**Object and Color detection**

**Aqeela Sultan**

**Ayesha Nawaz**



DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD,

ATTOCK CAMPUS – PAKISTAN

SESSION 2017-2021

**Object and Color detection**

*Undertaken by*

**Ayesha Nawaz**

CIIT/FA17-BCS-088/ATK

**Aqeela Sultan**

CIIT/FA17-BCS-072/ATK

*Supervised By:*

**Miss Sadia Ejaz**

A DISSERTATION SUBMITTED AS A PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF COMPUTER SCIENCE

DEPARTMENT OF COMPUTER SCIENCES

COMSATS UNIVERSITY ISLAMABAD,

ATTOCK CAMPUS – PAKISTAN

SESSION 2017-2021

UNDERTAKING

We certify that this is my/our own work. The work has not, in whole or in part, been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged. If this statement is untrue, we acknowledge that we will have committed an assessment offence and shall be liable to punishable action under the plagiarism rules of HEC.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Aqeela Sultan Ayesha Nawaz

Ciit/ FA17-BCS-072/ATK CIIT/ FA17-BCs-088/ATK

Dated: \_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_

**FINAL APPROVAL**

Certified that we have read this project report submitted by Miss. Aqeela Sultan(CIIT/FA17-BCS-072/ATK) and Miss. Ayesha Nawaz(CIIT/FA17-BCS-088/ATK) and it is, in our judgment, of sufficient standard to warrant its acceptance by Department of Computer Science, University of Islamabad, Attock Cantt, for the (BS/MSc degree) in Computer Science.

***Committee:***

1. External Examiner \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Examiner Name)

Designation

University Name

2. Supervisor \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Supervisor Name)

3. Chairperson \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Chairperson Name)

4. Dean/Director \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Dean/Director Name)

**DEDICATION**

*This project would not have become possible without the kind support and encouragement of many individuals. We would like to thanks all of them one by one but foremost, all praises and admiration goes to Almighty ALLAH and his beloved Prophet Hazrat Muhammad (Peace Be upon Him). We want to show our deepest gratitude to our parents, siblings, for their encouragement and prayers of day and night, for motivating us in doing our best but most importantly, giving us the education, we have and that we are proud of. We owe our Heartiest thanks to our supervisor Miss Sadia Ejaz for always keeping us in path to write a good dissertation, for giving me the right tools and experience and for always being available to answer to our questions or doubts.*

*.*

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His bound less knowledge by virtue of which we were able to accomplish this challenging task. We are greatly indebted to our project supervisor “Miss Sadia Ejaz”. Without her personal supervision, advice and valuable guidance, completion of this project would have been doubtful. We are deeply indebted to them for their encouragement and continual help during this work. And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty and hard work.

Aqeela Sultan Ayesha Nawaz

--------------------------- ---------------------------

**PROJECT BRIEF**

PROJECT NAME OBJECT AND COLOR DETECTION

ORGANIZATION NAME COMSATS UIVERSITY ISLAMABAD, ATTOCK CAMPUS

OBJECTIVE TO HELP USER TO OVERCOME VISUAL DISABILITIES, LEARNING PERCEPTIVE, TO SAVE EFFORTS, RESOURCES AND MONEY

UNDERTAKEN BY AQEELA SULTAN, AYESHA NAWAZ

SUPERVISED BY MISS SADIA EJAZ

COMPUTER SCIENCE

COMSATS UIVERSITY ISLAMABAD, ATTOCK CAMPUS

Started On FALL 2017

Completed On FALL 2021

COMPUTER USED HP (ELITEBOOK 8470P)

SOURCE LANGUAGE PYTHON, JAVA, XML

OPERATING SYSTEM WINDOW 10

TOOLS USED ANDROID STUDIO 4.0.1, PYCHARM 2021.1,

ANACONDA 2021.11

**ABSTRACT**

A model based on scalable object detection using deep neural networks approaches employ Convolutional neural networks (CNNs) techniques to identify the object in images. Object detection has the same effect as other computer vision techniques, such as image recognition and image separation, allowing us to understand and analyze scenes in images. Detecting objects, with the use of machine learning techniques, can be used in more than one business as well as social programs. The key intention of this application is to dispose of the need for dedicated gadgets, different wearable devices, and also multiple use of android application.

This application discussed the classification of objects by using computer vision techniques. By going through past techniques in this regard, we analyzed the working of all machine learning algorithms. These are the following steps of deep learning to carry out our project i.e. preprocessing and classification. CIFAR10 dataset is used to train the CNN model. Saved image in the gallery or run-time capture image is input to the application, it is first preprocessed by operation and then fed for classification. Then object class is classified and also the color of it in output. The purpose of this project is to improve the life of color blind and people with low vision and hearing.

**Table of Content**

[1 Introduction 17](#_Toc72607493)

[1.1 Brief 17](#_Toc72607494)

[1.2 Relevance to Course Modules 18](#_Toc72607495)

[1.3 Project Background 19](#_Toc72607496)

[1.4 Literature Review 19](#_Toc72607497)

[1.4.1 Machine Learning Based Approaches 20](#_Toc72607498)

[1.4.2 Artificial Neural Network 20](#_Toc72607499)

[1.4.3 Support Vector Machine 20](#_Toc72607500)

[1.4.4 Convolution Neural Network (CNN) 21](#_Toc72607501)

[1.4.5 Comparison of CNN research work based on the CIFAR10 dataset 21](#_Toc72607502)

[1.5 Analysis from Literature Review 22](#_Toc72607505)

[1.6 Methodology and Software Lifecycle for this Project 23](#_Toc72607506)

[1.6.1 Agile Development 23](#_Toc72607507)

[1.6.2 Rationale behind selected methodology 24](#_Toc72607508)

[2 Problem Definition 26](#_Toc72607509)

[2.1 Problem Statement 26](#_Toc72607510)

[2.2 Deliverables and Development Requirements 26](#_Toc72607511)

[2.2.1 Deliverables 26](#_Toc72607512)

[2.2.2 Development Requirements 27](#_Toc72607513)

[3 Requirement Analysis 29](#_Toc72607514)

[3.1 Use Case Diagram 29](#_Toc72607515)

[3.2 Detailed Use Case Diagram 30](#_Toc72607516)

[3.2.1 Use-case # 01 (Upload / Capture Image) 30](#_Toc72607517)

[3.2.2 Use-case #02 (Preprocessing) 30](#_Toc72607518)

[3.2.3 Use-case #03 (Object Classification) 31](#_Toc72607519)

[3.2.4 Use-case #04 (Color Classification) 32](#_Toc72607520)

[3.2.5 Use-case #05 (Result) 32](#_Toc72607521)

[3.2.6 Use-case #06 (Close Application) 33](#_Toc72607522)

[3.3 Functional Requirement 33](#_Toc72607523)

[3.4 Non-Functional Requirement 34](#_Toc72607524)

[4 Design and Architecture 36](#_Toc72607525)

[4.1 System Architecture 36](#_Toc72607526)

[4.2 Data Representation [Diagram + Description] 37](#_Toc72607527)

[4.2.1 Data Flow Diagram 37](#_Toc72607528)

[4.3 Process Flow / Representation 39](#_Toc72607529)

[4.3.1 Activity Diagram 39](#_Toc72607530)

[4.4 Design Models 40](#_Toc72607531)

[4.4.1 Sequence Diagram 40](#_Toc72607532)

[4.4.2 Class Diagram 41](#_Toc72607533)

[5 Implementation 44](#_Toc72607535)

[5.1 Algorithm 44](#_Toc72607536)

[5.1.1 Dataset 44](#_Toc72607537)

[5.1.2 Preprocessing 45](#_Toc72607538)

[5.1.3 Object Classification 46](#_Toc72607539)

[5.1.4 Color Classification 51](#_Toc72607540)

[5.2 User Interface 51](#_Toc72607541)

[5.2.1 Splash Screen 52](#_Toc72607542)

[5.2.2 Main Activity 53](#_Toc72607543)

[5.2.3 Without upload Image 54](#_Toc72607544)

[5.2.4 Upload Image from gallery 55](#_Toc72607545)

[5.2.5 Runtime Image capture 56](#_Toc72607546)

[5.2.6 Connection with server 56](#_Toc72607547)

[5.2.7 Classification Examples 58](#_Toc72607548)

[6 Testing and Evaluation 64](#_Toc72607553)

[6.1 Manual Testing 64](#_Toc72607554)

[6.1.1 System Testing 64](#_Toc72607555)

[6.1.2 Unit Testing 65](#_Toc72607556)

[6.1.3 Functional Testing 66](#_Toc72607557)

[6.1.4 Integrated Testing 67](#_Toc72607558)

[6.2 Tools 68](#_Toc72607559)

[7 Conclusion and Future Work 71](#_Toc72607561)

[7.1 Conclusion 71](#_Toc72607562)

[7.2 Future work 71](#_Toc72607563)

[8 References 73](#_Toc72607565)

**List of Figures**

[Figure 1‑1 CIFAR10 Objects Image 18](#_Toc72608220)

[Figure 1‑2 Agile Development 23](#_Toc72608221)

[Figure 3‑1 Use Case Diagram 28](#_Toc72608222)

[Figure 4‑1 System Architecture 36](#_Toc72608223)

