particular strategies, the feature extraction can be done with the help of CNN and the classification achieved by using SVM or a special neural network. The paper reviewed two methods of deep learning namely DNN and Overfeat.

2.2.1 Disadvantages of Paper

a. Poor real time because of large number of network parameters.

b. Hard to train

2.3 Object Detection Using Convolutional Neural Networks

In this paper [5] ”Convolutional Neural Network (CNN)” is used for object detection. It uses an activation function called Rectified Linear Unit. It uses transfer learning which is a powerful deep learning technique in which pre-trained models can be used for feature extraction. It uses the Tensorflow library for high-performance numerical computation. In this paper [5] two models are compared that are ”Single Shot Multi-Box Detector” and ”Faster Region-based Convolutional Neural Network”. SSD with MobileNetv1 is used in object detection because it is lightweight, accurate and small in size and can, therefore, be used in mobile devices.

2.3.1 Disadvantages of Paper

a. SSD is less accurate

b. Faster R-CNN has low speed

2.3.2 How to overcome the problems mentioned in Paper

a. Use SSD when accuracy isn’t the top priority but speed is, as in real-time detection systems.

b. Use Faster R-CNN where high accuracy is required over speed such as in medical imaging.

2.4 Application of deep learning in object detection

This paper [2] deals with the application of deep learning in object detection. It gives a summary of some commonly used datasets such as ImageNet, PASCAL VOC, COCO. And create a new dataset for a football game. It gives a summary of the series of algorithm based on R-CNN such as R-CNN, SPP-Net, Fast R-CNN, Faster R-CNN. Faster R-CNN is used because the mAP for Faster R-CNN is 0.732 on the VOC2001 dataset. Using Faster R-CNN on a newly created dataset the mAP for 4 classes is player 0.7902, soccer goal 0.8377, corner flag 0.3508, football 0.4752.

2.4.1 Disadvantages of Paper

a. Dataset is of uneven quantity.

b. Dataset size is uneven.

2.4.2 How to overcome the problems mentioned in Paper

a. Use the right evaluation metrics.

b. Resample the training set.

   (a) Under-sampling.

   (b) Over-sampling.

2.5 Research on face recognition based on deep learning

In this era of digitization, new technologies are flourished, deep learning is advancing in various areas. Deep Learning is the sub-field of artificial neural network (ANN), it consists of different techniques/algorithms encouraged by the construction of the human brain. Handwriting recognition, image recognition, semantic analysis, weather forecasting, marketing predictions, etc uses deep learning. In the given paper [4], it mainly focuses on the complexity in deep learning regarding face recognition and their solutions to improve in the results and accuracy. Deep learning methods, its applicable knowledge along with face recognition for further delve.

2.5.1 Disadvantages of Paper

a. How to analyze and review input when understanding the input character after learning is the optimum result?

b. How to further enrich the DB resource?

2.6 Technical Review

The technologies used in the project are geared towards machine learning and GUI development. For the base of the project the main and only programming language used is Python. Python is an interpreted, high-level and general-purpose programming language. Machine learning and alike are the specializations of the Python language. Python is also used in the GUI development through the use of PyQt, a python binding for Qt application framework. OpenCV is an open source computer vision library. Using opencv, face detection algorithm Haar Cascade, object detection algorithm YOLO and face recognition algorithm LBPH are implemented.

2.6.1 Advantages of Technology

a. Python is the obvious choice of language for deep learning tasks as it has all the functionalities and tools required for it.

b. PyQt5 uses the same language as the core language, making integration painless.

c. OpenCV is an easy to use, open sourced library which saves a lot of time.

d. YOLO algorithm can be pre-trained with significant amount of data before deployment of the product.

2.6.2 Reasons to use this Technology

a. Python is the best suited language for this project.

b. PyQt5 is used so that the codebase can be uniform, allowing seamless system integration.

c. OpenCV is a community driven, open source library with all the required methods for this project.

d. YOLO algorithm's accuracy can be increased by training it with the appropriate dataset.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Shaikh Arbaz	Database, UI Design
2	Shaikh Rehan	UI design, Database
3	Shaikh Sohail	UI design

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Shaikh Arbaz	Team Leader	Face detection and recognition
2	Shaikh Rehan	Programmer	UI programmer and file system
2	Shaikh Sohail	Programmer	Object Detection

3.3 Assumptions and Constraints

1. The user wants to run the product locally.
2. The user needs to organize their images based on the visual information in them.
3. The user wants to organize the images keeping the directory structure intact.
4. The user is willing to allow the product to move images from their original location to their new folder(s).

3.4 Project Management Approach

The project will start with literature survey of technical papers, followed by doing a review of an already existing, similar product. This gives us the requirements to proceed with the design of the project. We design the project using UML. The members would then start working on the coding of the project module by module. One member would do the GUI development, the others would do research and coding of the actual program. The code changes would be handled using Git versioning system. A remote repository would be maintained on GitHub. Then the testing of the use cases would be done. And the final product would be ready. A typical waterfall model of SDLC.

3.5 Ground Rules for the Project

- The members would have their own machine to code on.
- Each member would have copies of the code and designs.
- The code would be maintained on a remote repository on GitHub.
- Any changes made would be committed to this repository.

3.6 Project Budget

- This project was built using already owned computer systems.
- The project is entirely a software product, so no additional hardware cost.
- The budget for the project was hence, zero.

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The product is an entirely new, self-contained project built from scratch. This is not a part of a larger system neither this project is a follow-on member of a product family.

4.1.2 Product Features

The major features of this product is object detection, face detection and recognition. Using these three technologies images are sorted on a user's machine. The sorted images are then organized by moving the appropriate images to their respective directories. Users can also search by an image and the matching images would be displayed to them.

4.1.3 User Classes and Characteristics

This Software is intended for any user who owns a computer and wants to organize the image on there compute locally. It does not required any kind of expertise to use this software.

4.1.4 Operating Environment

Software requirements

- Python v3.6 for deep learning and machine learning algorithms.
- PyQt framework for GUI development.
- OpenCV and DarkNet computer vision libraries.
- MS COCO image dataset for deep learning and machine learning algorithms.

Hardware Requirements

- Intel Core i5 quad core @ 2.4 GHz or AMD equivalent x86 CPU.
- 8GB DDR3 RAM.
- 10GB free hard drive space.

4.1.5 Design and Implementation Constraints

The product has no corporate or regulatory policies that need to be followed. The technologies used can seamlessly work together as all the code is written in Python, be it for GUI development or otherwise.

4.2 System Features

The major features of this product is object detection, face detection and recognition. Using these three technologies images are sorted on a user's machine. The sorted images are then organized by moving the appropriate images to their respective directories. Users can also search by an image and the matching images would be displayed to them.

4.2.1 Object Detection

Object detection is used to detect objects in the image. It uses YOLO Algorithm to detect object and for datasets it uses MS COCO datasets which has various classes.

Description and Priority

Object detection is one of the major features of this system which is used to detect object in the image and its of High priority.

Stimulus/Response Sequences

Stimulus: User select the folder in which the image resides.
Response: The folder is selected.
Stimulus: User will double click on the folder.
Response: The image will be moved to the new folder created by the object name if there is no faces detected in the image.

Functional Requirements

REQ-1: Object must be identified in the image in order to move the image.

4.2.2 Face Detection

Face detection is used to detect faces in the image. It uses Haar cascade classifier to detect faces in the image.

Description and Priority

Face detection is one of the major features of this system which is used to detect faces in the image and its of High priority.

Stimulus/Response Sequences

Stimulus: User select the folder in which the image resides.
Response: The folder is selected.
Stimulus: User will double click on the folder.
Response: If face is detected in the image then face recognition module will be called to recognized the face.

Functional Requirements

REQ-1: Face must be detected in the image.

4.2.3 Face Recognition

Face recognition is used to recognized face in the image. It uses LBPH Algorithm to recognize face.

Description and Priority

Face recognition is one of the major features of this system which is used to recognize face in the image and its of High priority.

Stimulus/Response Sequences

Stimulus: User select the folder in which the image resides.
Response: The folder is selected.
Stimulus: User will double click on the folder.
Response: If face is detected in the image then face recognition module will be called to recognized the face.
Stimulus: After face is detected face recognition module will identified the face.
Response: The image will be moved to the new folder created by the person name.

Functional Requirements

REQ-1: Face must be detected in the image.
REQ-2: Face data must be provided.

4.3 External Interface Requirements

4.3.1 User Interfaces

The user interface shows the user their file system on their machine in a window. The user is able to browse the directories by single clicking the folder or by clicking the expand arrow. By double clicking on a folder, all the images on the clicked folder would be scanned and the sorting and organization of images would take place. The results would be displayed in the console.

4.3.2 Hardware Interfaces

The product can be run on an x86 architecture based system. The user can navigate and use the product using just the mouse as the pointing device and a monitor to run and display the program window. The user will need the mouse to point and click/double click the elements on the monitor display. The output would be displayed on the monitor display.

4.3.3 Software Interfaces

The operating system could be either Windows or Linux based. The Graphical User Interface (GUI) would be connected to the backend. The user's input from the GUI would be fed to the backend program, the backend would then show the progress of the operations through the console. The backend would scan the user's system and get all the files and directories. Then it would identify the directories that contain only images. The structure is then displayed using the GUI. The Gui would pass selected directory to be scanned back to the backend. The backend would then process the information and the result is displayed on the console.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

The product is for processing and handle images in bulk. So the performance should be good enough to do that in a timely manner without it affecting system performance.

