Smart Gallery

Visual Image Search in Android via Machine Learning

By:

Arslan Arif

Supervised by:

Syed Muhammad Naqi

Department of Computer Sciences,

Quaid-i-Azam University, Islamabad, Pakistan.

BS Computer Science

(2015 –2019)

ACKNOWLEDGEMENT

First and foremost, I have to thank my project supervisor, Syed Muhammad Naqi. Without his assistance and support throughout the process, this project would have never been accomplished. He has also been one of the best teachers I ever met in my educational career. I would like to thank you very much for all your support and understanding over the years.

I would also like to show gratitude to all other teachers, including Dr. Shuaib Karim, Dr. Asim Rafique, Dr. Muhammad Imran and others. Dr. Shuaib Karim has been my most favorite teacher. His teaching style and enthusiasm made a strong impression on me and I have always carried positive memories of his classes with me. Dr. Asim Rafique has played major role in my final year studies. He is the reason I opted Machine Learning as my career choice. Dr. Muhammad Imran has been one of the most knowledgeable teachers in my university career. I learned most of the technical and advance concepts of OOP from him. I am very grateful to these teachers specially for bringing out the best in me.

I would also like to say thank you to all the teachers from other departments who have taught me many things in life other than academics. I would especially like to show my gratitude to Ms. Saira Ferdous from Psychology, Dr. Sardar Muhammad Bilal from Mathematics, Mr. Fayyaz Hussain from Area Studies, Ms. Khafsa Malik from Plant Sciences, Dr. Shamoona Fawad Qazi from Physics and Mr. Muhammad Tayyab from Economics. I learned a lot from them.

Getting through my undergraduate required more than academic and technical support and I have many people to thank for listening to and, at times, having to tolerate me over the past four years. I cannot begin to express my gratitude and appreciation for their friendship. I would like to thank all of my friends, fellows and seniors for their support and coordination.

At the end, I would like to mention the most important people i.e. my family. None of this could have happened without my family. My father, who always believed in me and my brother who supported me financially throughout my educational career. Without their support I would have never completed my BS. To my parents and my siblings – I am forever grateful.

Arslan Arif (2015-2019)

ABSTRACT

Smart Gallery is an android smartphone photo gallery application that provides visual search option to the user. It is hard for user to find an image from thousands of images in the phone storage. The internal storage of the android smartphones is increasing day by day and one of the main features of latest android phones is mobile photography. People take and share a lot of photos and the smartphone storage is so big that sometimes users cannot find a photo by simple browsing. So the motivation of this project is to provide a visual search option to the users. Users can search images for more than eighty categories of objects so it becomes easy to find a photo.

This is all possible due to deep neural networks. The application uses a model trained using deep neural network. The model is imported to android application after being trained on high performance computer. The application then uses the trained model to perform object detection and recognize objects from the images. Thus the user can find any images by just searching an object name like cat, person, car etc.

Table of Contents

Chapter-1

Software Project Management Plan

1.1 Introduction

This chapter describes the Software Project Management Plan of the project i.e. Smart Gallery.

1.1.1 Project Overview

This project is an android mobile gallery application designed to provide visual search option to the user. User of this application can see the images in gallery and search the images by object names such as ‘car’, ‘person’, ‘cat’ etc.

1.1.2 Project Deliverables

There are following deliverables in this project:

- Software Project Management Plan (SPMP)
- Software Requirement Specifications (SRS)
- Software Design Description (SDD)
- Software Implementation (Source Code)
- Software Implementation Document (SID)
- Software Test Documentation (STD)

1.1.3 Vocabulary

Following terms are used throughout the document. There is a specific meaning of each term as defined.

Table 1: Vocabulary

Term	Description
SPMP	Software Project Management Plan
SRS	System Requirements Specification
SSD	Software Design Description
SID	Software Implementation Document
STD	Software Test Document

Smart Gallery	It is the title of project that is used throughout the document.
UC	Use Case
TC	Test Case

1.2 Project Organization

It contains the overall organization details about the project i.e. the process model used, the roles and responsibilities, tools and techniques and the other resources used in the project.

1.2.1 Software Process Model

The project is divided into two major parts. First part is android app development and the second part is training object detection algorithms on datasets of different objects. I used incremental process model for the development of this project.

1.2.2 Roles and Responsibilities

It is an individual project, so, I am responsible for all the planning, management, design, construction and deployment of this project while the project supervisor and the project evaluation committee are responsible for reviews and feedback of all deliverables on due dates.

1.2.3 Tools and Techniques

It is an android application so android studio is used for development of this project. 'Java for Android' and 'Kotlin' are used as programming languages for implementation. Python 3 along with Tensorflow is used for object detection algorithms. Android smartphone with API level 28 (Android version 8.0) is used primarily for deployment testing.

1.2.4 Resources Used

Following resources are used for the documentation and implementation of this project.

Table 2: Material Resource used

Resource Name	Used for
MS. Word	Documentation
Project Libre	Project Plan and Gantt chart
Argo-UML	UML diagrams design
Android Studio	Front end application development
PyCharm	Transfer Learning and converting model.

1.3 Project Plan

Project plan consists of the milestones, time to achieve those milestones and other requirements and resources needed for the project completion.

	Name	Duration	Start	Finish	Pred...	Resource Names
1	**Smart Gallery**	162 days	**12/3/18 8:00 AM**	**7/16/19 5:00 PM**		
2	Problem understanding	1 day	12/3/18 8:00 AM	12/3/18 5:00 PM		
3	**Make Software Project Management Plan**	5 days	**12/4/18 8:00 AM**	**12/10/18 5:00 PM**	**2**	**Arslan Arif;MS. Word;Arg...**
4	Write Introduction	1 day	12/4/18 8:00 AM	12/4/18 5:00 PM		
5	Define Project Organization	2 days	12/5/18 8:00 AM	12/6/18 5:00 PM		
6	Define Project Management Plan	2 days	12/7/18 8:00 AM	12/10/18 5:00 PM	5	Project Libre
7	**Make Requirements document**	56 days	**12/11/18 8:00 AM**	**2/26/19 5:00 PM**	**6**	**Arslan Arif;MS. Word;Arg...**
8	**Make Software Requirement Specification Do**	11 days	**12/11/18 8:00 AM**	**12/25/18 5:00 PM**		
9	Give Introduction and Overview	1 day	12/11/18 8:00 AM	12/11/18 5:00 PM		Arslan Arif;MS. Word
10	Define Scope	1 day	12/12/18 8:00 AM	12/12/18 5:00 PM		
11	Define Purpose and objective	1 day	12/13/18 8:00 AM	12/13/18 5:00 PM		Arslan Arif;MS. Word
12	Identify Specific Requirements	1 day	12/14/18 8:00 AM	12/14/18 5:00 PM		Arslan Arif;MS. Word
13	Explain External Interfaces	1 day	12/17/18 8:00 AM	12/17/18 5:00 PM	12	Arslan Arif;MS. Word
14	Identify Use Cases	3 days	12/18/18 8:00 AM	12/20/18 5:00 PM	13	Arslan Arif;MS. Word
15	Make UseCase Diagram	1 day	12/21/18 8:00 AM	12/21/18 5:00 PM	14	Arslan Arif;Argo UML
16	Define UseCase descriptions	2 days	12/24/18 8:00 AM	12/25/18 5:00 PM	15	Arslan Arif;MS. Word
17	Review and Refine SRS	2 days	12/11/18 8:00 AM	12/12/18 5:00 PM		Muhammad Naqi

Figure 1: Project Plan (1)

18		**Make Software Design Description Document**	**36 days**	**12/13/18 8:00 AM**	**1/31/19 5:00 PM**	**17**	**Arslan Arif;MS. Word;Arg...**
19		Give Introduction and Overview	1 day	12/13/18 8:00 AM	12/13/18 5:00 PM		Arslan Arif;MS. Word
20		Make System Workflow Diagram	4 days	12/13/18 8:00 AM	12/18/18 5:00 PM		Arslan Arif;Argo UML
21		Review and Refine System Workflow Diagram	2 days	12/19/18 8:00 AM	12/20/18 5:00 PM	20	Muhammad Naqi
22		Make System Architectural Design	2 days	1/21/19 8:00 AM	1/22/19 5:00 PM		Arslan Arif;Argo UML
23		Review and Refine Architecture Diagram	1 day	1/23/19 8:00 AM	1/23/19 5:00 PM	22	Muhammad Naqi
24		Make Sequence Diagrams	2 days	1/24/19 8:00 AM	1/25/19 5:00 PM	22;23	Arslan Arif;Argo UML
25		Review and Refine SD	1 day	1/28/19 8:00 AM	1/28/19 5:00 PM	24	Muhammad Naqi
26		Identify Classes	1 day	1/29/19 8:00 AM	1/29/19 5:00 PM	25	Arslan Arif;MS. Word
27		Make Class Diagram	2 days	1/30/19 8:00 AM	1/31/19 5:00 PM	26	Arslan Arif;Argo UML
28		Review and Refine Software Design Description	1 day	12/13/18 8:00 AM	12/13/18 5:00 PM		Muhammad Naqi
29		Make User Interfaces	3 days	12/11/18 8:00 AM	12/13/18 5:00 PM		
30		Give Description of UI	2 days	2/11/19 8:00 AM	2/12/19 5:00 PM		
31		Review and Refine UI	2 days	2/13/19 8:00 AM	2/14/19 5:00 PM	30	Muhammad Naqi
32		**Make Software Test Document**	**4 days**	**2/15/19 8:00 AM**	**2/20/19 5:00 PM**	**31**	
33		Make Test Cases	3 days	2/15/19 8:00 AM	2/19/19 5:00 PM		Argo UML
34		Review and Refine Test Document	1 day	2/20/19 8:00 AM	2/20/19 5:00 PM	33	Muhammad Naqi
35		Review Analysis and Design Document	3 days	2/21/19 8:00 AM	2/25/19 5:00 PM	34	Muhammad Naqi
36		Provide 1st Deliverable	1 day	2/26/19 8:00 AM	2/26/19 5:00 PM	35	
37		Project Implementation	100 days	2/27/19 8:00 AM	7/16/19 5:00 PM	36	Android Studio;Arslan Arif;...