[Figure 4‑2 Data flow diagram Level 0 37](#_Toc72608224)

[Figure 4‑3 Data Flow Diagram Level 1 38](#_Toc72608225)

[Figure 4‑4 Activity Diagram 39](#_Toc72608226)

[Figure 4‑5 Sequence Diagram 40](#_Toc72608227)

[Figure 4‑6 Class Diagram 41](#_Toc72608228)

[Figure 5‑1 Network Architecture 45](#_Toc72608229)

[Figure 5‑2 Trained CNN Model 48](#_Toc72608230)

[Figure 5‑3 Accuracy Graph 49](#_Toc72608231)

[Figure 5‑4 Loss Graph 49](#_Toc72608232)

[Figure 5‑5 Epochs 50](#_Toc72608233)

[Figure 5‑6 Splash Screen 51](#_Toc72608234)

[Figure 5‑7 Main Activity 52](#_Toc72608235)

[Figure 5‑8 without upload image 53](#_Toc72608236)

[Figure 5‑9 Upload image from gallery 54](#_Toc72608237)

[Figure 5‑10 Runtime Capture Image 55](#_Toc72608238)

[Figure 5‑11 OkHttp Dependency 56](#_Toc72608239)

**List of Tables**

Table 1‑1 Comparison of CNN research work 22

Table 3‑1 Use case 01 Input image 29

Table 3‑2 Use case 02 Preprocessing 29

Table 3‑3 Use case 03 Object Classification 30

Table 3‑4 Use case 04 Color Classification 31

Table 3‑5 Use case 05 Result 31

Table 3‑6 Use case 06 Close Application 32

Table 6‑1 Test Case 1 65

Table 6‑2 Test Case 2 65

Table 6‑3 Test Case 3 66

Table 6‑4 Test Case 4 66

Table 6‑5 Integrated Testing 66

Table 6‑6 Tools 67

**Chapter 1**

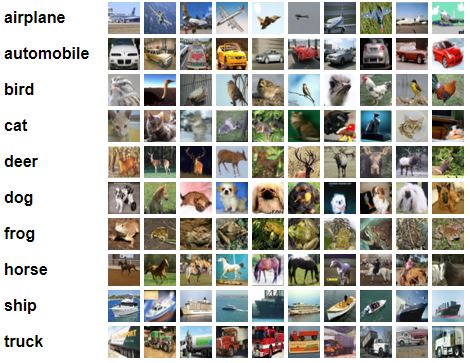
**INTRODUCTION**

# Introduction

Computer vision has been around for decades and has many different uses in the real world. Some examples are facial recognition, disease detection, and object detection. The whole concept of computer vision depends heavily on machine learning, especially in today's world, and the way new technologies are emerging. Object recognition is a largely forthcoming scope for detecting objects using machine learning methods which are used for social and industrial applications. Color is a feature that has been heavily exploited in digital image processing, as it is a powerful tool that often facilitates the classification and identification of objects. The Android application is a product application that sudden spikes in demand for the Android platform. Android provides a comprehensive application framework that allows us to create modern apps for mobile devices in a Java language environment. Today, smartphone users are leaping. It is estimated that the 80% of the populace utilizes cell phones. What's more, they utilize their cell phones to meet their day by day needs. The purpose of this project is object and color detection for helping low vision, color blind people, and learning perspective by using machine learning techniques, image processing and, speech APIs and text labeling. Thus, we conclude that object detection is useful in day to day life for learning and low visually impaired users.

## Brief

Object recognition is a largely forthcoming scope for detecting objects using machine learning methods which are used for social and industrial applications. The saved image in gallery or run-time capture image is input to the application. It detects that image from all angles. It is first preprocessed by operation and then fed for classification, then object class and color will be classified in output. This system proved that it improves accuracy and correctly detect the objects. This application will improve each day lives of color blind and people with low vision and hearing. By studying the literature, we have analyzed that system’s performance can be better by improving the computational speed of it and the accuracy of the model for object detection. For increasing the performance, we have focused on building an application that has better computational speed, and accuracy. We proposed an efficient classification technique able to perform an automated classification of different categories of object images. We propose a system, based on image preprocessing, object and color classification. The proposed system takes images and will automatically detect the object class accurately. Many researchers researched for object detection and their researched work also gave promising results but with some drawbacks. As the object has various kinds but our scope revolves around only ten types. These are cat, deer, dog, frog, horse, ship, truck, airplane, automobile, bird.



**Figure 1‑1 CIFAR10 Objects Image**

## Relevance to Course Modules

#### Android development

Android Studio is the IDE that provides Google to develop professional Android applications. It is used to develop a different variety of applications for the Android operating system. It is an IDE and platform to design a user-friendly interface by drag and drop.

#### Machine Learning

In this course, we have studied about different algorithms which are used to train models to perfume task automatically after getting the training data.

#### Report Writing Skills

This course is about learning how to write reports and other formal documentation, and, in our project, we need to write our documentation, so this course is helping a lot in this task.

#### Human Computer Interaction

This course shows the user-friendly behavior of a computer in today’s world. When a system is receptive to a user in all contexts then flexibility is provided to that user. Each system has a frontend, when an input is given to the system through software means, it returns output on that graphical user interface.

#### Digital Image processing

Image processing is an alteration of an image into digital set-up and performs a process on it to get improved data and extract valuable information.

## Project Background

The idea behind this project particularly occurred in our mind when we think about low vision and color blind persons. So we prefer this idea to improve each day lives of color blind and people with low vision and hearing. Previously major’s work has been done for this field but we are here to give a project which is a full package. It’ll have all of the features including audio synthesis. This application is based on machine learning and image processing techniques, when the user puts an image as input the application will correctly predict the object in the image and then will classify its color and the whole result will be shown to the user in form of labeling and audio.

The algorithm is implemented in PYCHARM with language python and the front end is designed in Android Studio with language JAVA and XML and then connect these with help of the flask platform.

## Literature Review

With a specific end goal to influence the task, to yield appropriate output, it is important to experience the past techniques and procedures. This heading contains the complete description of the past work and reports that have been completed in such a manner. Different approaches have been used to detect an object. We illustrate how these approaches differ from each other and bring to light the merits and limitations of each.

### Machine Learning Based Approaches

Machine learning in object recognition plays an important role and gives promising results with improved accuracy. The machine learning also ensures the analysis of the different object data and machine learning also provide the ability to make classy and good algorithms that help in recognizing. For examining the object data, machine learning provides a worthy approach for making efficient algorithms. Following are the approaches that have been proposed by researchers.

### Artificial Neural Network

Sten Sootla purposed a simple fully connected feed-forward artificial neural network with one hidden layer that can classify 10 classes of images from the CIFAR-10 dataset. The resulting artificial neural network has an accuracy of 38% on previously unseen data. It took approximately 223 minutes for the network to finish the computation but it is by no means successful due to the low accuracy.

### Support Vector Machine

In the research work proposed by Karishma Agrawal kdagrawa, Soumya Shyamasundar sshyamas and Pradheep Shanmugam pshanmug classification algorithm is done using Support Vector Machines (SVM). It is used to classify multiple classes using a technique ’One Vs Rest’ also known as ’One Vs All’. In this method, SVM model is created for each class. So for it they need 10 SVM models one for each class in CIFAR 10 dataset. Each model treats one class as a positive data points and the rest of the class as negative data points and calculates the probability of the class. Similarly, each model would classify one class. Using the SVM classification algorithm they classified the object type with the original dataset 41.87%. They also proposed object image classification by applying preprocessing like grey scaling (poor result), principle components analysis (PCA) (No difference), zero component analysis whitening (ZCA whitening) (No difference), mirroring, (increases the training set by flipping the training images horizontal) (improved accuracy 44.86%), adding Gaussian noise to the data(accuracy 43.305%), K-Means clustering( 55.276%.

### Convolution Neural Network (CNN)

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox and, Martin Riedmiller proposed a classification approach which has pure CNN architecture. They find that max-pooling can simply be replaced by a convolutional layer with increased stride without loss in accuracy on several image recognition benchmarks. They propose a new architecture that consists solely of convolutional layers and yields competitive or state of the art performance on several object recognition dataset (CIFAR-10). The accuracy they achieved on CIFAR-10 dataset with data augmentation 92.75 without data augmentation 90.92%. Our project is based on the same approach and dataset they have used to achieve better accuracy and computational time.