4.4.2 Safety Requirements

The product does not modify the images or other files on the system in any way. It just uses the visual information of the images to sort them and then move them to their respective folders.

4.4.3 Security Requirements

This product does not pose a security or privacy risk as all the operations are done locally on the user's machine and is not connected to the internet. The product also does not change or modify system settings in any way.

Chapter 5

System Design

5.1 System Requirements Definition

This system should be able to access all the images stored on the user's computer. The system should then be able to process each and every image that the user desires to sort and organize. The system should select the correct images for object detection or face recognition. The system should then accurately identify the faces or objects and then move the images to their respective folders on the user's machine.

5.1.1 Functional requirements

- Provide user with navigation to their file system through the product.
- Allow user to sort the images based on the visual information in the image such as by object detection or face recognition.
- System should accurately select the images needed for processing.
- After the processing, the images should be immediately moved to their appropriate directories either by name of the object or name of the person.
- Allow the user to immediately switch to another directory after all the images of one directory has been sorted and organized
- Allow the user to close the application.

Use-case Diagram

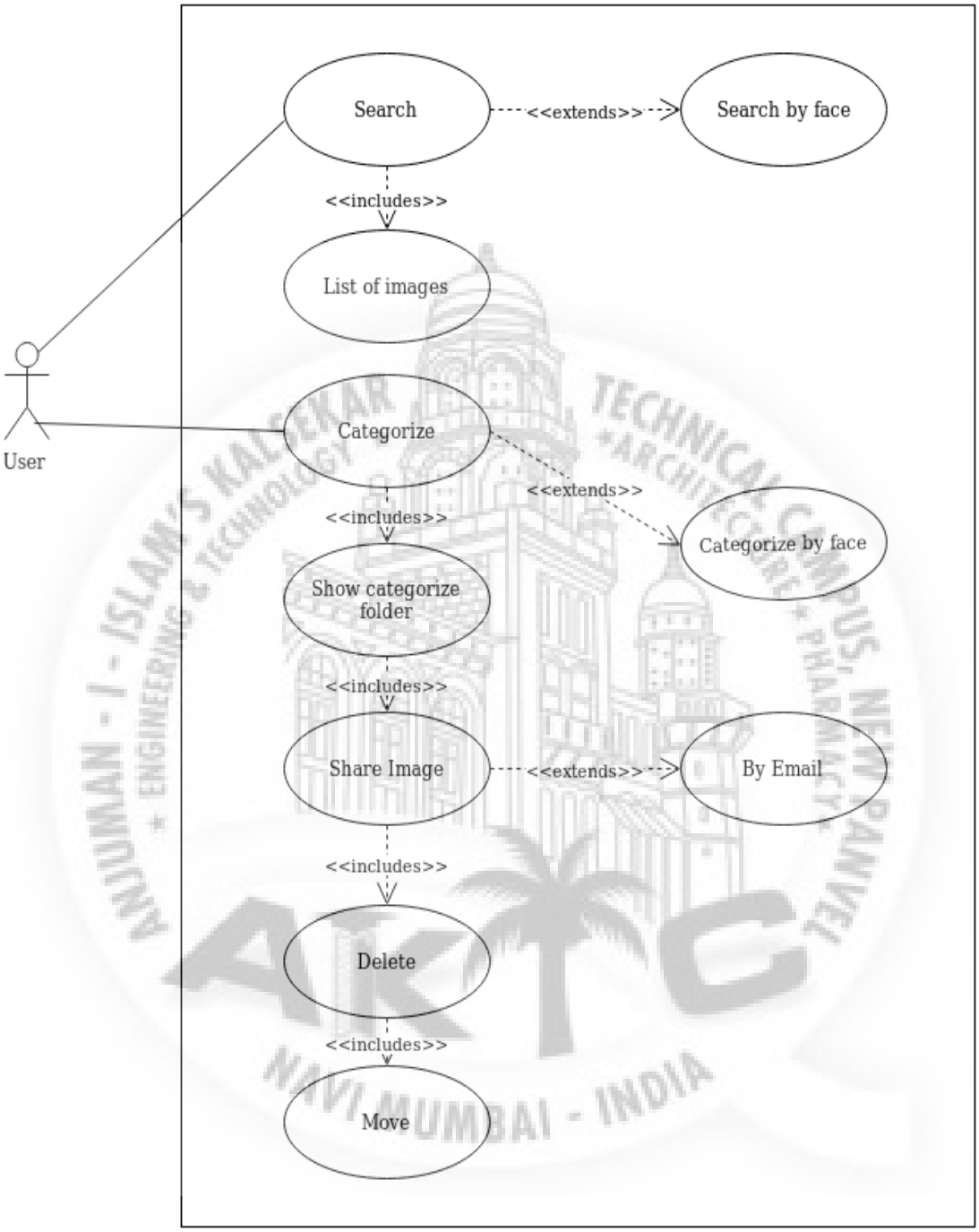

Figure 5.1: Use Case diagram

Data-flow Diagram

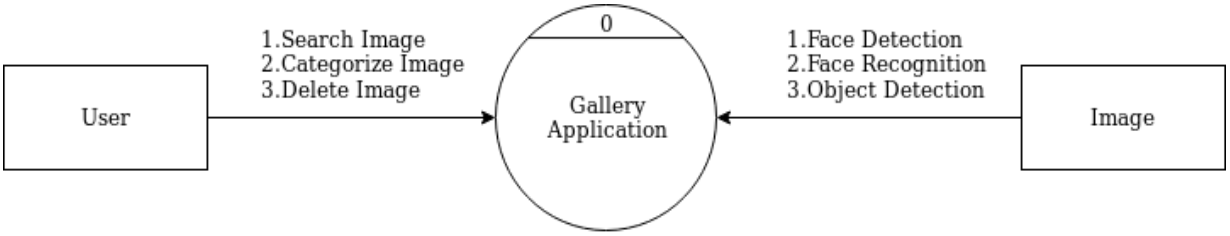

Figure 5.2: DFD level 0

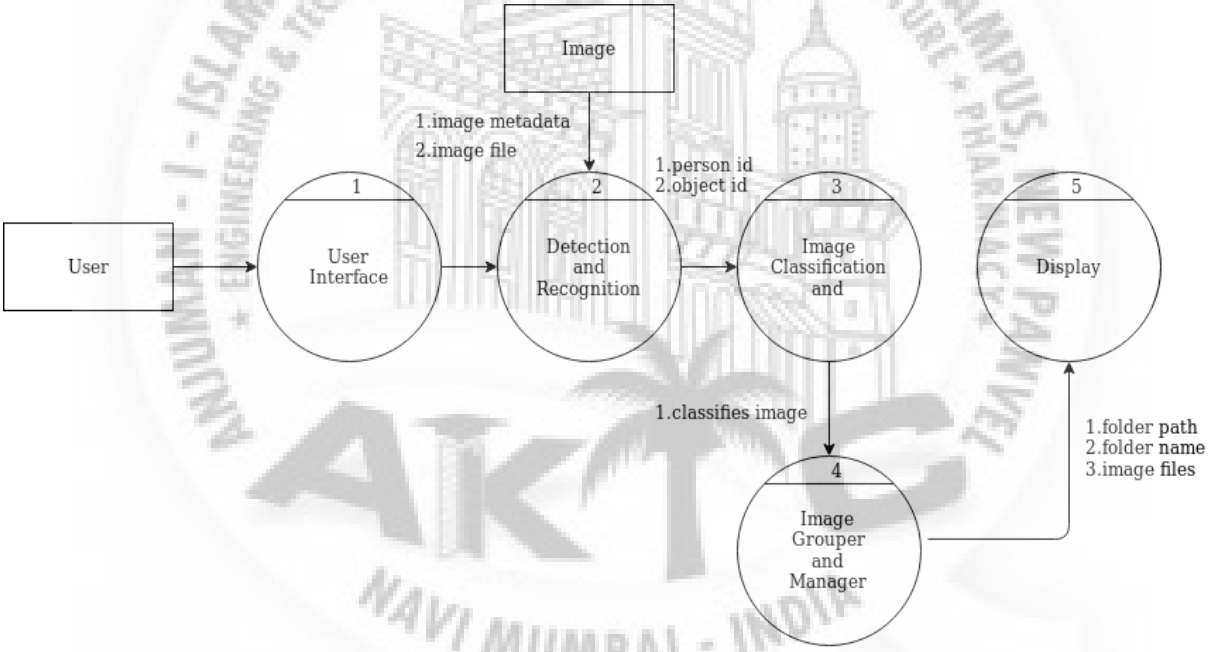

Figure 5.3: DFD level 1

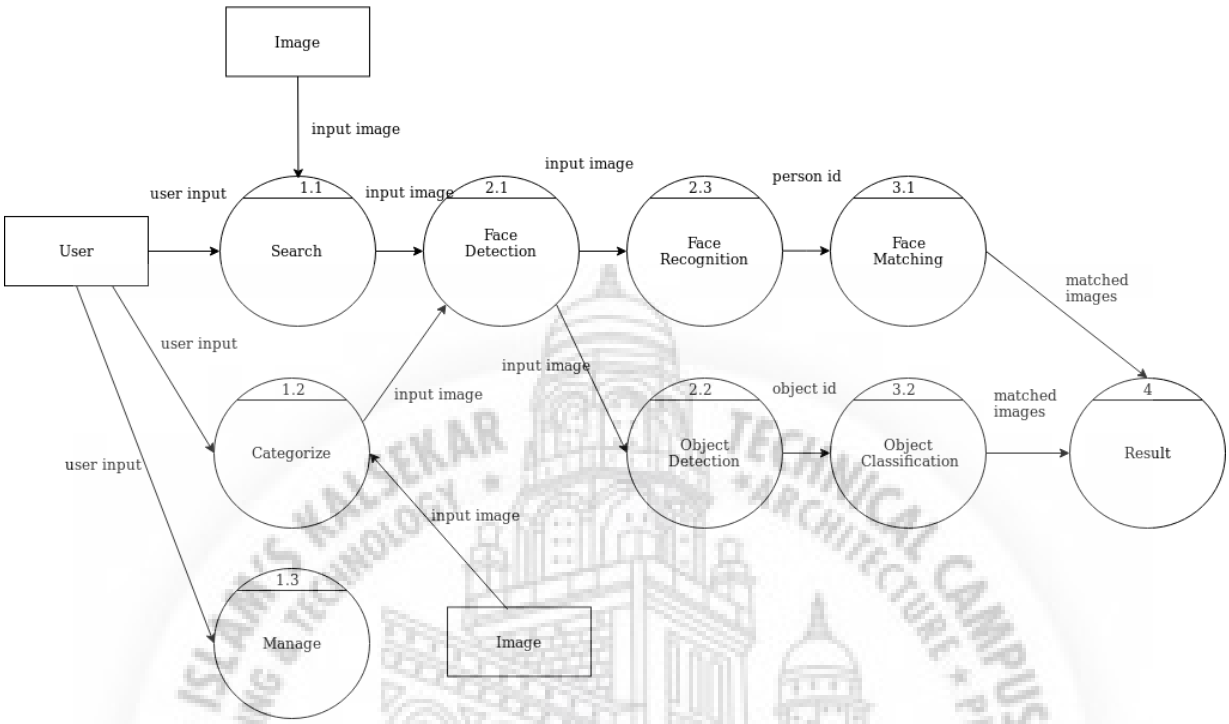

Figure 5.4: DFD level 2

5.1.2 System requirements (non-functional requirements)