Figure 2: Project Plan (2)

1.4 Summary

In this chapter overall project plan and management of the system is discussed. The project definition, the deliverables and the resources are listed. The next chapter is about the requirements specification of the system.

Chapter – 2

System Requirements Specification

2.1 Introduction

This chapter specifies all the requirements gathered for the development of the proposed system.

2.2 Purpose

This section contains the system requirements specifications for the Smart Gallery Project. It explains the proposed features of the system, the working principles of this system and the general information about the functional and non-functional requirements of the system. This document is intended for both stakeholders and developers who are working on such projects.

2.2.1 Objective & Motivation

Nowadays, people use their smartphones to capture photographs as well as to save/download images from the internet. Users often find it hard to search an image from thousands of images in the gallery. The main objective of this project is to make a smart gallery for android smartphones that could provide a visual search option to the users. The application applies machine learning technique i.e. object detection on images to recognize objects from the images.

2.2.2 Scope of the Project

The system is an android application running on android smartphones. The major inputs, outputs and functions are described below.

Inputs:

Application uses stored images on android media storage so there is only one external user input for the application and that is search keyword(s). The user can enter some object name in the search query as input to search images.

Outputs:

The output of this system is images. Application displays the images as output that contain the object searched by user in search query.

Functions:

The major functions of the system are following.

- Detect objects from the gallery images.
- Save/retrieve detected objects information to/from application's internal storage file.

2.3 Overall Description

The overall description and work flow of the system is described in this section.

2.3.1 Product Perspective

Smart Gallery is a standalone application that shows android media photos to the user and provides search facility using the machine learning.

2.3.2 Product Features

There are following features in the system.

- Search images by objects names.
- Display Images as grid view.
- Display an Image on full screen.
- Share, Delete and Edit an Image.
- Display Image folders.
- Display Image categories.
- Delete Image.
- Share Image.

2.3.3 User Interfaces

User interface is similar to standard android gallery application and contains an additional search option at the top. Detailed UI design is provided in Chapter-4, Software Implementation Document.

2.3.4 Hardware Interfaces

This application uses the following hardware interface(s):

- Android touch screen interface for receiving input from the user.

2.3.5 Software Interfaces

This application uses the following software interfaces:

- Android media storage interface for accessing all the images stored in the device.

2.3.6 Network Interfaces

This application does not use any network interface.

2.3.7 Communication Protocols

The application does not need internet connection so it does not use any communication interface.

2.4 Software Product Functions

This section describes all the functional requirements of the system.

2.4.1 List of Use Cases

There are following use cases in the system.

1. Search Image
2. View Image
3. Share Image
4. Delete Image
5. Process Images
6. View Folders
7. View Categories

2.4.2 Use Case Diagram

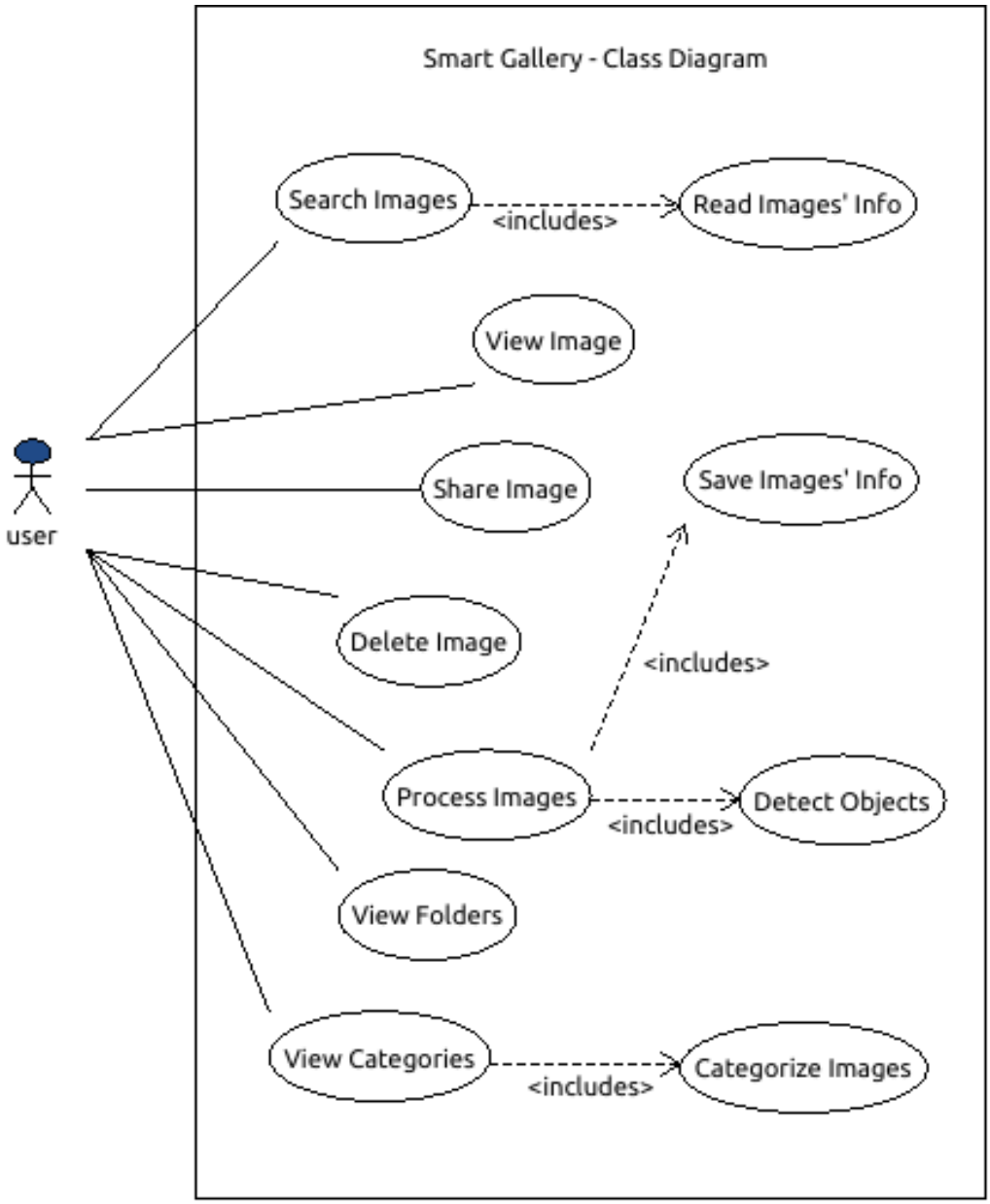

Figure 3: Use Case Diagram

2.4.3 Use Case Description

Use cases are described below in details.

2.4.3.1 Search Image

Table 3: UC-1: Search Image

UC No.	UC-1
UC Name	Search Image
Primary Actor	Android user
Input	Query keyword(s)
Pre-Condition	The image(s) have been processed for object detection.
Post-Condition	Image(s) appears on the screen.
Main Scenario	1. User selects search option. 2. User enters search keyword in the search option. 3. System searches all the images for the keyword and finds relevant images. 4. Relevant image appear on the screen.
Alternative Scenario(s)	3(a). Keywords do not match. 4(a). No results found. No image appears.

2.4.3.2 View Image

Table 4: UC-2: View Image

UC No.	UC-2
UC Name	View Image
Primary Actor	Android user
Input	Touch Input (Image Thumbnail)
Pre-Condition	The image thumbnail is displayed in the images grid.
Post-Condition	Image appears on the full screen.
Main Scenario	1. User taps on image thumbnail. 2. Relevant image appear on the screen.

2.4.3.3 Share Image

Table 5: UC-3: Share Image

UC No.	UC-3
UC Name	Share Image
Primary Actor	Android user
Input	Image
Pre-Condition	Image is opened.
Post-Condition	Image is shared on selected platform.
Main Scenario	1. User selects share option. 2. List of sharing apps appears. 3. User Selects an app. 4. Image is shared with app.
Alternative Scenario(s)	5. 2(a). No photo sharing app is available. 6. 4(a). App does not accept image.