### Comparison of CNN research work based on CIFAR10 dataset

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Author** | **Method** | **Target** | **Dataset** | **Accuracy** |
| Aarya Brahmane | CNN involves a Rectified Linear Unit (ReLu), a convolution, and a pooling layer, a drop out and a dense layer. | Object Classification | CIFAR-10 | 78% |
| Parneet Kaur | CNN involves a Rectified Linear Unit (ReLu), convolution, pooling, drop out, flatten and dense layer. Also used data augmentation. | Object Classification | CIFAR-10 | 85.79% |
| Abhijit kumar | CNN involves a Rectified Linear Unit (ReLu), 6 layered convolution neural network followed by flatten layer. The output layer is dense layer with softmax activation. With Data Augmentation and Regularization | Object Classification | CIFAR-10 | 90% |
| Attyuttam saha | CNN involves a Rectified Linear Unit (ReLu), a convolution, and a pooling layer | Object classification | CIFAR-10 | 70% |

**Table 1‑1 Comparison of CNN research work**

21

5. SYSTEM REQUIREMENT:

4.2.1 Problems with R-CNN

•It still takes a huge amount of time to train the network as you would have to classify 2000 region

1

## Analysis from Literature Review

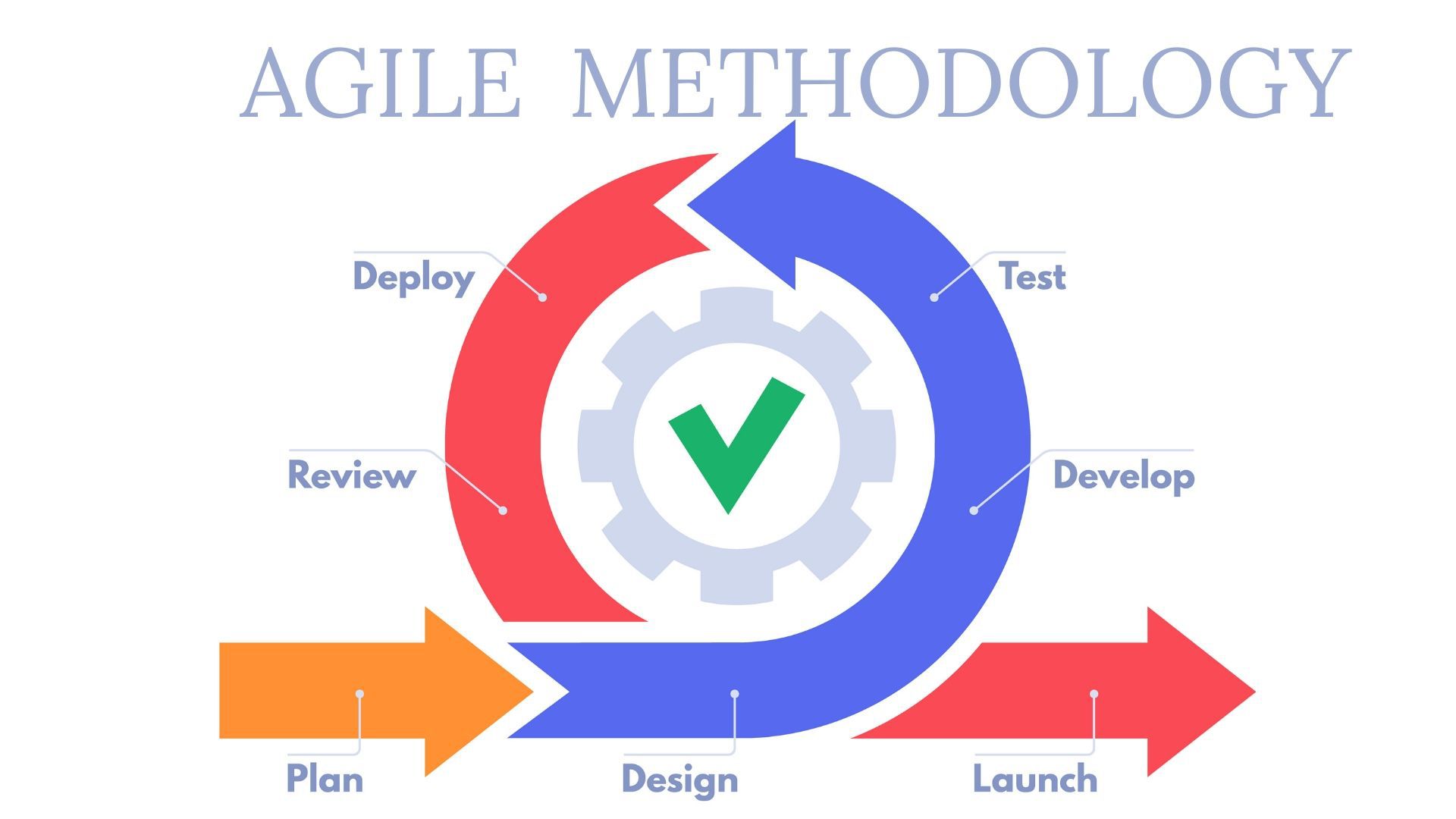
A task is accomplished with the specified goal of achieving the desired output, it is necessary to go through past research and techniques to evaluate the details. Different approaches have been used in the past few decades for object classification. These techniques differ from each other, as discussed above the artificial neural network, support vector machine and convolutional neural network with a method, limitation and, accuracy.

## Methodology and Software Lifecycle for this Project

The project methodology is an important phase of any project because it is a key element and sets the overall tone. For this, we must first understand the steps that are involved in project methodology. We use an agile development strategy to develop our model.

### Agile Development

Agile development is an interactive software development strategy that involves cooperation and self-organizational cross-functional teams. This involves the delivery of project incrementally rather build and deliver the whole project at once. Object and color detection application is basically android based. We must deliver a progress report to the supervisor on an incremental basis and on that we gradually develop this application. Agile is the combination of methodology and technique.



**Figure 1‑2 Agile Development**

### **Rationale behind selected methodology**

We have used Agile Methodology in building our application. We have built our application in different layers. While building each layer we have discussed each and everything after discussing each idea we had to work on the best idea. We have followed each sequence of agile methodology while building each layer of the project.

**Chapter 2**

**PROBLEM DEFINITION**

# Problem Definition

Multiple external devices and mobile applications are used for object or color detection. The main problem is, if a low vision and color blind people wants to recognize an object and its color then how he/she will do it? By downloading multiple android apps for multiple purposes which is not the effective way and also resources and a lot of effort required for it. Moreover, the low vision people unable to read text labeling on the screen.

## Problem Statement

We have seen many projects but they are all performing a single task. If someone is hired to detect objects, they will download the object detectors app and if someone is hired to detect color, they will download the color detectors app. What if a person is suffering from multiple disabilities? These all devices and multiple application is not effective for such type of person because it required more efforts and extra money. We have created this app to get rid of these problems. This could be accomplished by providing low vision and color-blind people with information about the things which will be helpful for them. That’s why we built this project which performs more task at a time.

## Deliverables and Development Requirements

### Deliverables

The term deliverable refers to an outcome or product that is provided to the user at the end after the project is completed. A software program or hardware system can be a deliverable along with a user manual. We developed android application of “Object and color detection” so that the user inputs the object image in it and after going through a set of algorithms, it shows output about the object label. The frontend is very user-friendly and flexible as it simply takes an object image and predicts the label and color of it in the form of audio and labeling.

### Development Requirements

#### Software Requirement

* **Operating System:** Windows 10
* **Front end:** Android Studio 4.0.1
* **Backend:** PYCHARM 2021.1, ANACONDA 2021.11

#### Hardware Requirement

* 8.00 GB Installed memory in laptop
* Android Mobile

#### Language

* Python 3.8
* Java
* XML

#### Library

* Numpy
* Matplotlib
* Tensorflow
* Keras
* Flask
* Werkzeug
* PIL
* Scipy
* Webcolors
* pickle

**Chapter 3**

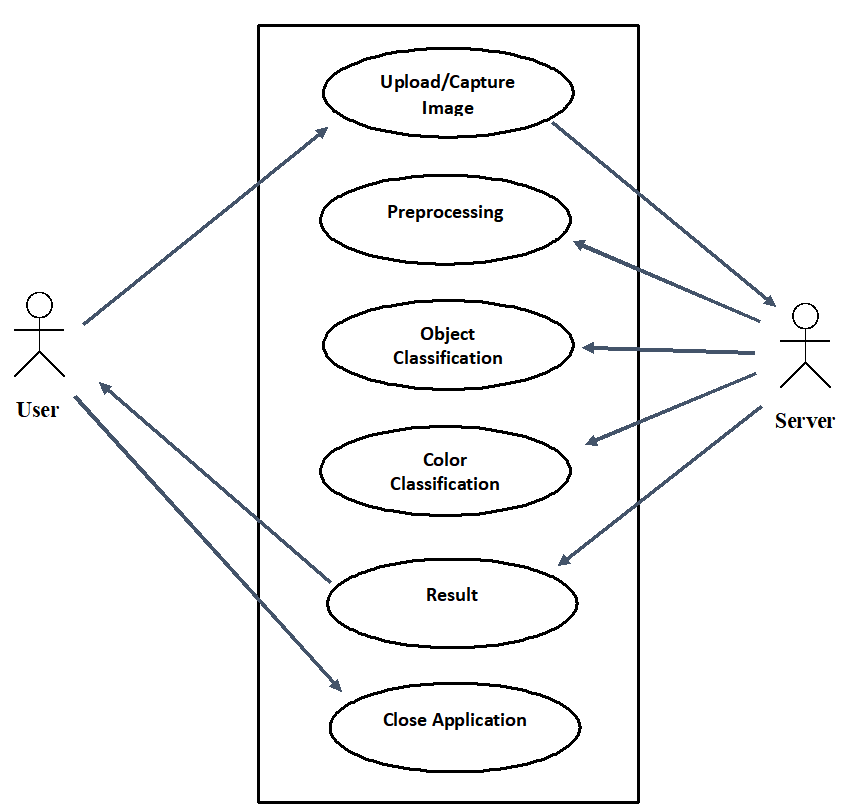
**REQUIREMENT ANALYSIS**

# Requirement Analysis

Software Requirement Analysis (SRS) provides the basic understanding of functional as well as non-functional requirements. We can consider it as a starting point of the project because it serves as a written contract for the client about the features and functionalities of the project.