- The system should process the images in a reasonable time.
- The system should not pose any risk to the user machine's system files and settings.
- The system should not risk damage or unintentional deletion of user files.
- The system should not crash while operating.
- The system should not freeze or crash the user's machine while operating.
- The system should not violate user privacy.

Database Schema/ E-R Diagram

5.2 System Architecture Design

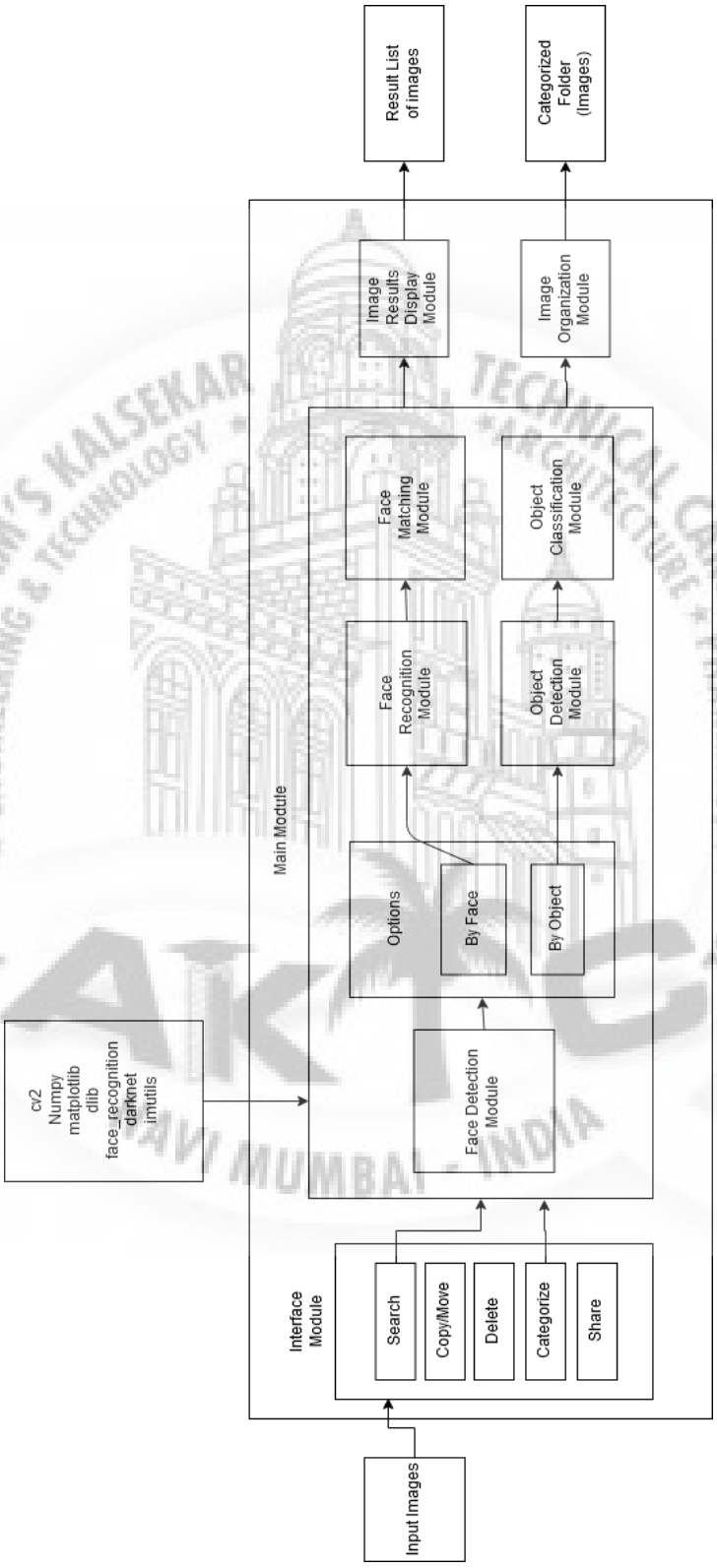

Figure 5.5: System Architecture

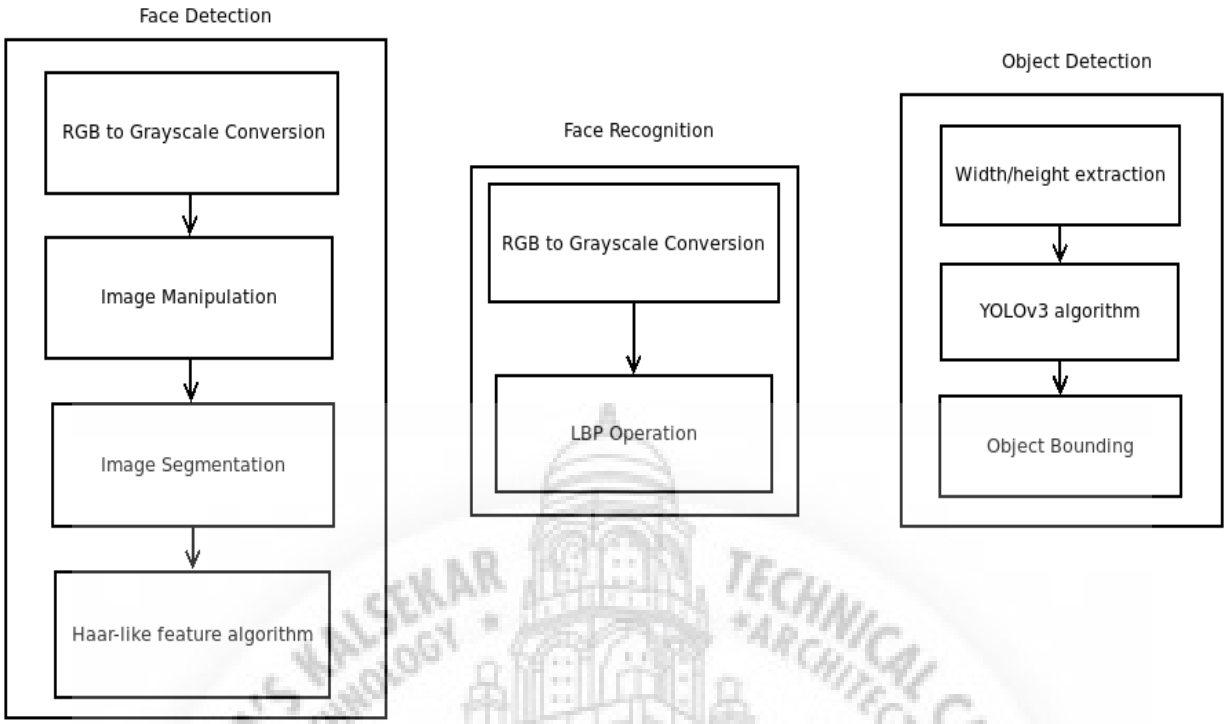

Figure 5.6: Face detection, face recognition and Object detection modules

5.3 Sub-system Development

The user interface is used by the user to interact with the product. Face detection module checks if an image has a face in it or not. If face found the face recognition module would try to identify the person in the image if possible. Face matching module would then match the identified faces with other images on the system. If face not found the object detection module would get to work and start identifying objects in the image instead. The object classification module would do the same work as the face matching module but for objects instead. The image organization module would then fetch the filenames and start putting the images into their appropriate folders.

5.3.1 User Interface

The user interface is the module from which the user would interact with the application. This interface would show all the options available to the user. The options are; search, categorize, copy/move, delete and share. The search and categorize options are used when the user wants to classify the images based on either the objects or faces in them then display as a search result or categorize the images in folders, respectively. An Option flag would be used to choose between search and categorize operations.