2.4.3.4 Delete Image

Table 6: UC-4: Delete Image

UC No.	UC-4
UC Name	Delete Image
Primary Actor	Android user
Input	Image
Pre-Condition	Image is opened.
Post-Condition	Image is deleted from the system storage and disappears from the Smart Gallery.
Main Scenario	1. User selects delete option. 2. System asks for delete confirmation. 3. User confirms to delete.

	4. Image is deleted from the storage.
Alternative Scenario(s)	5. 3(a). User cancels delete.

2.4.3.5 Process Images

Table 7: UC-5: Process Images

UC No.	UC-5
UC Name	Process Images
Primary Actor	Android user
Input	User Command (Process)
Pre-Condition	The images contain some known objects. (Known objects refer to the objects that our system recognizes)
Post-Condition	Images are processed and information is saved in a file.
Main Scenario	1. User selects process option. 2. System starts detection. 3. All images are processed and known objects are detected from all the images. 4. Results are saved in the file.
Alternative Scenario(s)	3(a). Processing is stopped by user.

2.4.3.6 View Folders

Table 8: UC-6: View Folders

UC No.	UC-6
UC Name	View Folders
Primary Actor	Android user
Input	User Command (touch event)
Pre-Condition	There are some folders with images in the internal storage.
Post-Condition	Image Folders appear on the screen.
Main Scenario	1. User selects folders option. 2. Image folders appear on the screen.

2.4.3.7 View Categories

Table 9: UC-7: View Categories

UC No.	UC-7
UC Name	View Categories
Primary Actor	Android user
Input	User Command (touch event)
Pre-Condition	There are some images with in at least one category.
Post-Condition	Image Categories appear on the screen.
Main Scenario	3. User selects categories option. 4. Image categories appear on the screen.

2.5 Software System Attributes

Software system attributes refer to the qualitative features of the system. These attributes are described below.

2.5.1 Availability

Application is available 24/7 to the android user as it is a fully installed app on android smartphone.

2.5.2 Security

There are the following terms and conditions on security and privacy of user data:

- The application needs access to media storage of the android device as it's is the prime requirement. However it will never share any media with any 3rd party without the permission of user.
- Photos of users will never be transmitted over any network as application does not require internet connection and works completely offline.

2.5.3 Portability

This application can be installed on any android smartphone having the android version 5.0 Lollipop (API level 21) or higher.

2.5.4 Performance

The main requirement of application is searching images. The search option provides search results in real time. The detection process has a speed of 5 images per second. It needs to be called only when new images are added or for the first time after installation, to process all the images.

2.6 Database Requirements

There is no requirement of any database in this project as the images are already stored and managed by android OS. However internal file storage is used to store the processed information of images in a file.

2.7 Summary

All the basic requirements are collected by analyzing the scope, objective and available resources for the development of this application. Requirements gathering is the most important phase of a software system and these requirements have been refined many times before the actual implementation. The next chapter is about the conceptual and technical design of the system.

Chapter – 3

Software Design Document

3.1 Introduction

Software Design Description (SDD) is a document which provides the complete description of the design of the software to be developed before the actual development. The SDD document describes the system architecture design in detail and also provides a complete description of the different components. It also describes how the different components will communicate with each other. The SDD document also contains the architecture diagram, sequence diagrams and class diagram.

3.2 Domain Model

Domain model represents the real situation conceptual classes, not of software objects but the real life objects. Domain model is often considered as business model of a software. The conceptual class diagram or domain model of our system is represented below.

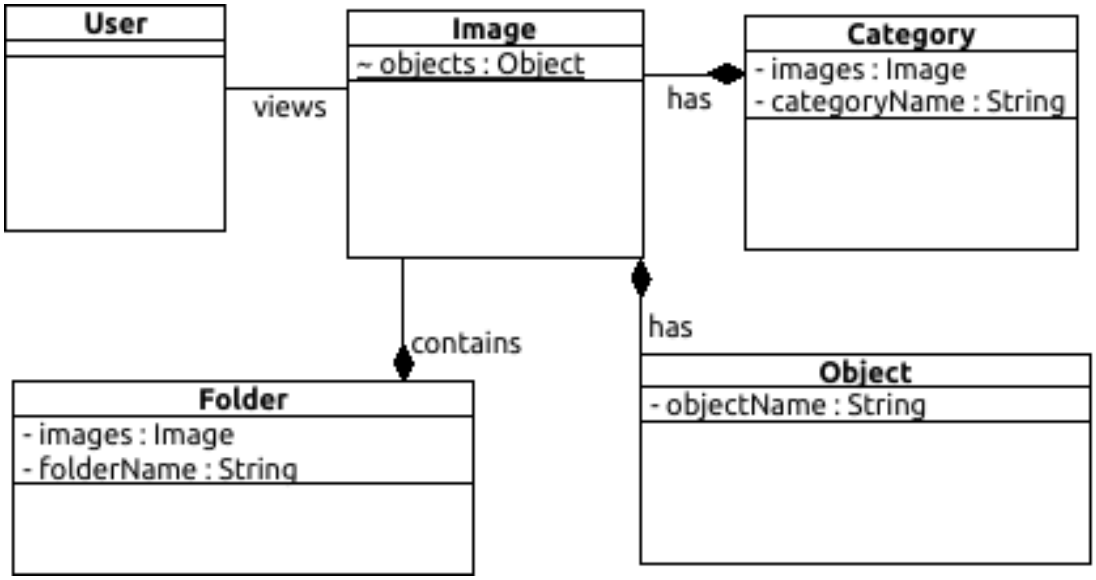

Figure 9: Domain Model

3.3 System Architectural Design

Software architecture design is the process of defining a structured solution that meets all of the technical and operational requirements, while optimizing common quality attributes such as performance, security, and manageability. Architectural design is the resolution of the requirements in the design of the software, the hardware and networking, operations, policies and so forth.

3.3.1 Chosen System Architecture

We have used 3-Tier architecture for our system. As android supports MVC architecture which is a 3-Tier Architecture so we have decided to use MVC as architecture for our system. Activities/ Fragments act as the controller and Models used for data and persistence. With this approach, Activities are in charge of processing the data and updating the views. Activities act like a controller in MVC but with some extra responsibilities that should be part of the view.

3.3.1.1 Presentation Layer

It represents the user interface. The user will interact with the application using the user interface. It is represented by View in MVC architecture. It is typically an Activity which contains code related to user interface.

3.3.1.2 Application Layer

This layer contains the actual logic implementation. The actual application logic of the whole system is implemented in this layer.

3.3.1.3 Data Layer

This layer is responsible for accessing data and providing it to application layer. In our system there are 3 sources of data, the images from android storage, the trained models and the internal file that contains the information about the image.

3.3.2 Architectural Diagram

The 3-Tier architecture of our system is represented by the following architectural diagram.