## Use Case Diagram

It represents the interaction of the actor which is also known as the user with the application or system. It is used to describe the use cases (set of different actions) that can be performed by the actors (external users) by using the system. With the help of the use case, you can visualize the behavior of the system when the user uses it.



**Figure 3‑1 Use Case Diagram**

## Detailed Use Case Diagram

Use case figure is one of the UML diagrams which models dynamic behavior. It is dynamic, consisting of some inside or outside components for making the connection. It is the combination of use cases, users (primary and secondary actor) and, relationships. A use case diagram is used to represent the main aim which is the functions of the system. Many use case diagrams represent the model of entire system. It is a generic way to define that use case diagram captures dynamic aspects of the entire system.

### Use-case # 01 (Upload / Capture Image)

**Table 3‑1Use case 01 Input image**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 1 |
| **Use Case Name** | Upload / Capture Image |
| **Actors** | User |
| **Description** | The user will input an image from the system directory to the application or Capture image run-time. |
| **Triggers** | The image will be uploaded into the system for further processing |
| **Pre-conditions** | The user must have an application |
| **Post-conditions** | The image will go for preprocessing |
| **Normal Flow** | No error |
| **Alternative Flow** | Capture run time or upload from gallery |
| **Expectation** | If the user upload no image, no processing will be done |
| **Business Rules** | Select only one image to be uploaded at a time |

### Use-case #02 (Preprocessing)

**Table 3‑2 Use case 02 Preprocessing**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 2 |
| **Use Case Name** | Preprocessing |
| **Actors** | System |
| **Description** | The image will be preprocessed |
| **Triggers** | The image has been uploaded into the system and ready to be preprocessed |
| **Pre-conditions** | The user has uploaded an image |
| **Post-conditions** | The image will go for classification |
| **Normal Flow** | If any button clicked without image uploading, then the error message will be displayed |
| **Alternative Flow** | No alternative path |
| **Expectation** | All operations of preprocessing will be applied in any order |
| **Business Rules** | Only one image to be preprocessed at a time |

### Use-case #03 (Object Classification)

**Table 3‑3 Use case 03 Object Classification**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 3 |
| **Use Case Name** | Object Classification |
| **Actors** | System |
| **Description** | The image will be classified |
| **Triggers** | The image is done with preprocessing and ready to be classified |
| **Pre-conditions** | The image has been preprocessed |
| **Post-conditions** | An object label is predicted |
| **Normal Flow** | No error |
| **Alternative Flow** | No alternative path |
| **Expectation** | The preprocessed image should be classified |
| **Business Rules** | Select only one image to be classified at a time |

### Use-case #04 (Color Classification)

**Table 3‑4 Use case 04 Color Classification**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 4 |
| **Use Case Name** | Color Classification |
| **Actors** | System |
| **Description** | The color will be classified |
| **Triggers** | The image is done with the object classification and ready to be classified for color |
| **Pre-conditions** | The object has been classified |
| **Post-conditions** | The color is classified |
| **Normal Flow** | No error |
| **Alternative Flow** | No alternative path |
| **Expectation** | The preprocessed image should be classified |
| **Business Rules** | Select only one image to be classified at a time |

### Use-case #05 (Result)

**Table 3‑5 Use case 05 Result**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 5 |
| **Use Case Name** | Result |
| **Actors** | System |
| **Description** | Result will be display |
| **Triggers** | The image is done with object and color classification |
| **Pre-conditions** | The classification has been done |
| **Post-conditions** | Respond back to user |
| **Normal Flow** | No error |
| **Alternative Flow** | No alternative path |
| **Expectation** | Object label is predicted and color is classified and displayed |
| **Business Rules** | Select only one image to be classified at a time |

### Use-case #06 (Close Application)

**Table 3‑6 Use case 06 Close Application**

|  |  |
| --- | --- |
| **Use Case ID** | Use case 6 |
| **Use Case Name** | Close Application |
| **Actors** | User |
| **Description** | Output is displayed and application is closed |
| **Triggers** | Object and color classification is done |
| **Pre-conditions** | Output is displayed to user |
| **Post-conditions** | Application will be closed |
| **Normal Flow** | No error |
| **Alternative Flow** | Yes |
| **Expectation** | Application is closed after display of result |
| **Business Rules** | Click on back button for once |

## Functional Requirement

A functional requirement describes the predictable behavior of the framework. These are the main framework requirements and without applying any of these requirements the framework should be fragmented.

**FR-1** A reliable model for classifying objects taking minimum response time.

**FR-2** Interface of the application is interactive and easy to understand.

**FR-3** The application will take the input image and perform pre-processing.

**FR-4** This preprocessed image has undergone classification method.

**FR-5** Image will be compared to trained model and object label is predicted.

**FR-6** Output received from the model will let the user know about the object label after color classification.

**FR-7** After object classification system will classify the color of the object.

**FR-8** The result will be received from the server and will be displayed to the user in the form of label and audio.

## Non-Functional Requirement

**NFR-1** Performance of our system is efficient, and it takes very little time to act.

**NFR-2** Our system must meet the agreed capacity.

**NFR-3** Our system is reliable and generates an accurate result.

**NFR-4** With the time and needs our system will be maintained properly and any feedback from the user is our priority while performing maintenance.

**NFR-5** This system is efficient as it does not require any kind of effort to use and also does not take much time.

**NFR-6** The system provides the user to load the image easily.

**Chapter 4**

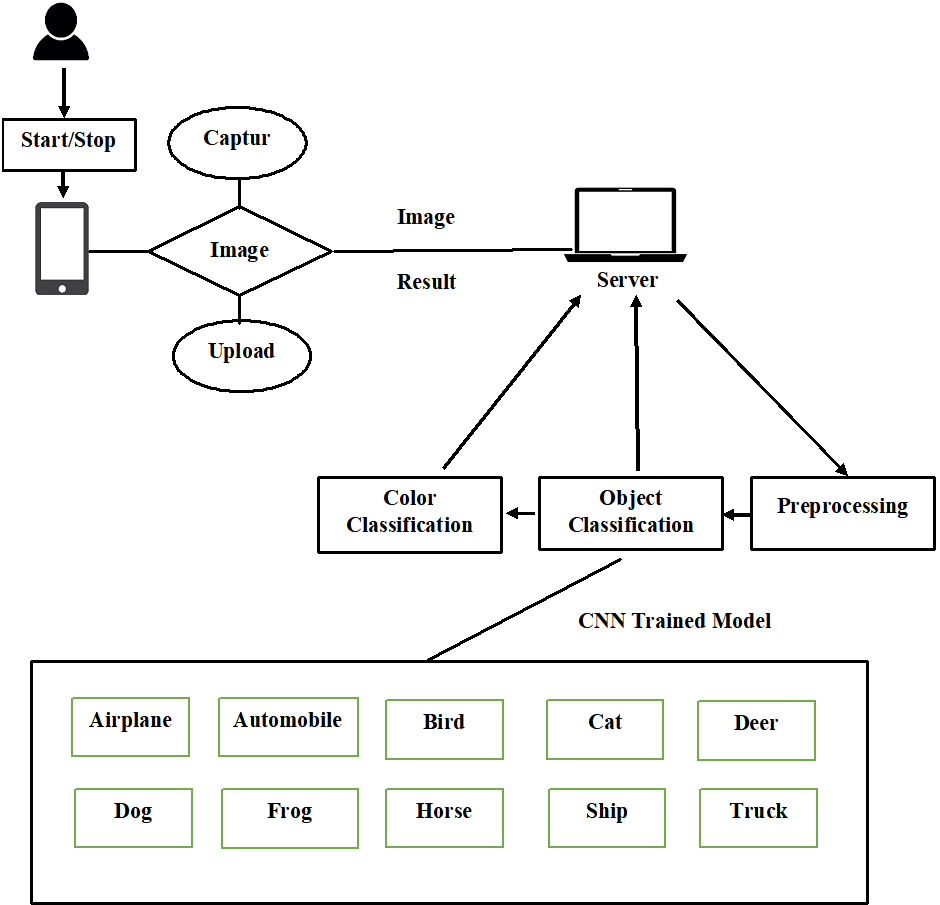
**DESIGN AND ARCHITECTURE**

# Design and Architecture

After gathering all the requirements, the next step is to start planning that how we are going to develop our project, how much resources, costs, time, benefits and other items are required. After planning we move to the designing and architecture phase that which techniques and methods we can use and how we are going to develop our project. It is the most challenging phase of project development. Our architecture consists of 2 main modules: Front end and backend. The front end is an interactive interface that allows the user to the input images. The selected image is then sent to server for classification where the respective result is predicted and response back to application to display the result on the screen. The overall architecture will explain the overall view of the developing model. The front end of the application is designed in android studio. For backend coding we use PYCHARM.