5.3.2 Face Detection Module

Face detection module, as the name suggests, this module is used to check whether in the input image there is face or not. If a face is found, then the face flag is set to true and the input image is passed further to the image recognition module. If no face is found, then the image is passed on to continue with object detection. Face detection is achieved using a Haar-Cascade face detector implemented using OpenCV.

5.3.3 Face Recognition Module

Face detection module is used to recognize the faces in the image that is obtained from the face detection module. After the faces of the persons in the image have been recognized, the module would then assign a unique ID which would be name, to the image. It will then pass the unique person ID to the Face Matching Module as its input. Face recognition is achieved using LBPH algorithm implemented through OpenCV.

5.3.4 Object Detection Module

Object detection module is the counterpart module to Face Detection module for objects. This module would take the image from the face detection module and apply object detection on the image. This would result as the object IDs of the objects being detected. These object IDs would be then passed onto the object classification module for further operations. The algorithm used for object detection is YOLO, which is implemented in OpenCV.

5.3.5 Face Matching Module

Face Matching is the module following face recognition module. This would take the person ID generated by the face recognition module and search through the current directory to find the images of faces with the same ID. If no matching IDs are found, then the problem for the same is returned as a message to the user. If matching images are found, then the current directory path and the filenames of the images are passed to either Image organization module or Image Results Display Module based on the Option flag chosen.

5.3.6 Object Classification Module

Object Classification is a counterpart module to the face matching module. It finds all the matching images with the same object IDs. It then takes the object IDs generated from the object detection module and then classifies them into their respective classes, and each class would be given its class name. These class names

and the filenames of the matched images and would be then passed to either Image Organization or Image Results Display modules based on the what the Option flag is set.

5.3.7 Image Organization Module

Image Organization module obtains the image filenames and their class names or person IDs from the object classification module or face matching module. It then creates new folders with the class names or person name as the new folder names. It then moves the images belonging to a certain class or a person to their corresponding folder. The same is then performed for all the classes and persons. The output of this module would then be the newly created folders. This module corresponds to Categorize operation.

5.3.8 Image Results Display Module

Image Results Display Module is a counterpart to image organization module. This module corresponds to Search operation. It takes all the class names or the person ID from Face Matching module or Object classification module, respectively. It would display all the images of a class or a person as a list to the user. The output would be the sorted list of images.

5.4 Systems Integration

The GUI or the User Interface module is connected to the Face Detection Module. The face detection module is connected to face recognition and object detection modules. The above modules are connected to face matching module and object classification module respectively. Image organization module and Image results display module are connected to the previous modules.

5.4.1 Class Diagram

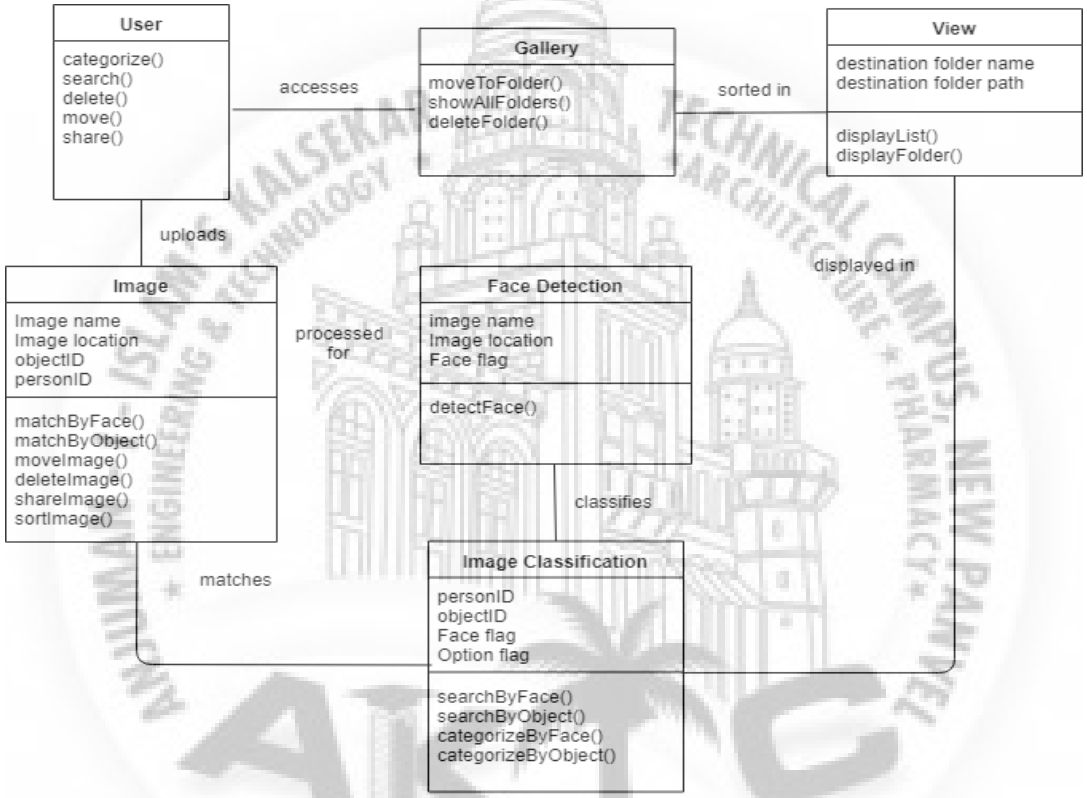

Figure 5.7: Class Diagram

5.4.2 Sequence Diagram

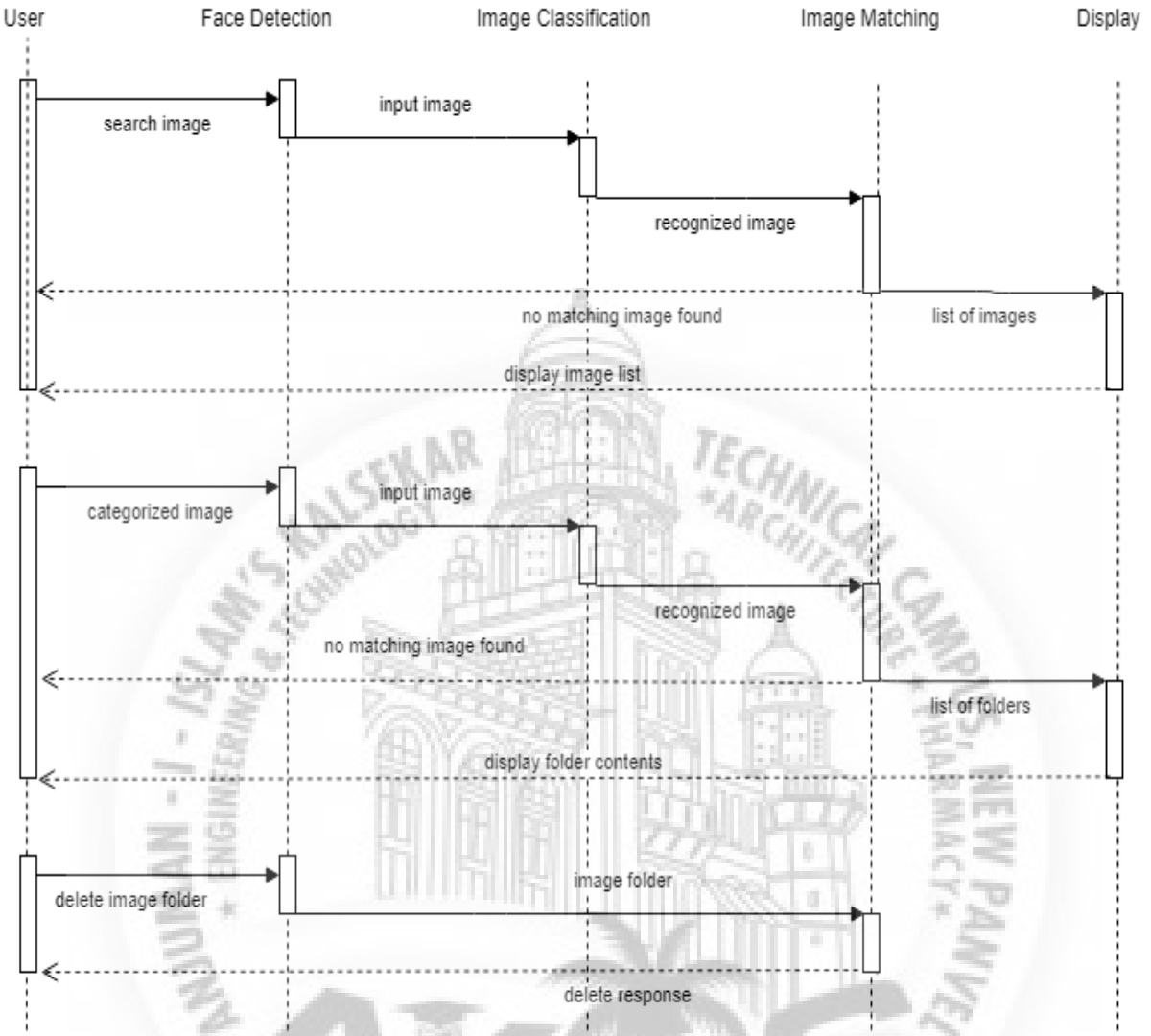

Figure 5.8: Sequence diagram

Chapter 6

Implementation

6.1 Object Detection Module

```
import cv2
import os
import numpy as np
from collections import Counter

def object_det(directory, filename):
  tagList = []
  args = {'image': '', 'yolo': 'yolo-coco', 'confidence': 0.5, 'threshold': 0.3}
  args['image'] = os.path.join(directory,filename)

  labelsPath = os.path.sep.join([args["yolo"], "coco.names"])
  LABELS = open(labelsPath).read().strip().split("\n")

  np.random.seed(42)

  weightsPath = os.path.sep.join([args["yolo"], "yolov3.weights"])
  configPath = os.path.sep.join([args["yolo"], "yolov3.cfg"])

  net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

  image = cv2.imread(args["image"])
  (H, W) = image.shape[:2]

  ln = net.getLayerNames()
  ln = [ln[i[0] - 1] for i in net.getUnconnectedOutLayers()]

  blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416),
    swapRB=True, crop=False)
  net.setInput(blob)

  layerOutputs = net.forward(ln)

  boxes = []
  confidences = []
  classIDs = []

  for output in layerOutputs:
    for detection in output:

      scores = detection[5:]
      classID = np.argmax(scores)
      confidence = scores[classID]