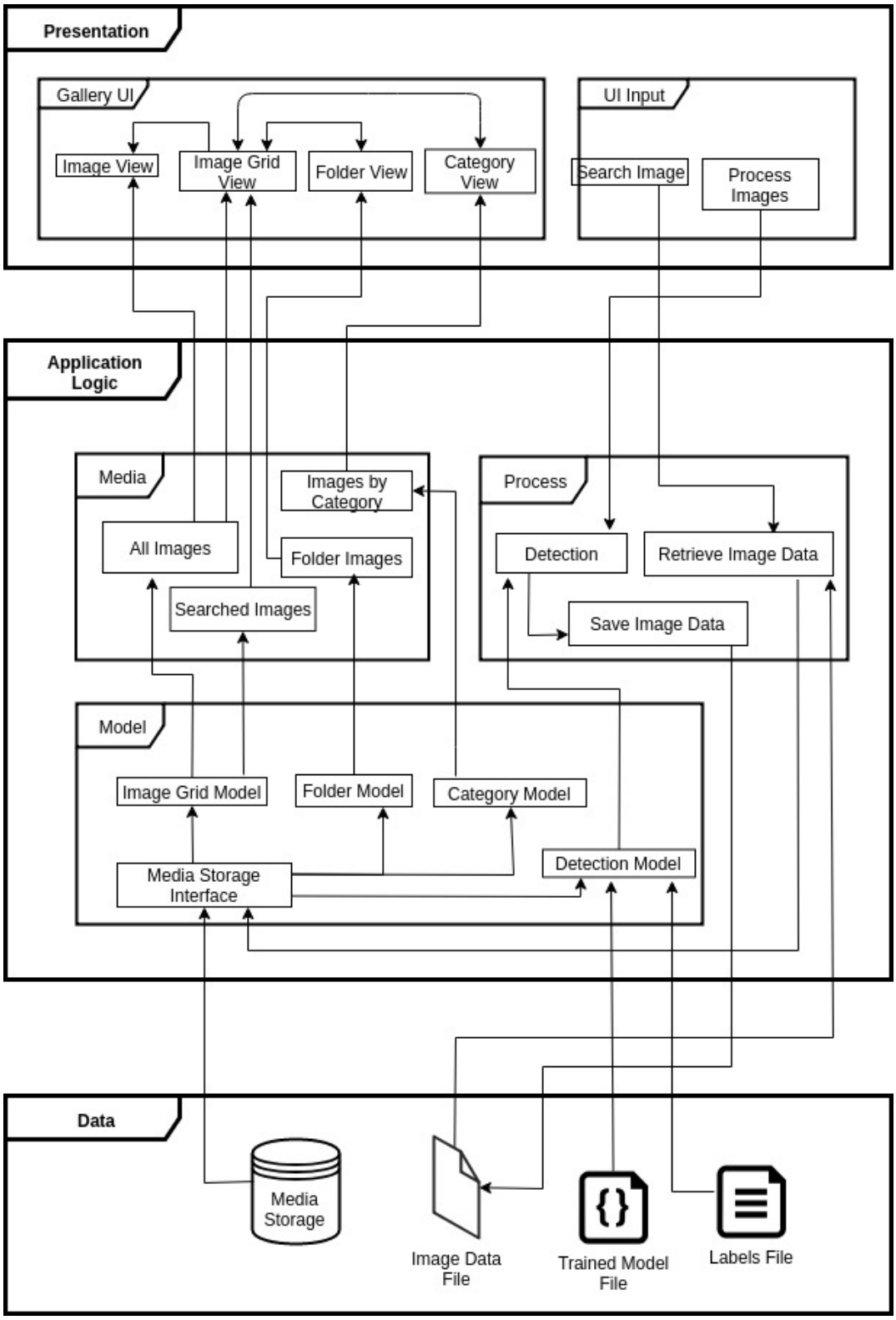

Figure 10: Architectural Diagram

3.3.3 Discussion of Alternative Designs

An alternative architectural design that could be used for the development of this system is 2-Tier architecture in which data layer is integrated into application layer. Another 3-Tier architecture MVP (Model View Presenter) can also be used for this system.

3.4 System Workflow Diagram

Workflow diagram represents the complete workflow of this system i.e. how the different components of the system interact with each other.

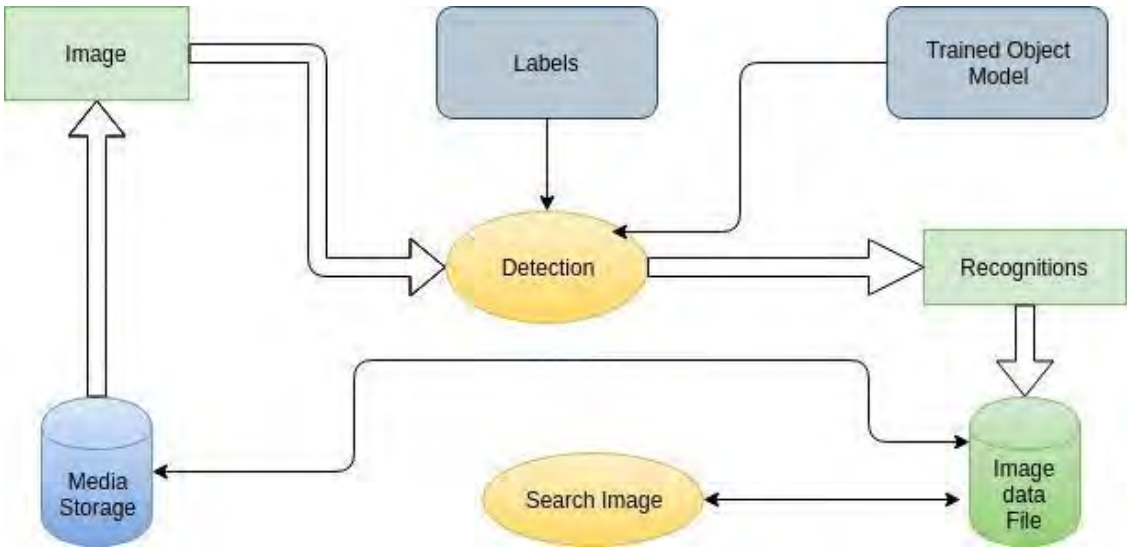

Figure 11: System Work Flow

3.5 Sequence Diagrams

Sequence diagrams in the UML are primarily used to model the interactions between the actors and the objects in a system and the interactions between the objects themselves. There are two sequence diagrams in our system that demonstrate the overall functionality of system and interaction between different objects.

3.5.1 Search Image Sequence Diagram

This represents the actual search functionality of the system and demonstrates how the image is searched.

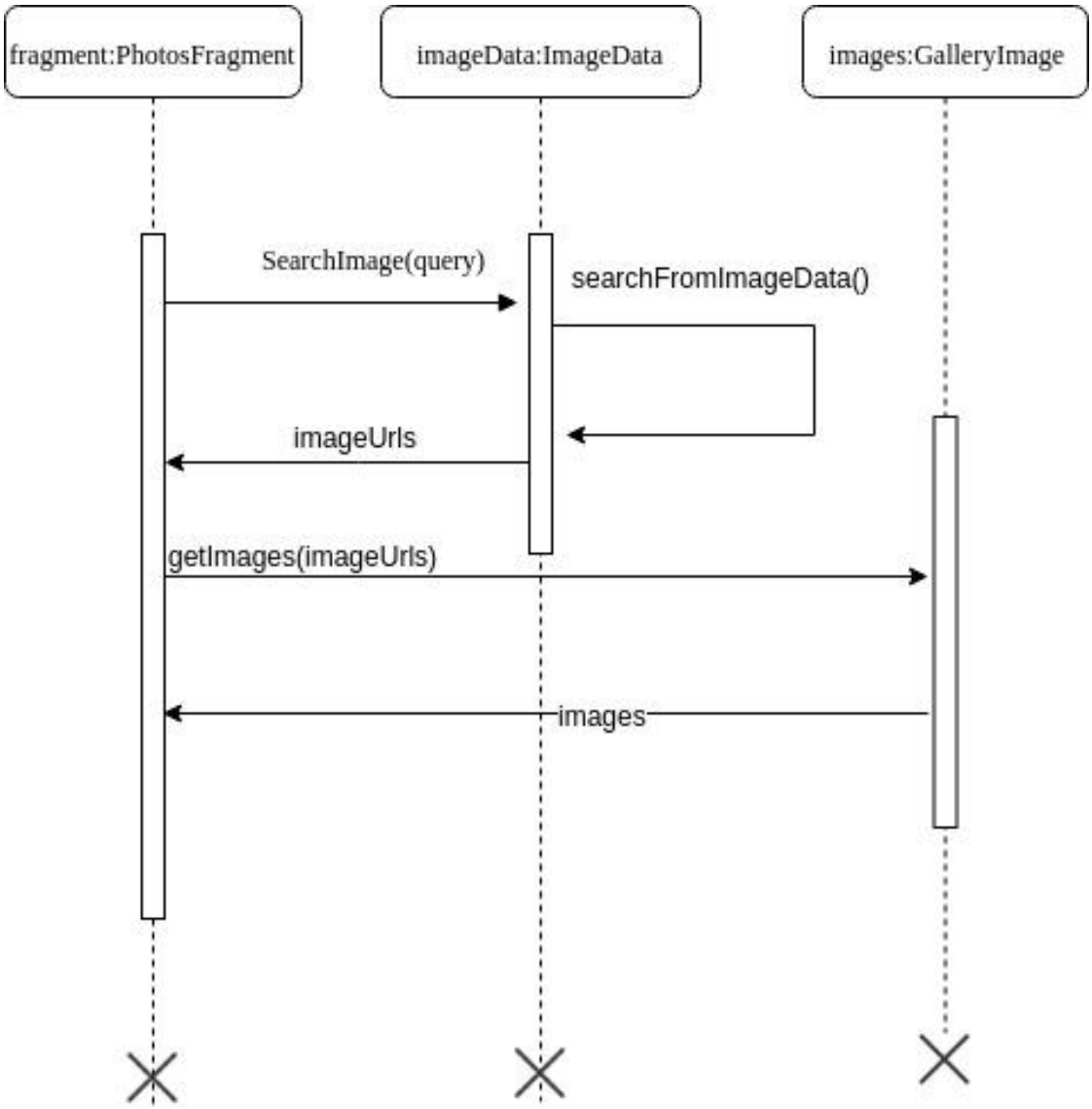

Figure 18: Search Image SD

3.5.2 Process Images Sequence Diagram

This actually represents how the system processes the images stored in the phone storage.