## System Architecture

“Object and color detection” is an application for predicting object labels and color by classifying an image. The developed application will allow the user to an input images, the image will be send to the server, then it will be preprocessed, classify to predict the label of the object. After classification of the object the next step is to identify the color in the image. The predicted object label and color will be sent back to the user application and will be displayed on screen in the form of label or audio. A framework design is a reasonable model that characterizes the construction, conduct, and more perspectives on a framework. An engineering depiction is a proper portrayal and portrayal of a framework, coordinated such that backings thinking about the constructions and practices of the framework. The system architecture of our application is shown as below:



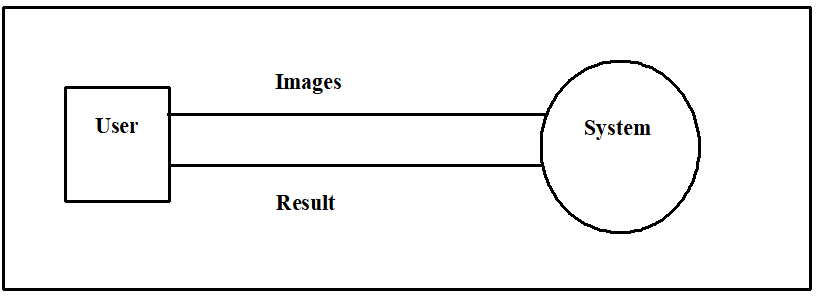
**Figure 4‑1 System Architecture**

## Data Representation [Diagram + Description]

### Data Flow Diagram

Data flow diagram which is the graphical interpretation of information move from a data framework is called a data flow diagram. A DFD is utilized for fundamental advance to outline the framework without going into an incredible angle, which can later be explained. The dataflow diagram shows complete flow of data between modules of the system. The main modules are “image upload”, “preprocessing”, “object classification” and “color classification”; image upload is done by user, application will preprocess and classify it and respond by predicting the label and color and display result.

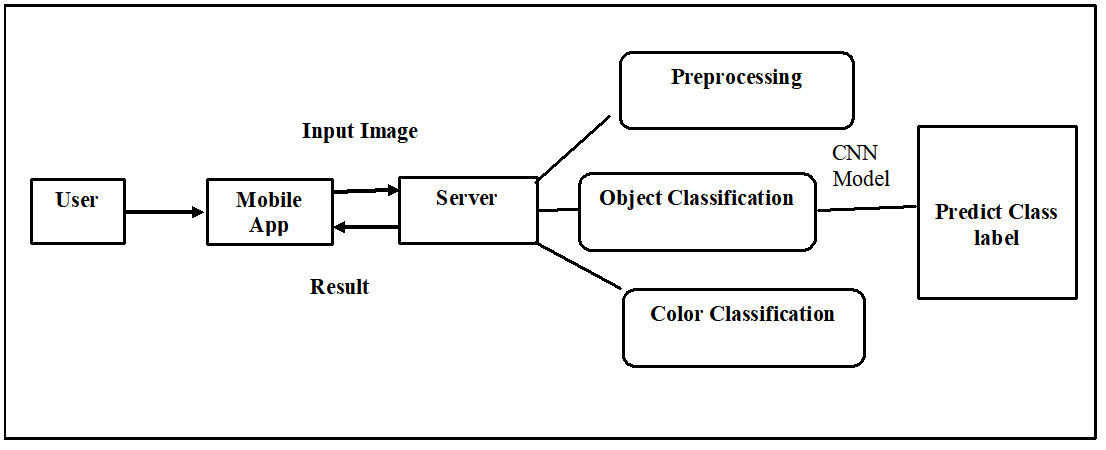
#### Data Flow Diagram Level 0

**Figure 4‑2 Data flow diagram Level 0**

In the level 0 we have 2 entities “user” and “system”. The user can input image and the application can predict the output. In Figure 4-2, the data flow diagram of the “Object and color detection” application is shown, represents that we are inputting an image and as a result application is responding in the form of predicting object label and color of the object.

#### Data Flow Diagram Level 1

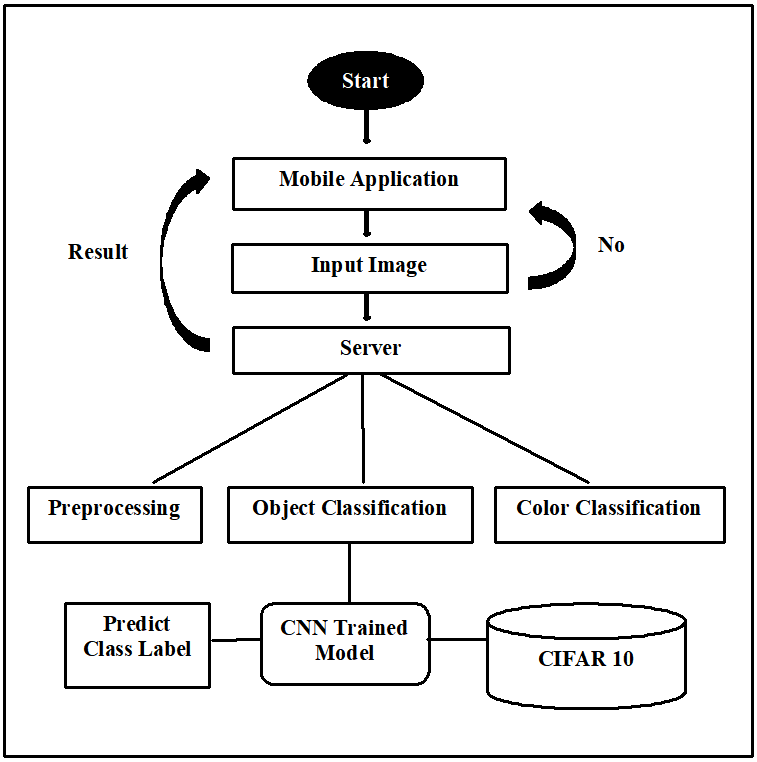
The complete flow of each module has been shown in figure below.

**Figure 4‑3 Data Flow Diagram Level 1**

## Process Flow / Representation

### Activity Diagram

An activity diagram is depicting the dynamic component of the framework. An activity diagram is essentially a flowchart to speak from one action to another movement. The movement may be relating to an operation of the framework. This diagram is used to explain the dynamic features of the system. It is more like a flow chart because it shows the flow of data from one activity to the other.

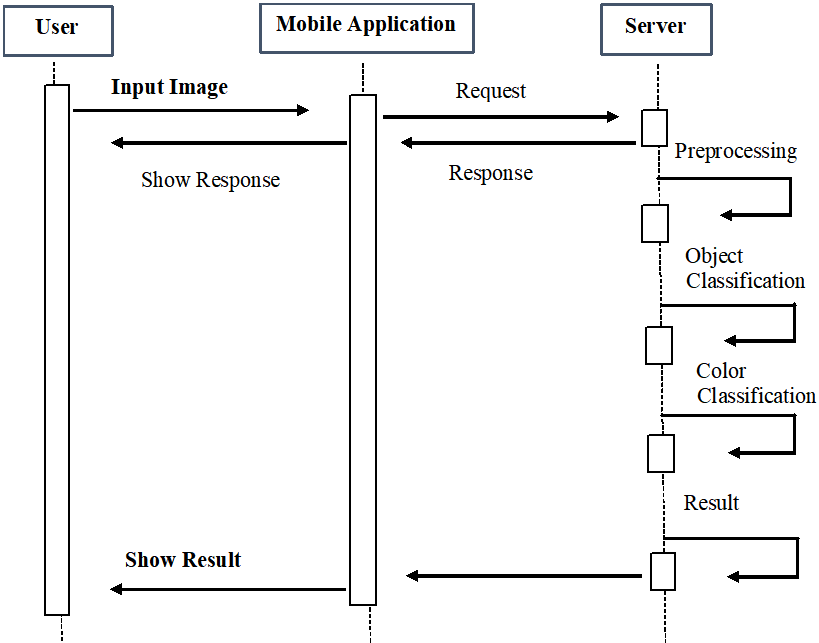


**Figure 4‑4 Activity Diagram**

## Design Models

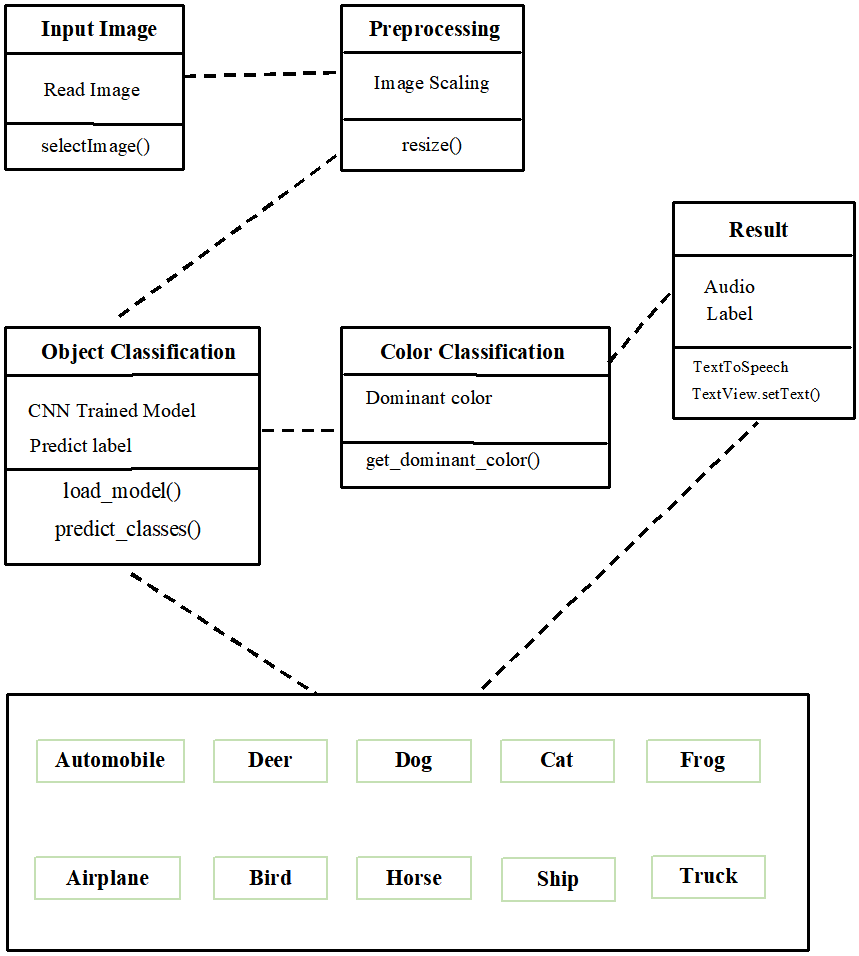
### Sequence Diagram

A cooperation graph that shows how objects work with each other and in what request is called a sequence diagram. It is built up of a message arrangement outline.