```

```
        if confidence > args["confidence"]:
          box = detection[0:4] * np.array([W, H, W, H])
          (centerX, centerY, width, height) = box.astype("int")

          x = int(centerX - (width / 2))
          y = int(centerY - (height / 2))

          boxes.append([x, y, int(width), int(height)])
          confidences.append(float(confidence))
          classIDs.append(classID)

    idxs = cv2.dnn.NMSBoxes(boxes, confidences, args["confidence"],
        args["threshold"])

    if len(idxs) == 0:
      pass
    else:
      for i in idxs.flatten():
          #if confidences[i] >= 0.9:
        tagList.append(LABELS[classIDs[i]])
        count = Counter(tagList)
      a, b = count.keys(), count.values()
      keysList = list(a)

      return keysList[0]
```

6.2 Face Detection Module

```
import cv2
import os

def detect(directory,filename):
    '''
    Detect Faces in a directory and return a list containing the location of the
        faces.
    Input:- takes directory where images are present.
    Output:- return list which containt location of faces
    '''
    face_cascade = cv2.CascadeClassifier('pre-trained_data/haarcascades/
        haarcascade_frontalface_alt.xml')

    imgpath = os.path.join(directory, filename) #path for the specific image
    img = cv2.imread(imgpath) #reading the image from current folder
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    #Checking if faces are detected or not
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.2, minNeighbors=5)

    return faces
```

6.3 Face Recognition Module

```
import cv2
import os
import pickle
import face_detect
def face_recognize(directory,filename,faces):

    imgpath = os.path.join(directory, filename)
    recognizer = cv2.face.LBPHFaceRecognizer_create()
    recognizer.read("training_data.yml") #NEED training data
    lables = {}
    with open('lables.pickle', 'rb') as f:
        lables_ = pickle.load(f)
        lables = {v:k for k,v in lables_.items()}

    image1 = cv2.imread(imgpath)

    gray = cv2.cvtColor(image1,cv2.COLOR_BGR2GRAY)

    for (x, y, w, h) in faces:
        roi = gray[y:y+h, x:x+w]

        id_, conf = recognizer.predict(roi)
        if conf>60:
            return lables[id_]
```

6.4 User Interface

```
import cv2
import os
import sys

from os import makedirs
from PyQt5.QtWidgets import *
from PyQt5.QtCore import *
from PyQt5 import QtGui
from win32api import GetSystemMetrics

import face_detect
import object_detect
import face_recognition

screen_res = (GetSystemMetrics(0),GetSystemMetrics(1)) #Get system resolution(
    Windows OS only)
moveToPath = "f:/testfolder/facefolder"

class Widget(QWidget):
    def __init__(self, *args, **kwargs):
        QWidget.__init__(self, *args, **kwargs)
        layout = QVBoxLayout(self)

        self.top = 128
        self.left = 128
        self.width = screen_res[0]/1.5
        self.height = screen_res[1]/1.5
        self.setGeometry(self.top, self.left, self.width, self.height)
        self.treeview = QTreeView()
        layout.addWidget(self.treeview)

        path = "f:/"

        self.dirModel = QFileSystemModel()
        self.dirModel.setRootPath(QDir().dirName())
        self.dirModel.setFilter(QDir.NoDotAndDotDot | QDir.AllDirs)

        self.treeview.setModel(self.dirModel)

        self.treeview.setRootIndex(self.dirModel.index(path))

        self.header = self.treeview.header()

        self.header.setSectionResizeMode(0, QHeaderView.Stretch)
        self.header.setSectionHidden(1, True)
        self.header.setSectionHidden(2, True)
        self.header.setSectionHidden(3, True)

        #Menu
        self.treeview.doubleClicked.connect(self.integration)

    def move_to_folder(self,directory,filename,make_dir_path):
        current_path = os.path.join(directory, filename)
        dest_path = os.path.join(make_dir_path, filename)
        os.replace(current_path, dest_path)
        print("Moved Successfully {}".format(dest_path))

    def integration(self,index):
        directory = self.dirModel.fileInfo(index).absoluteFilePath()
```

```
        try:
            os.makedirs(moveToPath) #create a new folder as destination
        except OSError:
            print ("Failed to create a new folder %s" % moveToPath)

        #Scanning all images in a given directory
        for filename in os.listdir(directory):
            if filename.endswith(".jpg") or filename.endswith(".jpeg") or
                filename.endswith(".png"):
                faces = face_detect.detect(directory,filename)
                if faces is ():
                    print("No Faces found in {}".format(filename))
                    object_detected = object_detect.object_det(directory,
                        filename)
                    if object_detected :
                        make_dir_path = os.path.join("f:/testfolder/objectfolder
                            ",object_detected)
                        #moving the image to the destination folder
                        if os.path.exists(make_dir_path):
                            self.move_to_folder(directory,filename,make_dir_path)
                        else:
                            os.makedirs(make_dir_path)
                            self.move_to_folder(directory,filename,make_dir_path)
                else:
                    print("Face found! in {}".format(filename))
                    recognized_face = face_recognition.face_recognize(directory,
                        filename, faces)
                    if recognized_face :
                        print("Face Recognized")
                        #moving the image to the destination folder
                        make_dir_path = os.path.join("f:/testfolder/facefolder",
                            recognized_face)
                        if os.path.exists(make_dir_path):
                            self.move_to_folder(directory,filename,make_dir_path)
                        else:
                            os.makedirs(make_dir_path)
                            self.move_to_folder(directory,filename,make_dir_path)


if __name__ == '__main__':
    app = QApplication(sys.argv)
    w = Widget()
    w.setWindowTitle("Image Grouper")
    w.setWindowIcon(QtGui.QIcon("icon.png"))
    w.show()
    sys.exit(app.exec_())
```

Chapter 7

System Testing

WRITE HERE.

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Navigation	Reach the directory to be scanned	Respond to the click	User finds the directory
T02	Processing of images	Complete the processing of images	Get image files and start processing	The processing finishes without errors
T03	Organization of images	Move the images to their respective folders	Files are moved from one directory to target directories	Correct images are moved

7.2 Sample of a Test Case

Title: Processing of images.
Description: The images should be scanned and face detection, object detection and face recognition algorithms are applied to them.

Precondition: The user must have already navigated to the desired folder.

Assumption: A supported platform i.e x86 is used.

Test Steps:

1. Double click on the folder.
2. Scan the folder for images.
3. Process the images

4. Pass control to start moving the images.
5. Stop.

Expected Result: The processing finishes without any errors.
Actual Result: The processing finishes without any errors.