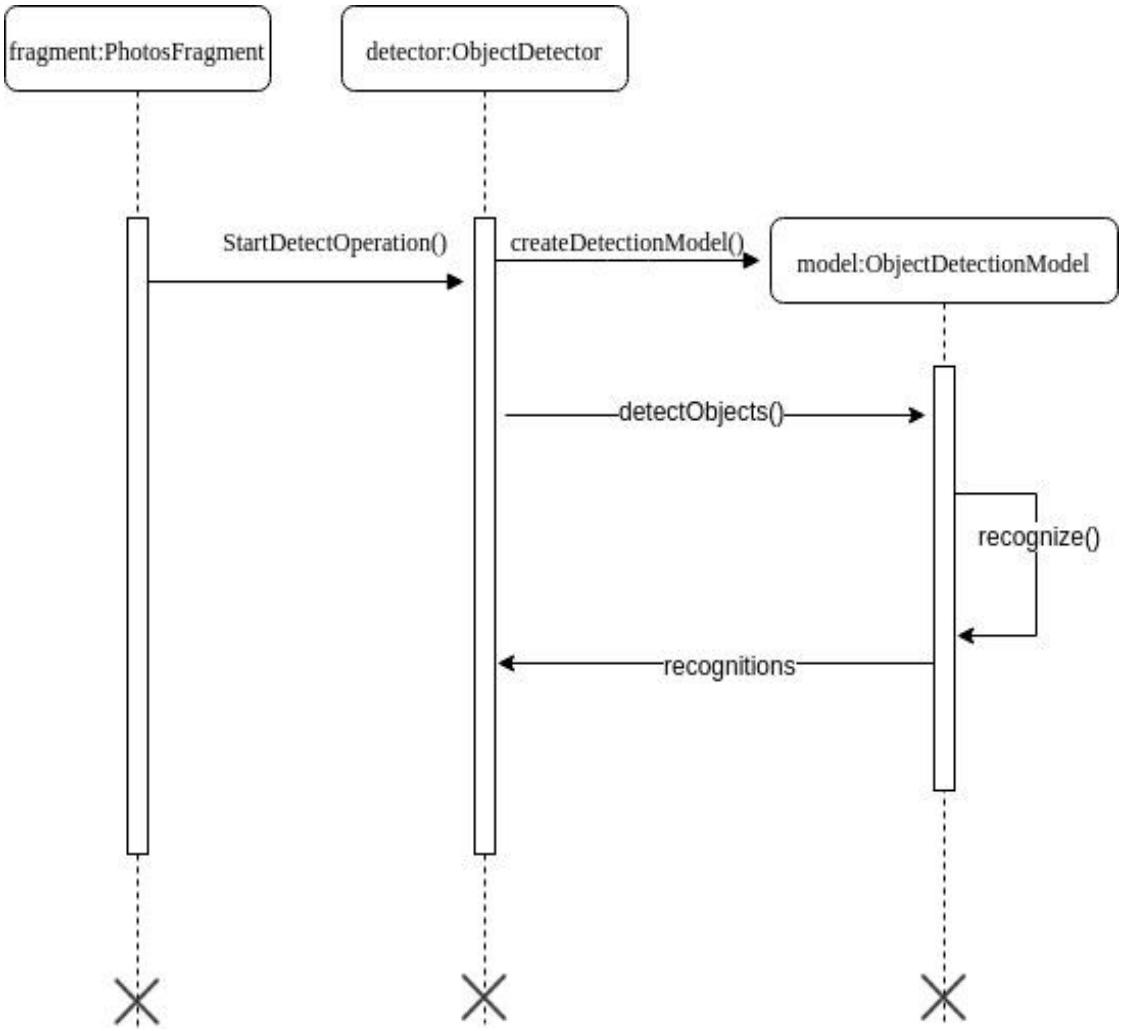

Figure 19: Capture Image SD

3.6 Design Class Diagram

This is the representation of actual software classes of the system. A class diagram represents both, the structural features i.e. attributes and the behavioral features i.e. functions of the software classes. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

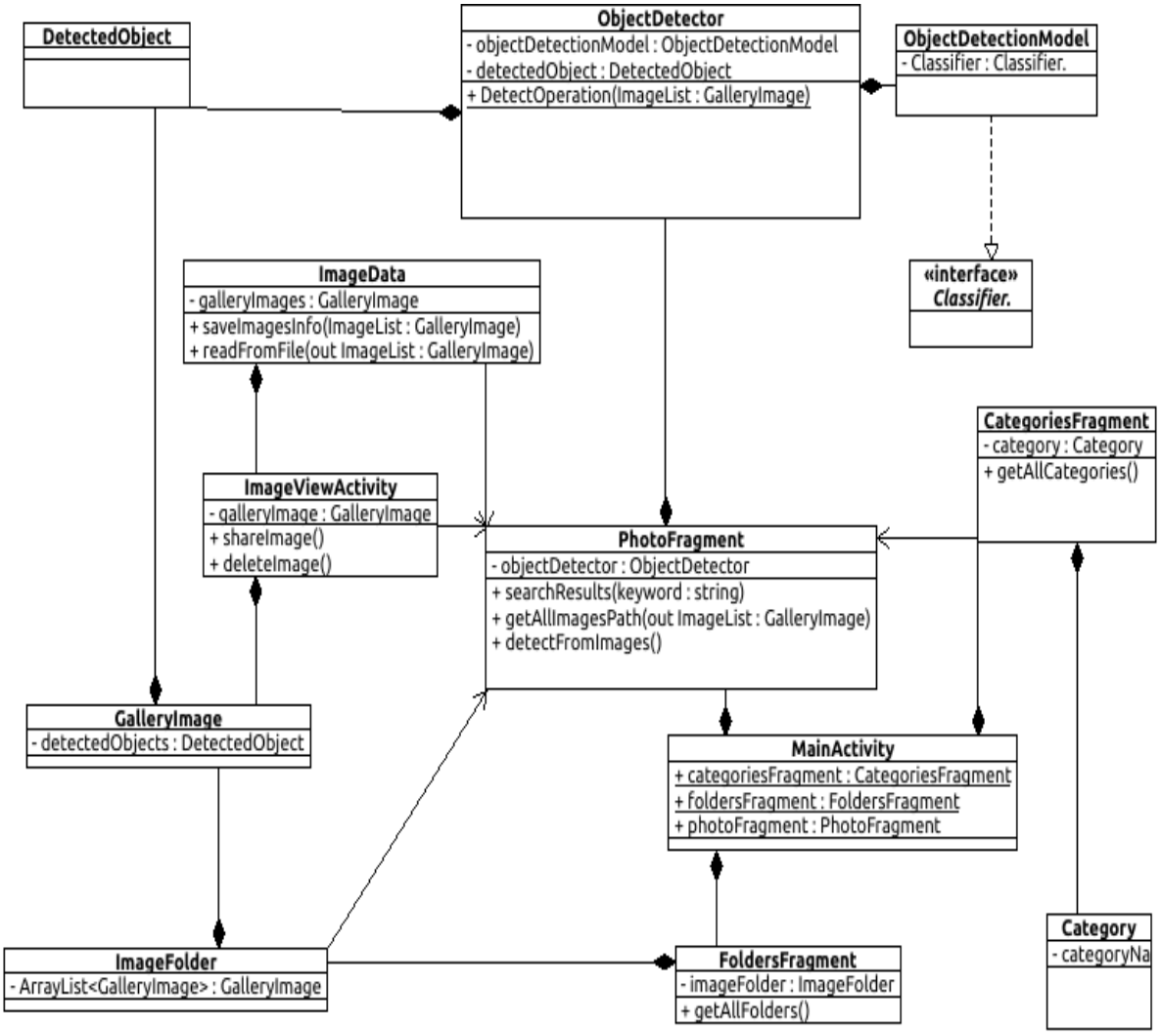

Figure 21: Design Class Diagram

3.7 Summary

Logical and proposed design of the system is described in this. The workflow and the proposed architecture of our system is also described. The details about the specific techniques and the technologies used in this project are explained in the next chapter i.e. Software Implementation Document. The detailed workflow of the machine learning algorithms is also described in this phase.

Chapter – 4

Software Implementation Document

4.1 Introduction to Machine Learning

Machine learning is an application of artificial intelligence (AI). It enables systems to learn things automatically and improve without humans' help. Machine learning is the talk of the town these days and the world is adapting AI and Machine learning. It can be used to solve a lot of real life problems like object classification, pattern recognition, place recognition, face recognition, speech recognition etc. Some machine learning techniques used in our projects are briefly described below. Our project uses object detection to fulfill its actual requirement i.e. visual search so it is important to learn what is object detection.

4.2 Object Detection

Object detection is a machine learning technique that is used to detect objects from an image. The object detection algorithm identifies and locates all instances of objects in an image from a known set of object classes. The algorithm takes an image as input and outputs the category that the object belongs to, along with a confidence score that it belongs to the category. The algorithm also predicts the object's location and scale..