**Figure 4‑5 Sequence Diagram**

### Class Diagram

In Unified Modeling Language (UML), the class diagram is a static diagrammatic structure representation in which we explain classes, functions, attributes, and relationships between them.



**Figure 4‑6 Class Diagram**

**Chapter 5**

**IMPLEMENTATION**

# Implementation

In this chapter, we will discuss the algorithms and user interfaces we use to develop the project. This phase is the most prominent phase of development; from this step, we transform our idea into a meaningful picture. This is a significant and challenging step towards learning or developing skills. An application is the result of the successful implementation of the project, various algorithms, testing approaches and the results.

## Algorithm

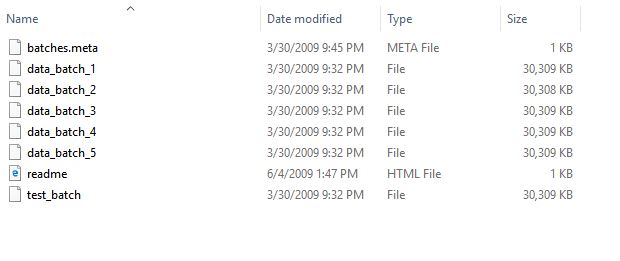
We have implemented a convolutional neural network algorithm in this project.

### Dataset

CIFAR-10 is an established computer-vision dataset used for object recognition. The CIFAR-10 data consists of 60,000 (32×32) color images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images in the data. The label classes in the dataset are:

* Airplane
* Automobile
* Bird
* Cat
* Deer
* Dog
* Frog
* Horse
* Ship
* Truck

The classes are completely mutually exclusive. It was collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class.



### Preprocessing

The initial step of any Machine Learning, Deep Learning or Data Science project is to pre-measure the information. We will characterize the names of the classes, over which the dataset is disseminated. There are 10 distinct classes of color images of size 32x32. Once we have set the class name. We need to standardize the picture with the goal that our model can prepare quicker.

#### Image Scaling

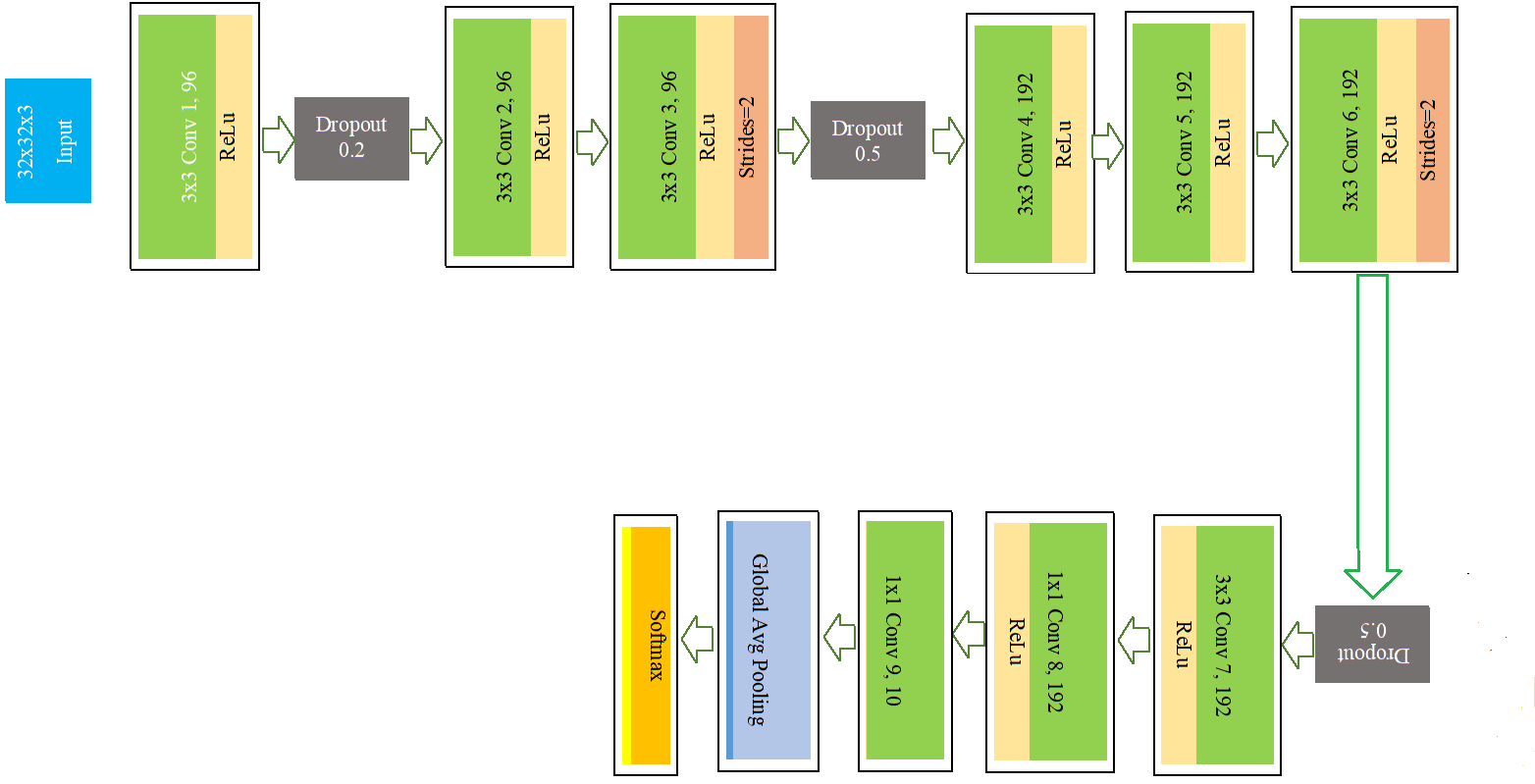
The dataset we take to train model is already processed. We only reshape the images and perform one hot encode over data. When the user will upload the image after running the application we will perform image scaling on that image to fed it to the model for label prediction as the model is train over 32x32 size of images.

### Object Classification

Based on features, the classifier compare the input test image with the trained data and displays the object type in the result.

#### CNN architecture

We implemented simple architecture proposed by Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox and Martin Riedmiller. The feature extraction is done by CNN from the pixels of object images.



**Figure 5‑1 Network Architecture**

##### Convolution

Each input image is passed to series of convolution layers with filters. We use 9 convolutions. In the first convolution layer we use 96 channels with 3x3 kernel size, in the second convolution layer we use 96 channels with 3x3 kernel size and in the third convolution layer is of 96 channels with 3x3 kernel size and stride size 2. In the fourth convolution layer we use 192 channels with 3x3 kernel size, in the fifth convolution layer we use 192 channels with 3x3 kernel size and in the sixth convolution layer is of 192 channels with 3x3 kernel size and stride size 2. In the seventh convolution layer we use 192 channels with 3x3 kernel size, in eight convolution layer we use 192 channels with 1x1 kernel size and in the ninth convolution layer is of 10 channels with 1x1 kernel size.

##### Dropout

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computationally cheap and remarkably effective regularization method to reduce over fitting and improve generalization error in deep neural networks of all kinds. We used 3 dropout layers in our model with dropout values 0.2, 0.5, 0.5.

##### Global average pooling

Global average pooling is used to replace the fully connected or densely connected layers in a classifier. Instead, the model ends with a convolutional layer that generates as many feature maps as the number of target classes and applies global average pooling to convert each feature map into one value. As feature maps can recognize certain elements within the input data, the maps in the final layer effectively learn to “recognize” the presence of a particular class in this architecture. By feeding the values generated by global average pooling into a Softmax activation function. We use one global average pooling layer in our model.

##### Output Layer

Activations can either be used through an Activation layer. In the activation layer, we use Softmax activation function which converts a real vector to a vector of categorical probabilities. It is the last layer of the network.

#### Implementation steps (On dataset)

#### Input Dataset

The dataset after decoded and pre-processing is import through the helper function.