Chapter 8

Screenshots of Project

8.1 User Interface

Figure 8.1: User Interface 1

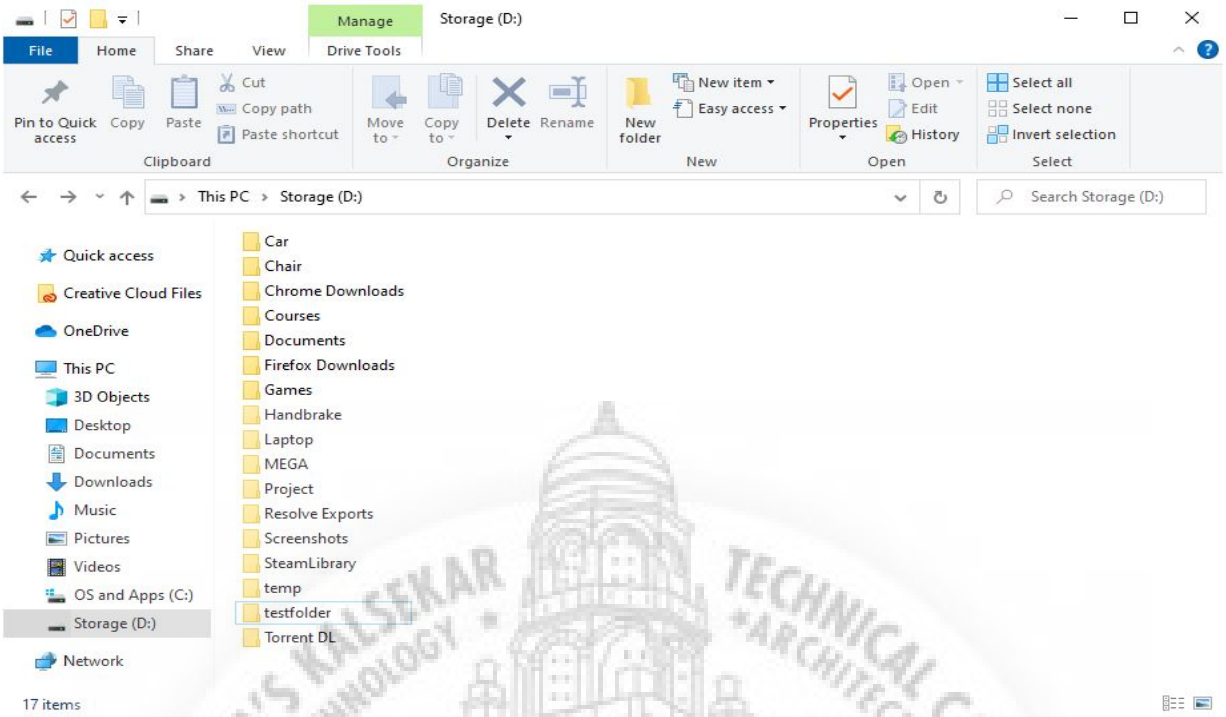

Figure 8.2: User Interface 2

8.2 Object Detection

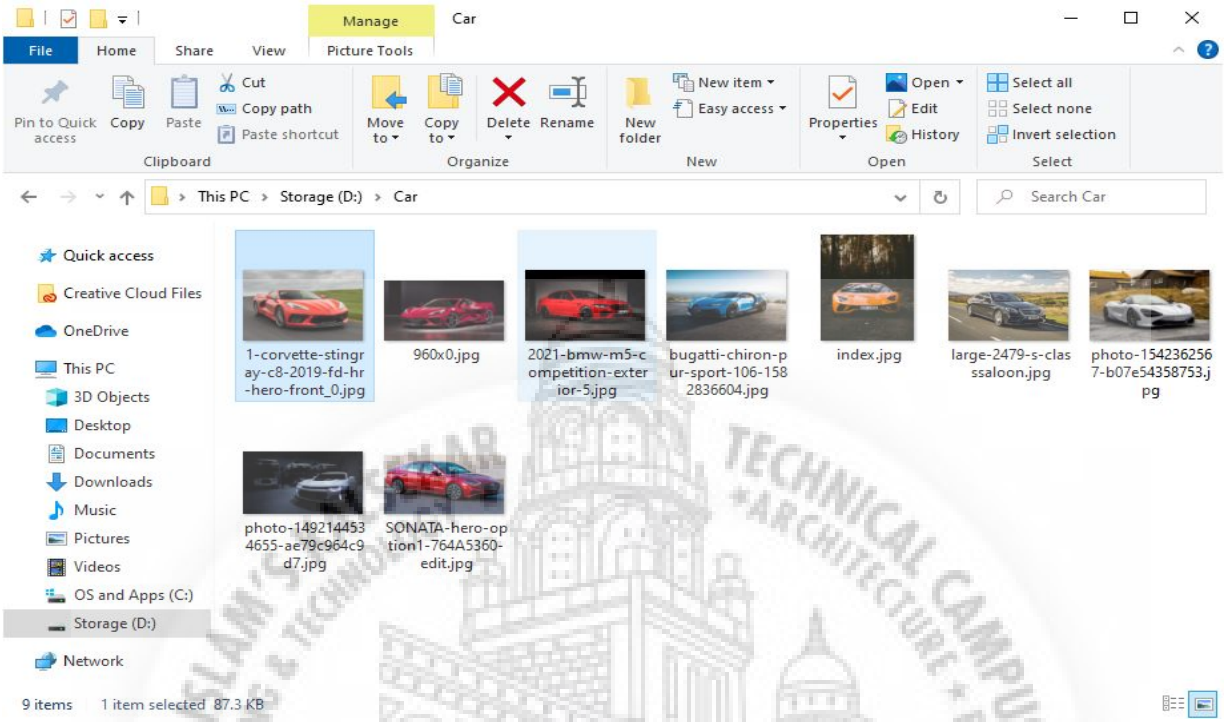

Figure 8.3: File system output

Figure 8.4: Console output

8.3 Face Detection

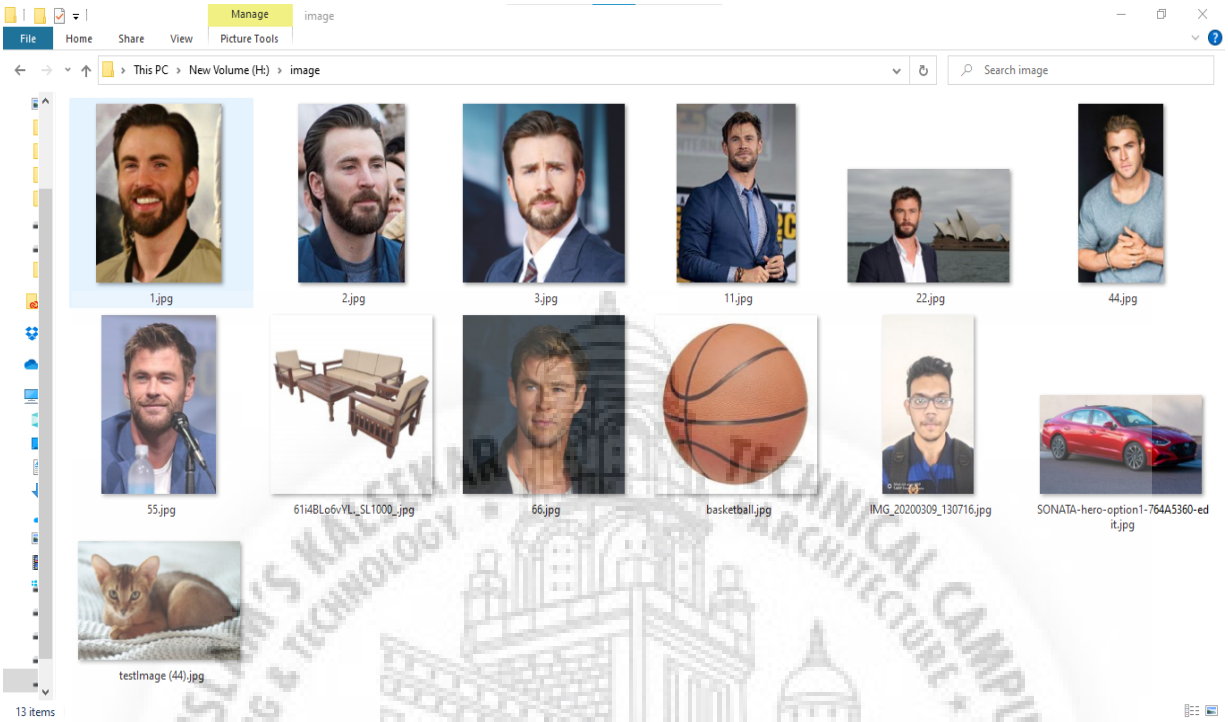

Figure 8.5: File system output

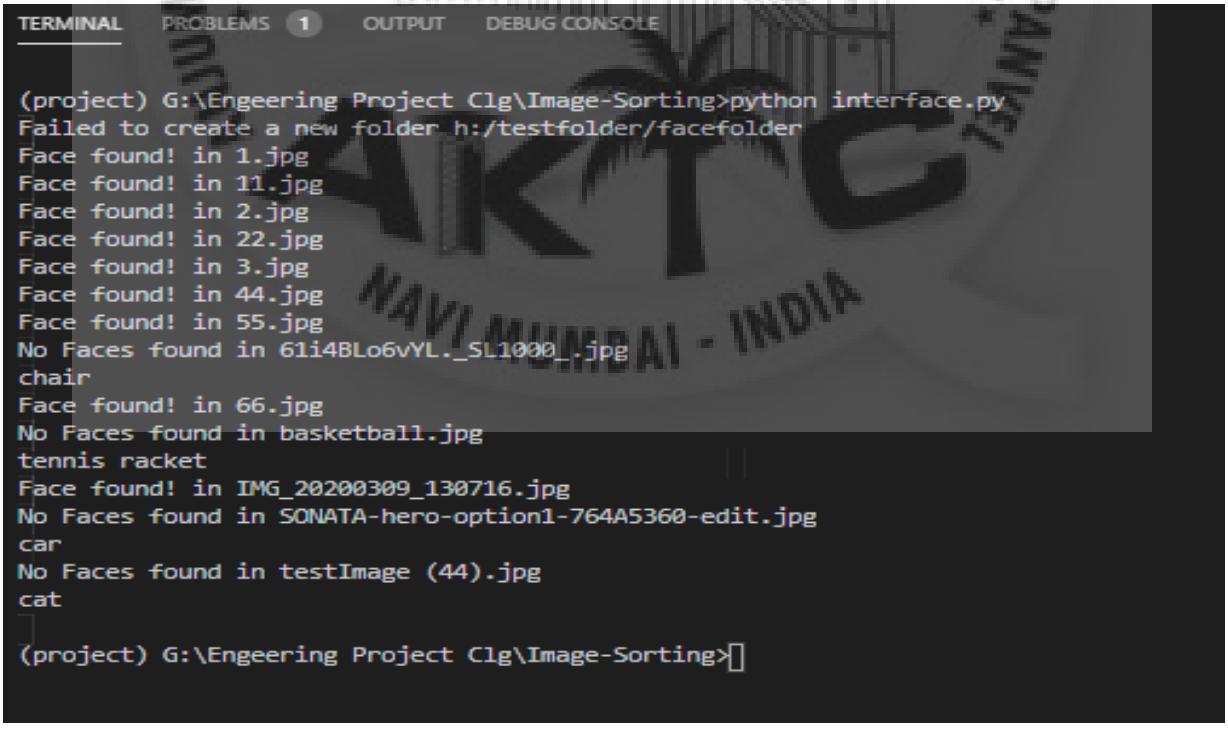

Figure 8.6: Console output

8.4 Face Recognition

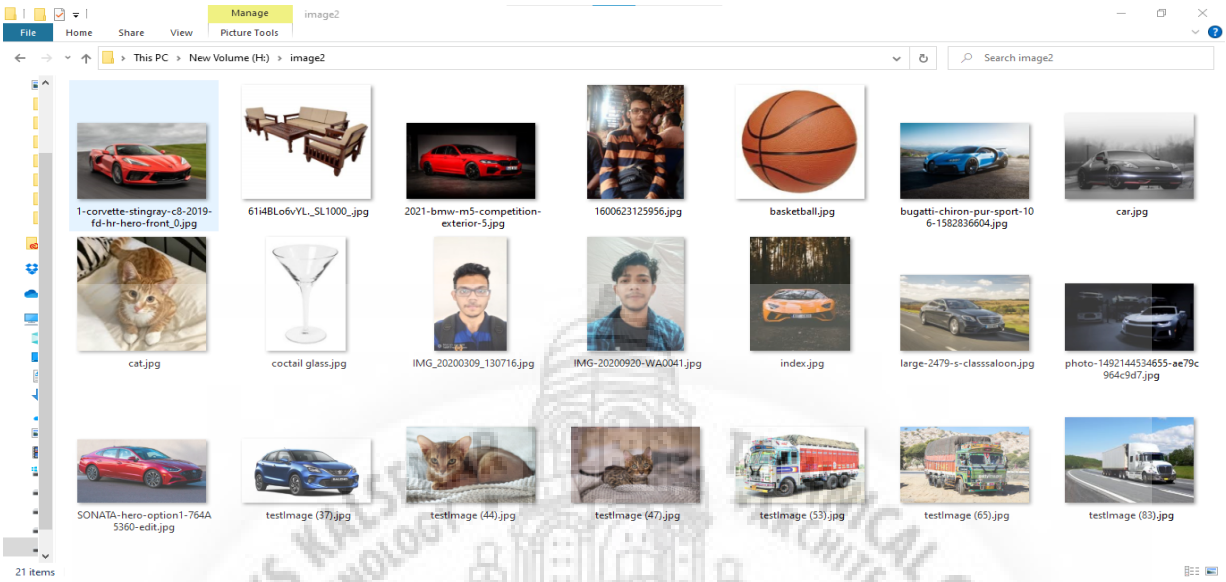

Figure 8.7: File system output

```
TERMINAL   PROBLEMS 1   OUTPUT   DEBUG CONSOLE

(project) G:\Engeering Project Clg\Image-Sorting>python interface.py
No Faces found in 1-corvette-stingray-c8-2019-fd-hr-hero-front_0.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\1-corvette-stingray-c8-2019-fd-hr-hero-front_0.jpg
Face found! in 1600623125956.jpg
Face Recognized
Moved Successfully h:/testfolder/facefolder\rehan\1600623125956.jpg
No Faces found in 2021-bmw-m5-competition-exterior-5.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\2021-bmw-m5-competition-exterior-5.jpg
No Faces found in 61i4BLo6vYL._SL1000_.jpg
chair
Moved Successfully h:/testfolder/objectfolder\chair\61i4BLo6vYL._SL1000_.jpg
No Faces found in basketball.jpg
tennis racket
Moved Successfully h:/testfolder/objectfolder\tennis racket\basketball.jpg