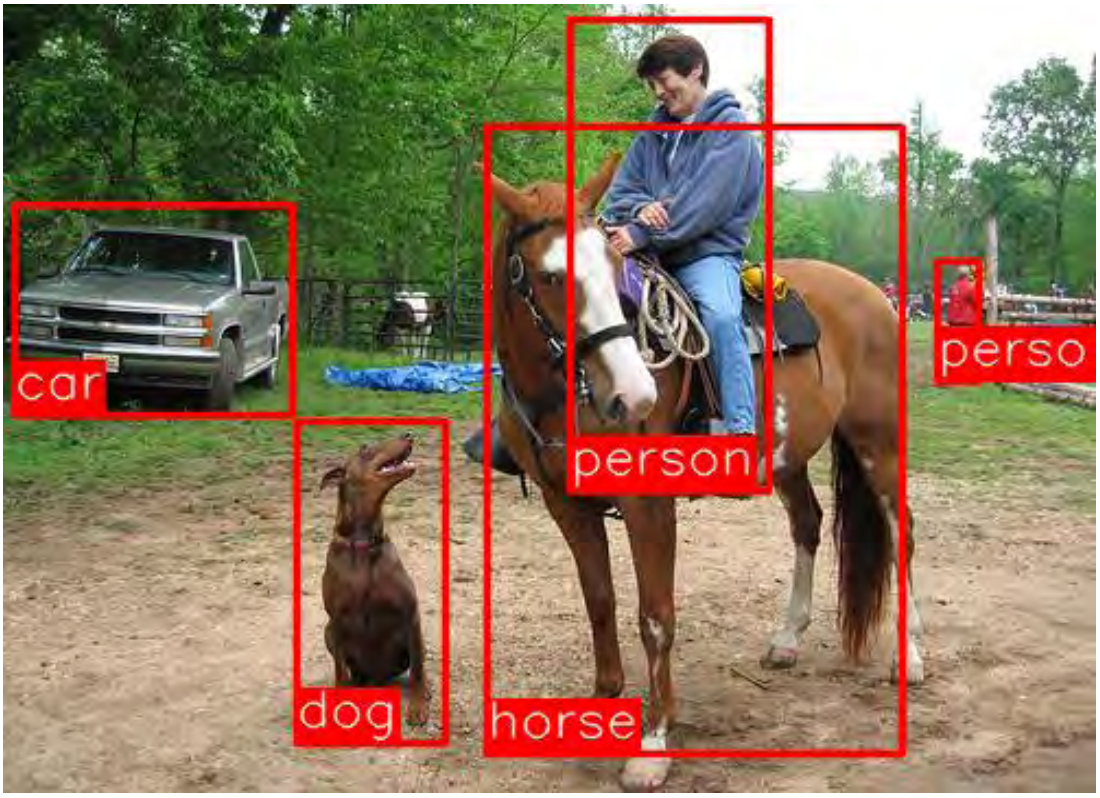

Figure 4: Object Detection Example [18]

4.2.1 Object Classification versus Object Detection

Classification, in the context of images, is the process that categorizes recognition of the dominant content in an image. The dominant content gets the best score irrespective of the location, orientation or size of that content. An example of a classification problem is when given an image of a cat or anything else, we want to know what dominant content is in the image. Thus a classification system should always label that image as “cat” no matter where the cat is in the image so long as the cat is the dominant content there. If the cat is no longer the dominant content, the system should change the label of the image to the next dominant content. In simple terms classification is concerned about what is in the scene and not where it is.

Sometimes we need to know how many cats or how many known objects are in the scene and where they are in a scene. The process of recovering the location of the objects is called localization. Detection is thus a process that involves both classification and localization. An object detection system is tasked to categorize and locate all known content in an image. For example, a pedestrian detection system must recognize pedestrians and provide the locations of each one of those recognized pedestrians.

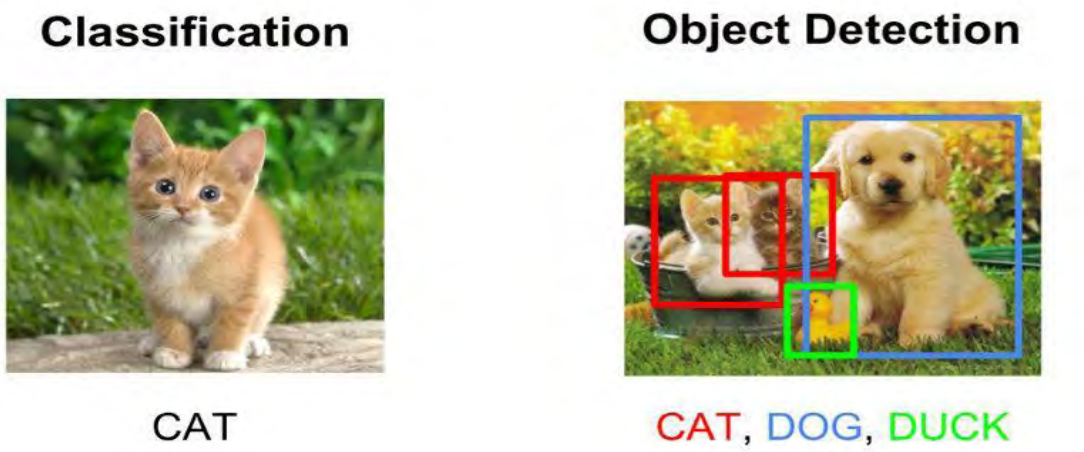

Figure 5: Object Classification and Detection [17]

4.3 Project Management Techniques

Project implementation is divided into two major parts. The first part is the machine learning part which is about the training of the model that is ultimately used for detection. While the second part is to implement android gallery application in Android Studio, implement all the business logic for the gallery application and then use the trained model to implement object detection logic on images. Some technical background of the tools and techniques used in this project is given below.

4.3.1 Neural Networks and Deep Learning

Neural networks are a set of algorithms that are designed to recognize patterns. They work like human brain neurons. The patterns they recognize are numerical, contained in vectors.

Deep learning is the name we use for multilayer neural networks that is networks composed of several layers. The layers are made of nodes. A node is just a place where computation happens. It is similar to a neuron in the human brain which fires when it encounters sufficient stimuli. A node combines input from the data with a set of coefficients, or weights that either amplify or dampen that input, thereby assigning significance to inputs with regard to the task the algorithm is trying to learn; e.g. which input is most helpful is classifying data without error? These input-weight products are summed and then the sum is passed through a node's so-called activation function, to determine whether and to what extent that signal should progress further through the network to affect the ultimate outcome, say, an act of classification. If the signals passes through, the neuron has been "activated."

A node layer is a row of those neuron-like switches that turn on or off as the input is fed through the net. Each layer's output is simultaneously the subsequent layer's input, starting from an initial input layer receiving your data.

Deep-learning networks are distinguished from the more common hidden-layer neural networks by their depth; that is, the number of node layers through which data must pass in a multistep process of pattern recognition.

In deep-learning networks, each layer of nodes trains on a distinct set of features based on the previous layer's output. The further we advance into the neural network, the

more complex the features our nodes can recognize, since they aggregate and recombine.

4.3.2 Tensorflow

Tensorflow is an open source platform for machine learning. It has a flexible set of tools libraries and resources. Tensorflow APIs are based on the Keras API standard for defining and training neural networks. Tensorflow provides a very easy way to train a new model. It also has a lot of pre trained models.

4.3.3 Tensorflow Lite

Tensorflow Lite is a set of tools to help mobile application developers in running Tensorflow models on mobile devices. It enables machine learning in mobile phones with low latency and good performance.

Tensorflow Lite consists of two main components:

- The Tensorflow Lite interpreter, which runs optimized models on different hardware types, including mobile phones.
- The Tensorflow Lite converter, which converts Tensorflow models into an efficient form for use by the interpreter, and can introduce optimizations to improve performance.

4.3.4 Android Gallery Application

Apart from machine learning part, Android gallery application itself is a big project. Gallery application is developed in Android Studio using Java and Kotlin.

4.4 Development Workflow

Tensorflow is used to train object detection model and then converting it to Tensorflow Lite model. The model is then imported to android application and used for the detection. The complete implementation process is divided into following tasks.

4.4.1 Pick a Model

The first step is to pick a model. We must start with a regular Tensorflow model and then convert the model to Tensorflow Lite. The model used in this project is 'COCO MobileNet-SSD-V1' and it is re-trained through transfer learning.

4.4.2 Retrain Model using Transfer Learning

Transfer learning allows us to take a trained model and re-train it to perform another task. For example, an image detection model could be retrained to recognize new categories of image. Re-training takes less time and requires less data than training a model from scratch. We can use transfer learning to customize pre-trained models to our application.

4.4.3 Convert to Tensorflow Lite

Tensorflow Lite is designed to execute models efficiently on mobile and other embedded devices with limited compute and memory resources. Tensorflow models must be converted into Tensorflow Lite format used by Tensorflow Lite. Tensorflow model can be converted to Tensorflow Lite using Tensorflow Lite Converter.

4.4.4 Import to Android Application

The converted Tensorflow Lite model is then imported to android application along with labels for the objects. This converted model is used for the detection in the Smart gallery application.

4.4.5 Implement Application Logic

Application logic of gallery application is implemented in Java and Kotlin. The Smart Gallery application provides all the basic features of android photo gallery application.

4.4.6 Use imported model to perform detection

The imported model is coupled with application logic to perform detection on all gallery images and recognize objects. This information is saved in an internal file on storage. The information is used to provide visual search feature.

4.5 Challenges in Development

The idea seems pretty simple but a lot of challenges are faced throughout the implementation of this project.

4.5.1 Training Time

Since the training process is computationally expensive so it is a big challenge to perform transfer learning on a pre-trained model with limited available resources. The training process takes a lot of time on normal PCs.

4.5.2 Accuracy

SSD, Single Shot Detection neural network is used which is fast but less accurate. So achieving both speed and accuracy is a big task.

4.5.3 Complexity of Application

The application logic is very complex as it is an android gallery with basic gallery features and additional visual search option. Not much help is available on gallery project so the whole implementation of gallery app from scratch is a really big challenge. Moreover, the Tensorflow Lite is a very new tool and it is not fully matured yet. There is no help available on implementing it in gallery app so smoothly coupling the Tensorflow Lite model with the gallery application logic is one of the major challenges.

4.6 System UI

The actual user interface of the application is presented below.

4.6.1 Photos Grid

This is the main screen where images are displayed like any smartphone gallery app.