* **Data Augmentation**

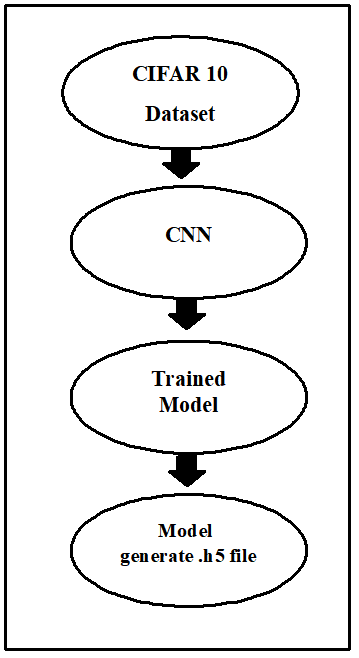
It has been known that augmenting training data often leads to better generalization. We decided to employ three primary data augmentations in the following experiments: flipping, rotating and, shifting. For flipping, we flipped each sample horizontally. In the case of shifting, we shifted it by 2 pixels to the top (20% chance), 2 pixels to the bottom (20% chance). After this, we rotate images with an angle of 45 degrees. This three step procedure makes the model more robust to slight shifting of an object in the image. The shifting was done without padding the borders of the image, preserving the original size of the image. The choice of whether to apply these augmentation procedures on the dataset was chosen on a per-case basis to maximize validation performance.

* **CNN Model**

CNN is a network that uses an image input. The image combine with filters that produce features. Using convolution layers, pooling, and softmax activation function at the output layer, the object classification is done. Convolutional layers, global average pooling, and the output layer with softmax activation layer are used.

* **Training of CNN Model**

CNN is a classifier, which gets trained on the CIFAT10 dataset of images. The trained model generates a (.h5) file, which will help in predicting object labels.

****

**Figure 5‑2 Trained CNN Model**

In the training phase, we used 50,000 images for it. Data augmentation is also done to reduce over fitting and to gain good accuracy. Firstly, we define the model after defining we built the model and then configure it. Then we fit it using the fit function. Using a learning algorithm, we update the filters at the convolution layers. Weights are updated and the learning algorithm learns the input using a learning or optimizer algorithm Adam to update the filters at the convolution layers. The learning algorithm takes a classification error or loss. The loss act as an input and back propagates the error into the network. Then weights and filters are updated.

This process continued. After training, the model we save the model with (.h5) file extension. The training is done in 350 epochs. This file has all the information of the network and ready to reuse again when we perform a test single image for GUI.

* **Testing**

We used 10,000 images for testing. Testing data give a glimpse of CNN was trained and able to classify the objects.

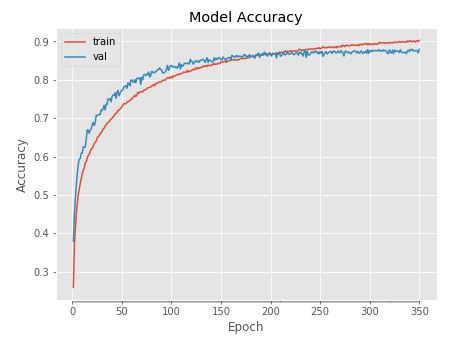
* **Output**

The accuracy of the whole model is returned in the output window. When all steps stated above run successfully, our model return that output as the accuracy of the whole network, testing accuracy, and training accuracy.

#### Classification Accuracies

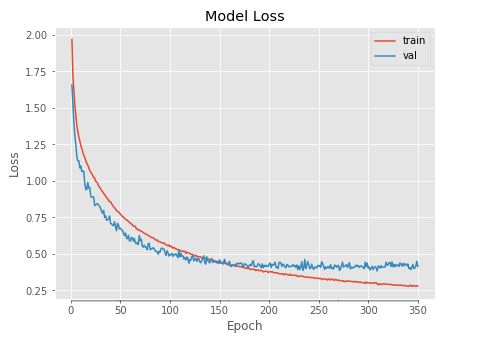
After training the accuracy we achieved on training data is 90% and on testing data 87.94%.

#### Accuracy Graph



**Figure 5‑3 Accuracy Graph**

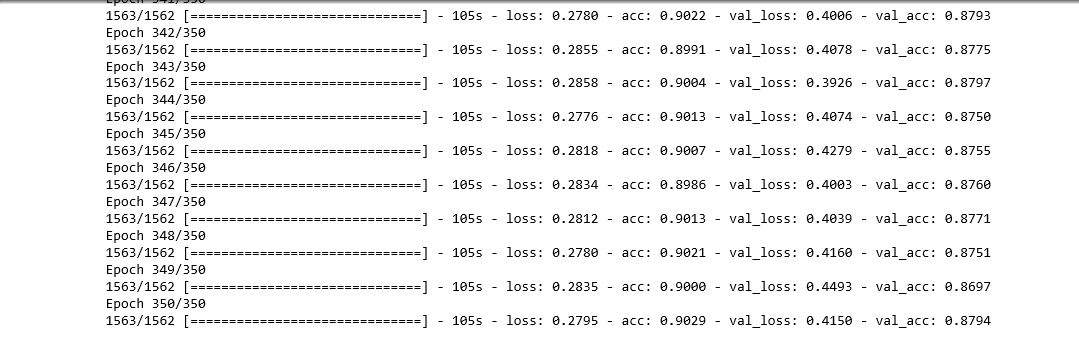
#### Loss Graph



**Figure 5‑4 Loss Graph**

#### Epochs

There are 350 epochs used to train a model.



**Figure 5‑5 Epochs**

### Color Classification

The dominant color of the image is identified using the technique of image processing to classify and display in the result. Color quantization reduces the number of colors in an image to a limited set of distinct colors, called a palette. For this purpose we used python imaging library (PIL), K-Means clustering is used internally to cluster color with the nearest mean. Clusters here are different 16 image colors which will be produced and select the cluster as dominate color which has the most value in RGB space.

## User Interface

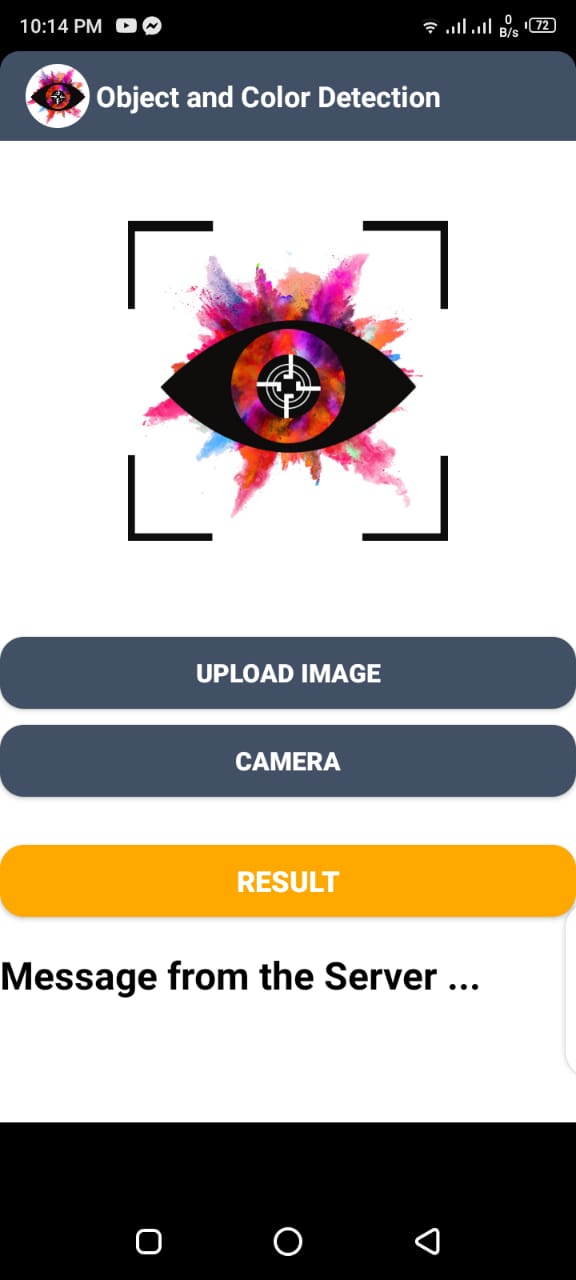
A simple interface will let the user understand easily. After installation of android studio, the interface of the application is designed. Buttons allow user for clicking and uploading the image; each button is named as per functionality so that it makes sense. As the application will start splash screen will open and loads the progress bar. As loading complete main screen will open and allow functionalities of all modules.