No Faces found in bugatti-chiron-pur-sport-106-1582836604.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\bugatti-chiron-pur-sport-106-1582836604.jpg
No Faces found in car.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\car.jpg
No Faces found in cat.jpg
cat
Moved Successfully h:/testfolder/objectfolder\cat\cat.jpg
No Faces found in coctail glass.jpg
vase
Moved Successfully h:/testfolder/objectfolder\vase\coctail glass.jpg
Face found! in IMG-20200920-WA0041.jpg
Face Recognized
Moved Successfully h:/testfolder/facefolder\sohail\IMG-20200920-WA0041.jpg
Face found! in IMG_20200309_130716.jpg
Face Recognized
Moved Successfully h:/testfolder/facefolder\rehan\IMG_20200309_130716.jpg
No Faces found in index.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\index.jpg
No Faces found in large-2479-s-classsaloon.jpg
car
Moved Successfully h:/testfolder/objectfolder\car\large-2479-s-classsaloon.jpg
No Faces found in photo-1492144534655-ae79c964c9d7.jpg
```

Figure 8.8: Console output

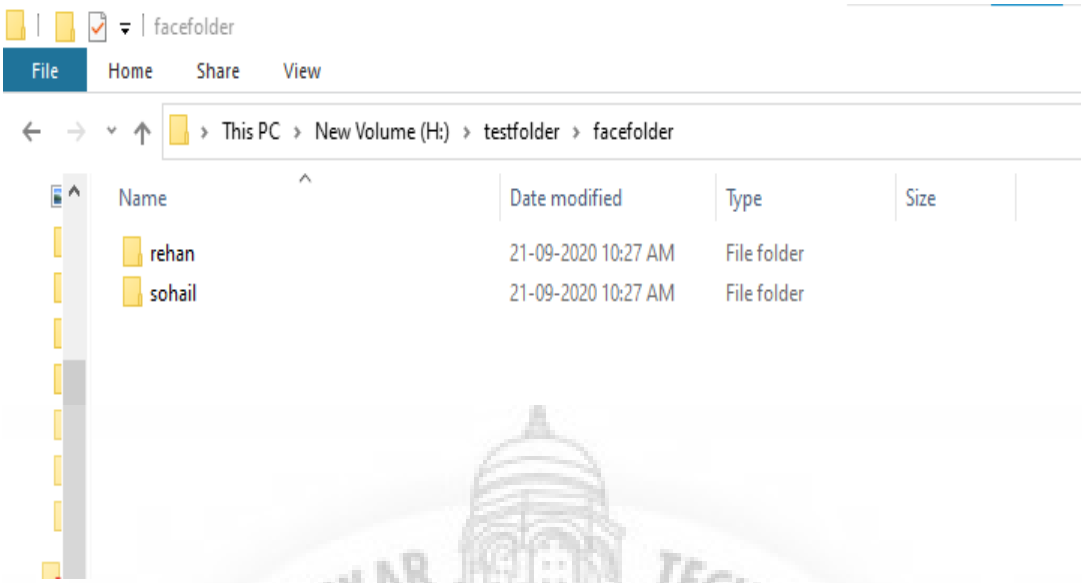

Figure 8.9: File system output

Figure 8.10: File system output

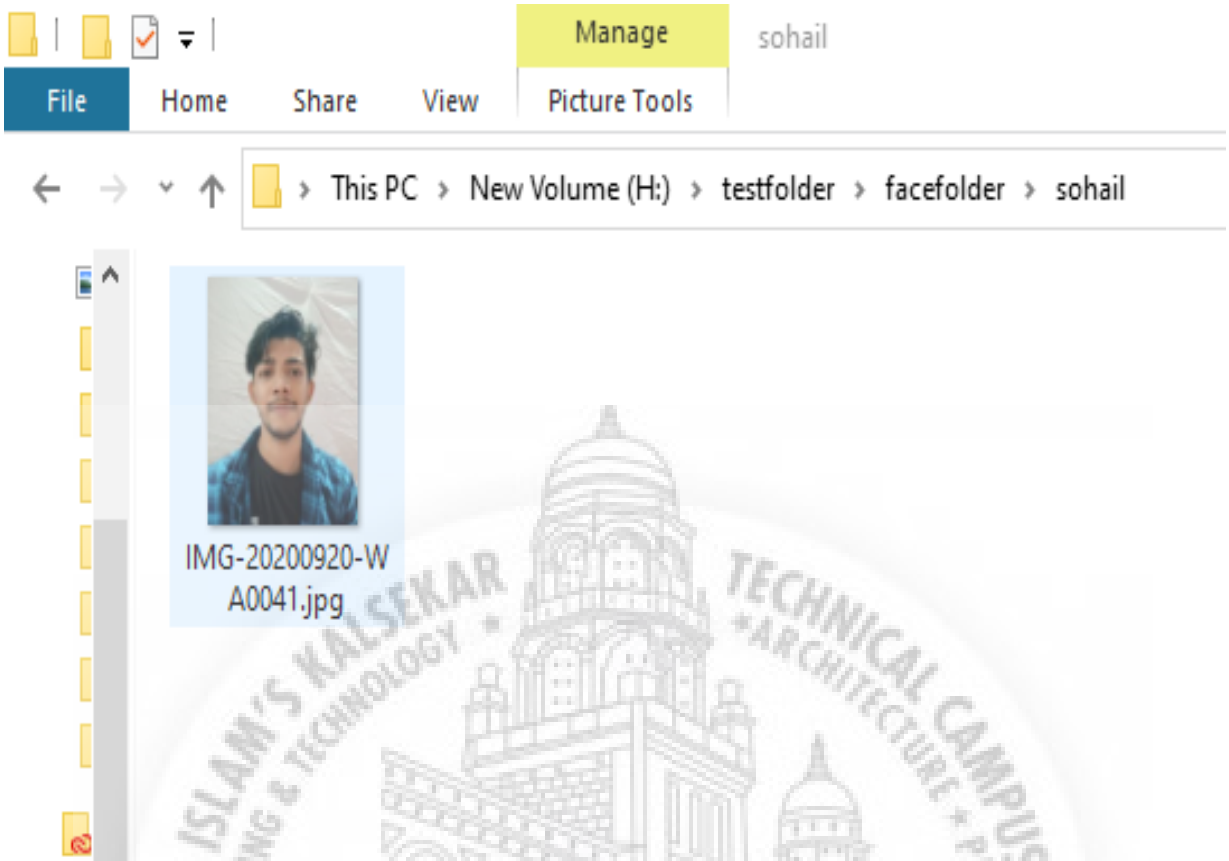

Figure 8.11: File system output

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

We realized the problem of organizing images in bulk i.e sorting, moving, deleting, sharing of images and decided to come up with a system to solve the same.