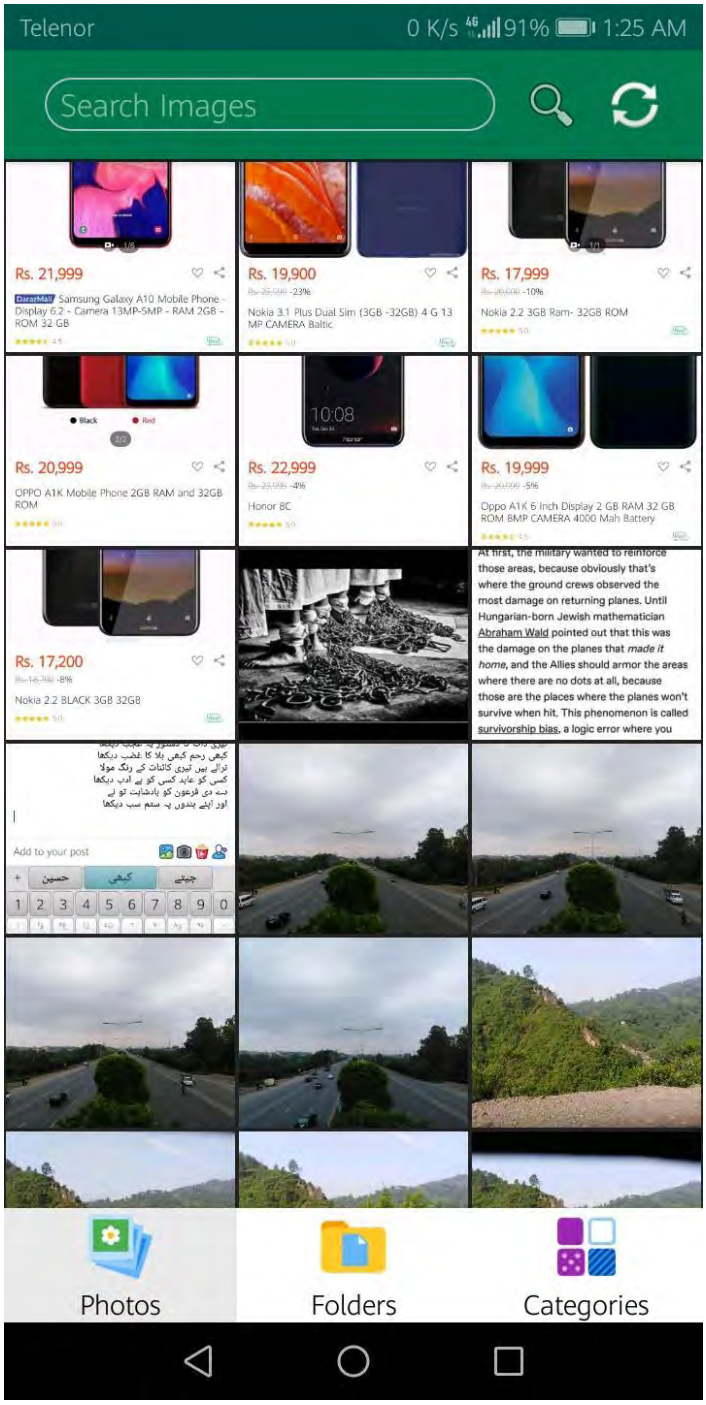

Figure 12: Photos Grid UI

4.6.2 Search Image

This is the search interface where user enters search keywords to search any image.

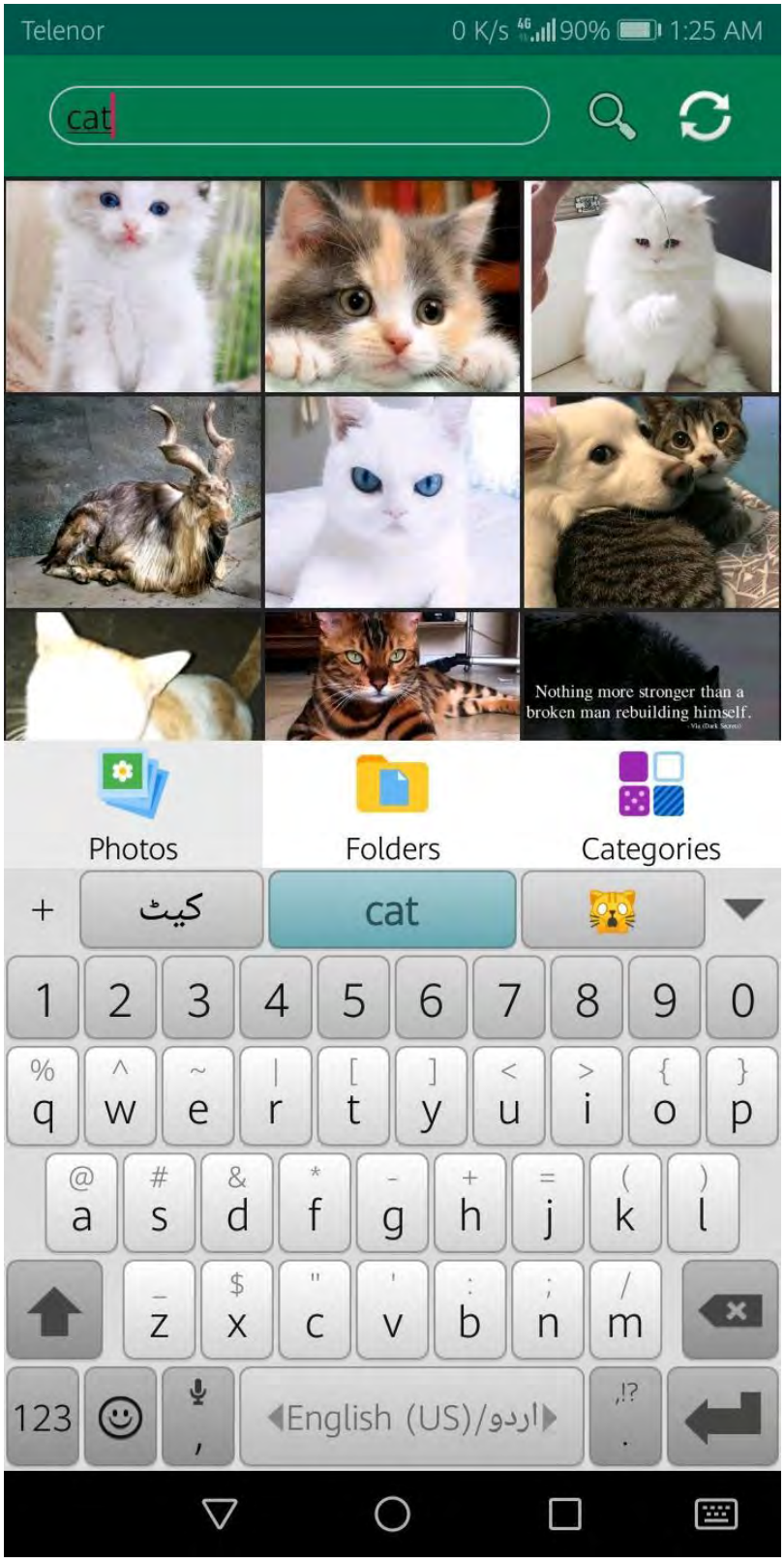

Figure 13: Search Image UI

4.6.3 Folders List

This is the list view of image folders.

Figure 14: Folder List UI

4.6.4 Categories List

This displays the list of image categories.

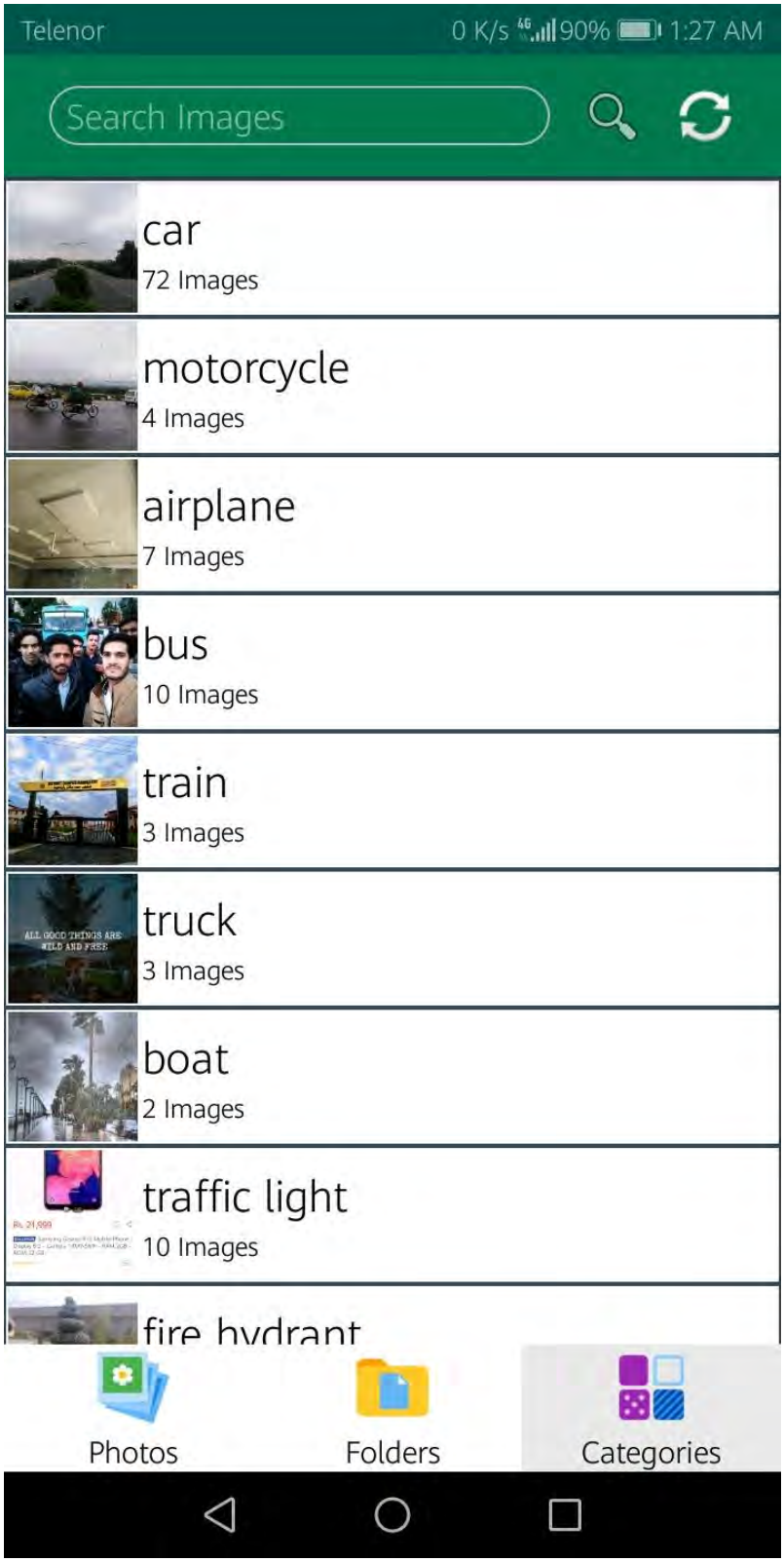

Figure 15: Categories Image UI

4.6.5 View Image

This is the full view of an image.

Figure 16: View Image UI

4.6.6 Share Image

This represents the pop up that appears when user shares image using any external media.

Figure 17: Share Image UI

4.7 Summary

The implementation details of the project are briefly described in this chapter. Some basic details about the deep neural networks and Tensorflow have been given. More details of neural networks and the type of neural networks can be added but since deep learning is a very vast and advanced field so it would be out of scope of this document to discuss all the details about deep learning neural networks. Next chapter is about the Software Testing.

Chapter – 5

Software Test Document

5.1 Introduction

Software testing is a verification and validation process which guides the tester through a sequence of steps to validate whether a software application is free of bugs and working as required by the end user. Software testing is a very important process that should be done during the development process because it is very useful to assess the quality of the product.

5.2 Test Strategy

Test strategy describes the test approach which is used for to test the system. It focuses that how the risks will be mitigated during the development of the software. We perform unit test on each component of the project. After unit testing we perform the integration testing. And then the Acceptance testing is performed.

5.3 Test Plan

A test plan is a detailed document that outlines the test strategy, testing objectives, resources required for testing, test schedule and test deliverables. The test plan serves as a blueprint to conduct software testing activities. It also describes that which features to be tested and which features are not to be tested.

5.3.1 Features to be tested

Following features will be tested from the system.

- Recognize known objects.
- Detecting objects from new Images.
- Categorize Images.

5.3.2 Testing Technique and Environment

For the testing purpose, different android smartphones with API level 21 to API level 30 are used. Input, behavior and output can be tested on any android smartphone having android API level 21 or higher.

5.4 Acceptance Test

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test

is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

5.5 Test Cases

A test case, in software engineering, is a set of conditions under which a tester will determine whether an application, software system or one of its features is working properly or not. Actual testing will be conducted on the basis of test cases.

5.5.1 Search Object Test Case

Table 11: TC-1: Search Object

ID	TC-1
Description	The test case is conducted to check that the system gives accurate search results on processed images.
Tested Module	Image Searching Module
Tester	User
Setup	1. The application contains pre-trained object model. 2. Images have already been processed on trained objects model. 3. There is at least one image in the gallery with that contains the object we are searching.
Input	1. Cat (query keyword)
Expected Result	All images that contains a cat should appear in the search result. All images that appear in the search results must contain a cat.

5.5.2 Detect Objects Test Case

Table 12: TC-2: Detect Objects

ID	TC-2
Description	The test case is conducted to check that the system performs detection operation smoothly on all the images.
Tested Module	Object Detection Module
Tester	User
Setup	1. The application contains trained model. 2. Some images with at least one known objects are present in the gallery.
Input	1. Images, Process Command.
Expected Result	Application starts processing the images and after the processing is completed, the TC-1 is satisfied.

5.5.3 Categorize Images Test Case

Table 13: TC-3: Categorize Images

ID	TC-3
Description	The test case is conducted to check that images in the gallery are categorized in object categories like cat, person, car etc.
Tested Module	Categorization Module
Tester	User
Setup	1. All images have been processed. 3. There are images which contain objects of some of the known categories.
Input	1. Show Categories Command
Expected Result	Different known categories are listed and each category has some image(s). All images in each category actually fall under the respective category.

5.6 Summary

Testing is one of the most important part of software development life cycle and a good text cannot be conducted without good test cases. This chapter provides all the essentials for testing the product. The overall conclusion and future enhancements are described in the next, the last, chapter.

Chapter – 6

Conclusion and Future Enhancements

6.1 Conclusion

The whole project development took lot of effort and there were a lot of challenges but finally the project has been completed within the given period of time. The end product is very user friendly and interactive and provides a very interactive feature to the users. User can search a specific image from thousands of images if he only remembers what was in the image. Though project took a lot of time and effort and it has gone through a lot of development complexities but the end product is very simple and attractive for android users.

The application uses a 'COCO-MobileNet-SSD-V1' model trained using Tensorflow library. The model is then converted to Tensorflow Lite. The Tensorflow Lite model is then imported to android application. Photo gallery application logic is implemented. The application then uses the trained model to perform object detection and recognize objects from the images. Thus the user can find any images by just searching an object name like cat, person, car etc.

6.2 Future Enhancements

A lot of improvements can be made in this application in the future enhancement. The application can be enhanced to a much more user attractive and useful application. So of the future enhancement plans are given below.

6.2.1 Add more Categories

Smart Gallery can recognize 80+ categories of objects from the images. In future, the number of categories will be increased to 150+ through transfer learning

6.2.2 Improve Accuracy

Accuracy can be improved in future with a more powerful model and better techniques.

6.2.3 Add character recognition option (OCR)

Smart Gallery can't read text from the images. In future, the optical character recognizer will also be added to the application which will enable the application to recognize images with text as well.

6.2.4 Add Facial Recognition

Smart Gallery cannot categorize faces in the current sprint but a facial recognition feature can be added in the future which will make the app more powerful and user interactive.

References:

1. https://ieeexplore.ieee.org00/document/278253 (Dec-23-2018)
2. https://github.com/CankayaUniversity/ceng-407-408-2017-2018-project-face-and-iris-recognition/wiki/Software-Requirements-Specification (Dec-23-2018)
3. https://github.com/CankayaUniversity/ceng-407-408-2017-2018-project-blood-vessel-segmentation/wiki/Software-Requirements-Specification (Dec-23-2018)
4. https://developer.android.com (Dec-25-2018)
5. https://medium.com/@williamkoehrsen/object-recognition-with-googles-convolutional-neural-networks-2fe65657ff90 (Dec-02-2018)
6. https://docs.aws.amazon.com/sagemaker/latest/dg/algo-object-detection-tech-notes.html (Dec-21-2018)
7. https://us.norton.com/internetsecurity-iot-how-facial-recognition-software-works.html (Jan-05-2019)
8. https://www.quora.com/What-is-the-difference-between-detection-and-classification-in-computer-vision (Jan-05-2019)
9. https://www.quora.com/How-does-Optical-Character-Recognition-OCR-technology-work(Jan-05-2019)
10. Book: Artificial Intelligence, A Modern Approach, Third Edition by Stuart J. Russell and Peter Norvig. (Dec-21-2018)
11. Book: Software Engineering: A Practitioner's Approach, Seventh Edition by Roger S. Pressman. (Dec-21-2018)
12. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram (Feb-8-2019)
13. https://www.tutorialspoint.com/uml/uml_class_diagram.htm (Feb-8-2019)
14. https://developer.android.com/guide/topics/providers/document-provider(Feb-09-2019)
15. https://www.uml-diagrams.org/examples/hospital-domain-diagram.html(Feb-09-2019)
16. https://blog.rapidapi.com/top-facial-recognition-apis/ (Feb-8-2019)
17. https://chainercv.readthedocs.io/en/stable/tutorial/detection.html (Feb-09-2019)
18. https://www.tensorflow.org
19. https://www.tensorflow.org/lite/guide