Our project makes use of new and emerging technologies of deep learning and machine learning to bring a new perspective to sort and group the images.

Finally, this project is fully open source so that other people can contribute their valuable ideas and concepts for this project.

9.2 Future Scope

- Mobile Application can be made for both Android and IOS.
- Add Support to view the images in the application.
- Ability to search images based on their content.

References

[1] Meera M K and Shajee Mohan B S, *"Object recognition in images,"*; 2016 International Conference on Information Science (ICIS), Kochi, 2016, pp. 126-130.doi: 10.1109/INFOSCI.2016.7845313

[2] X. Zhou, W. Gong, W. Fu and F. Du, *"Application of deep learning in object detection,"*; 2017 IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS), Wuhan, 2017, pp. 631-634.doi: 10.1109/ICIS.2017.7960069

[3] C. Tang, Y. Feng, X. Yang, C. Zheng and Y. Zhou, *"The Object Detection Based on Deep Learning,"*; 2017 4th International Conference on Information Science and Control Engineering (ICISCE), Changsha, 2017, pp. 723-728.doi: 10.1109/ICISCE.2017.156

[4] X. Han and Q. Du, *"Research on face recognition based on deep learning,"*; 2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC), Beirut, 2018, pp. 53-58.doi: 10.1109/DINWC.2018.8356995

[5] R. L. Galvez, A. A. Bandala, E. P. Dadios, R. R. P. Vicerra and J. M. Z. Maningo, *"Object Detection Using Convolutional Neural Networks,"*; TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 2023-2027.doi: 10.1109/TENCON.2018.8650517

[6] YOLO — You only look once, real time object detection explained *towardsdatascience.com*; Manish Chablani https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection

[7] Face Detection using Haar Cascades *towardsdatascience.com*; https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html

[8] Face Recognition: Understanding LBPH Algorithm *opencv-python-tutroals.readthedocs.io*; Kelvin Salton do Prado https://towardsdatascience.com/face-recognition-how-lbph-works-90ec258c3d6b

Achievements

1. Publication

Image Sorting Using Object Detection and Face Recognition; Shaikh Rehan, Shaikh Arbaz, Shaikh Sohail, Khan Mubashir; International Journal of Innovative Science and Research Technology (IJISRT), March 31, 2020 (https://ijisrt.com/image-sorting-using-object-detection-and-face-recognition)

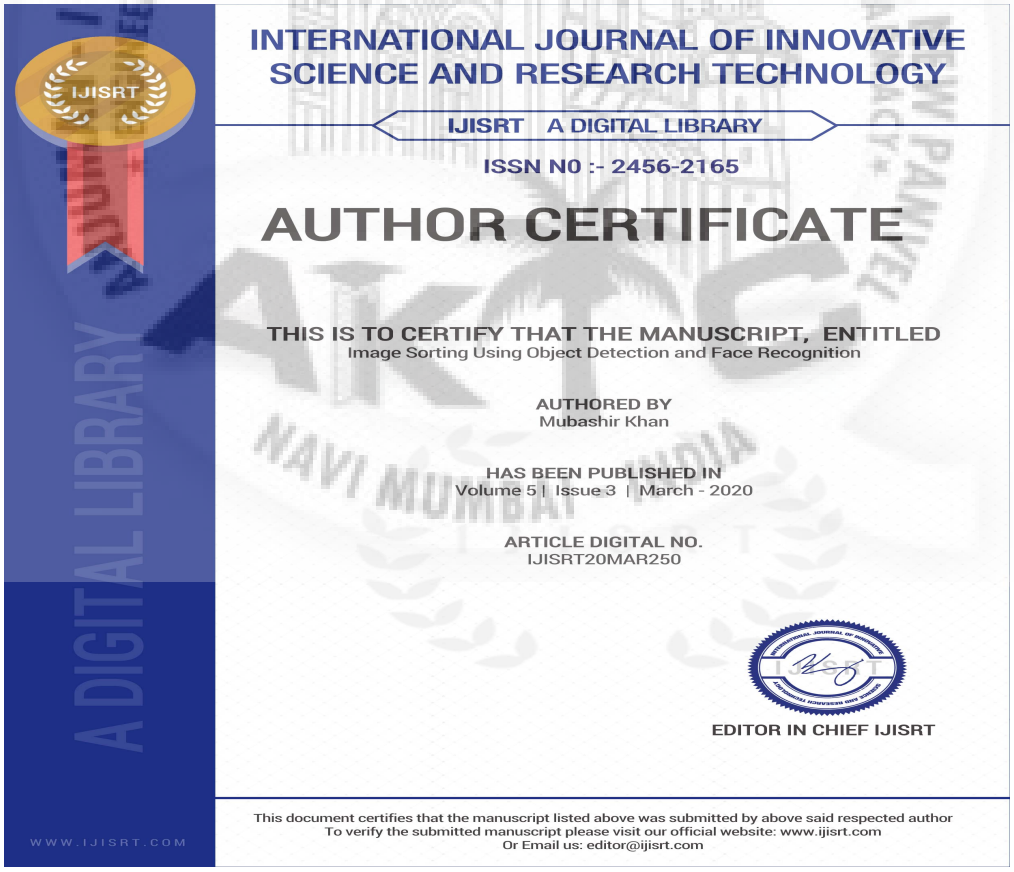

Figure 9.1

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Image Sorting Using Object Detection and Face Recognition

AUTHORED BY
Shaikh Rehan

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR250

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

Figure 9.2

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Image Sorting Using Object Detection and Face Recognition

AUTHORED BY
Shaikh Arbaz

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR250

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

Figure 9.3

INTERNATIONAL JOURNAL OF INNOVATIVE SCIENCE AND RESEARCH TECHNOLOGY

IJISRT A DIGITAL LIBRARY

ISSN NO :- 2456-2165

AUTHOR CERTIFICATE

THIS IS TO CERTIFY THAT THE MANUSCRIPT, ENTITLED
Image Sorting Using Object Detection and Face Recognition

AUTHORED BY
Shaikh Sohail

HAS BEEN PUBLISHED IN
Volume 5 | Issue 3 | March - 2020

ARTICLE DIGITAL NO.
IJISRT20MAR250

EDITOR IN CHIEF IJISRT

This document certifies that the manuscript listed above was submitted by above said respected author
To verify the submitted manuscript please visit our official website: www.ijisrt.com
Or Email us: editor@ijisrt.com

Figure 9.4

2. Project Competition

Image Sorting Using Object Detection and Face Recognition; Shaikh Rehan, Shaikh Arbaz, Shaikh Sohail, Khan Mubashir;
6th National Level Project Exhibition cum Poster Presentation, March 13, 2020 (Venue : Universal College of Engineering, Palghar)

Vidya Vikas Education Trust's

Universal College of Engineering

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)
Near Bhajansons and Punyadham, Kaman Bhiwandi Road, Vasai, Palghar-401208.
Accredited with B+ Grade by NAAC
In association with
I.E.T.E.-I.S.F.

"6th National Level Project Exhibition cum Poster Presentation"

Certificate of Participation

This is to certify that Ms./Mr. SHAIKH REHAN

of ANJUMAN-I-ISLAM'S KALSEKAR TECHNICAL CAMPUS has participated in the "6th National Level Project Exhibition cum Poster Presentation"

Date: 13th March 2020

Dr. J.B. Patil
(Campus Director)

Figure 9.5

Vidya Vikas Education Trust's

Universal College of Engineering

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)
Near Bhajansons and Punyadham, Kaman Bhiwandi Road, Vasai, Palghar-401208.
Accredited with B+ Grade by NAAC

In association with
I.E.T.E.-I.S.F.

"6th National Level Project Exhibition cum Poster Presentation"

Certificate of Participation

This is to certify that Ms./Mr. SHAIKH ARBAZ

of ANJUMAN -I- ISLAM'S KALSEKAR TECHNICAL CAMPUS has participated in the "6th National Level Project Exhibition cum Poster Presentation"

Date: 13th March 2020

Dr. J.B. Patil
(Campus Director)

Figure 9.6

Vidya Vikas Education Trust's

Universal College of Engineering

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)
Near Bhajansons and Punyadham, Kaman Bhiwandi Road, Vasai, Palghar-401208.
Accredited with B+ Grade by NAAC
In association with
I.E.T.E.-I.S.F.

"6th National Level Project Exhibition cum Poster Presentation"

Certificate of Participation

This is to certify that Ms./Mr. SHAIKH SOHAIL

of ANJUMAN -I - ISLAM'S KALSEKAR TECHNICAL CAMPUS has participated in the "6th National Level Project Exhibition cum Poster Presentation"

Date: 13th March 2020

Dr. J.B. Patil
(Campus Director)

Figure 9.7