### Splash Screen



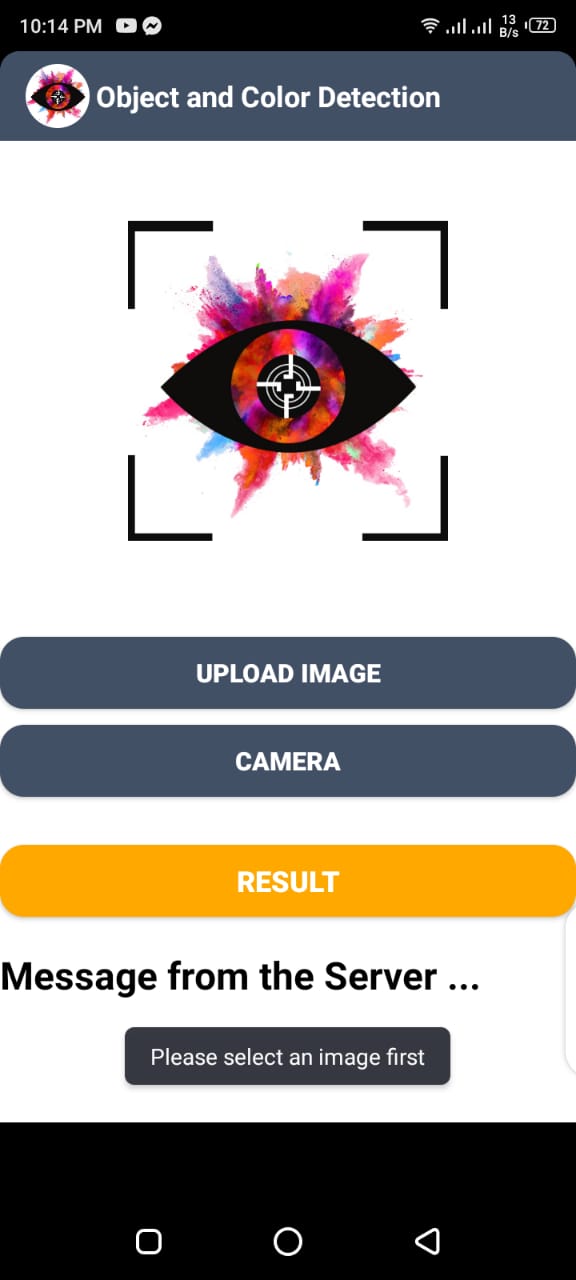
**Figure 5‑6 Splash Screen**

### Main Activity



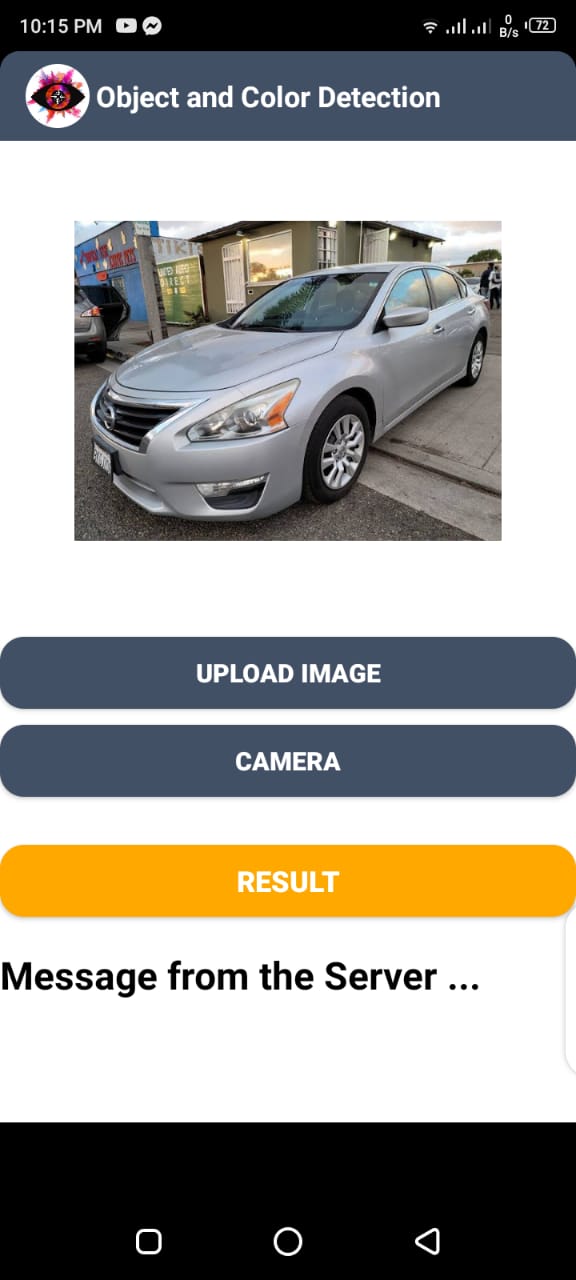
**Figure 5‑7 Main Activity**

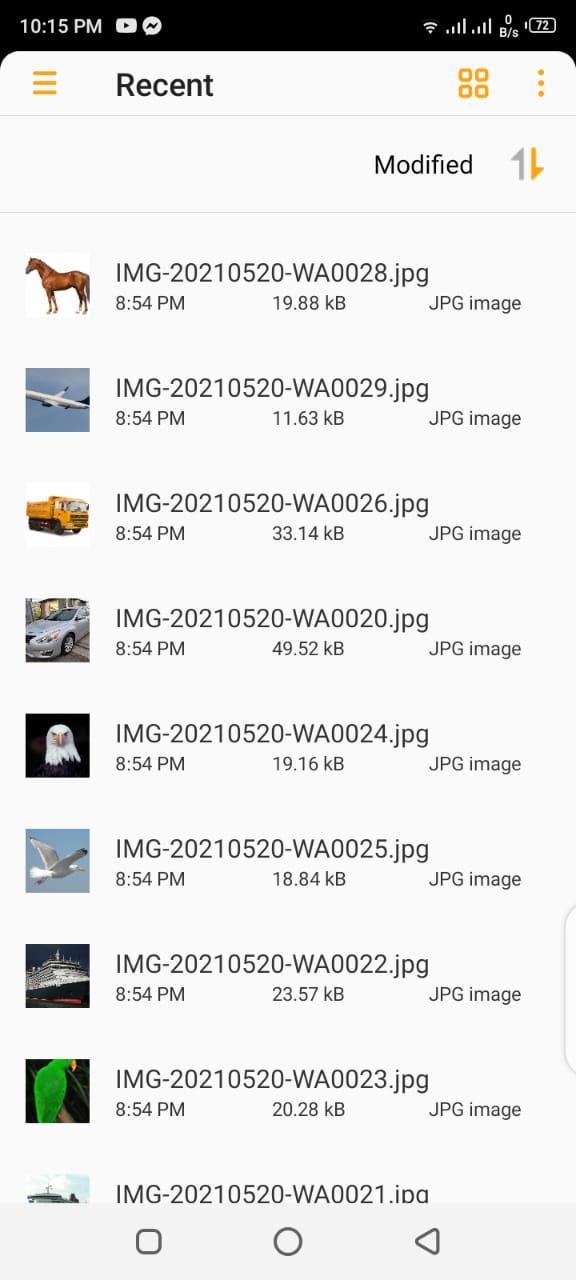
### Without upload Image



**Figure 5‑8 without upload image**

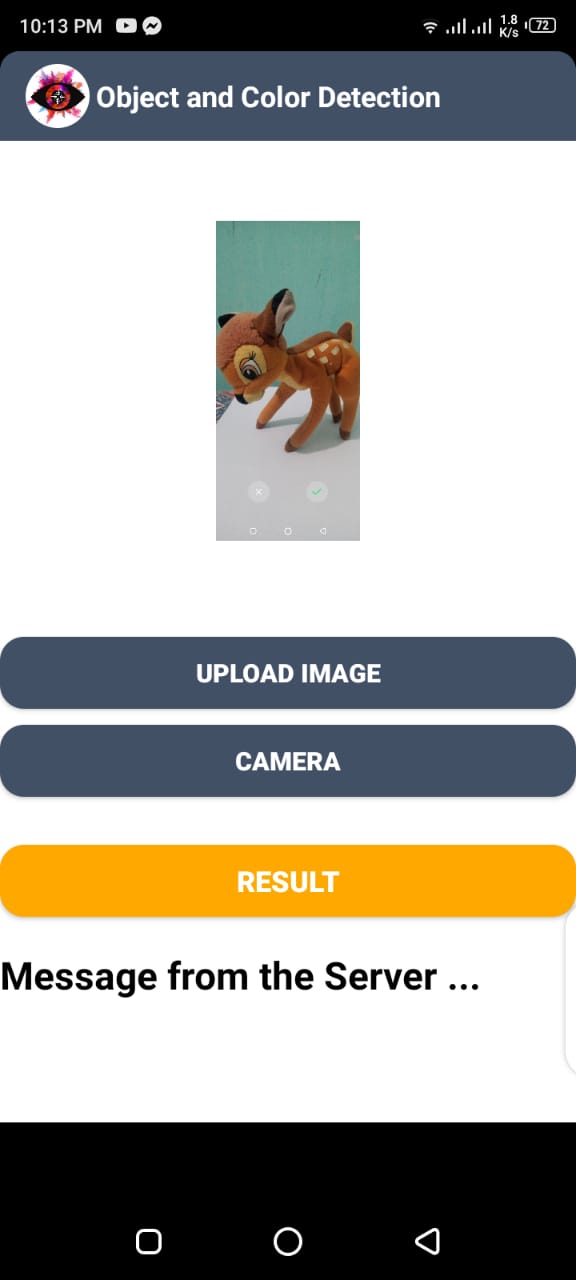
### Upload Image from gallery





**Figure 5‑9 Upload image from gallery**

### Runtime Image capture





**Figure 5‑10 Runtime Capture Image**

### Connection with server

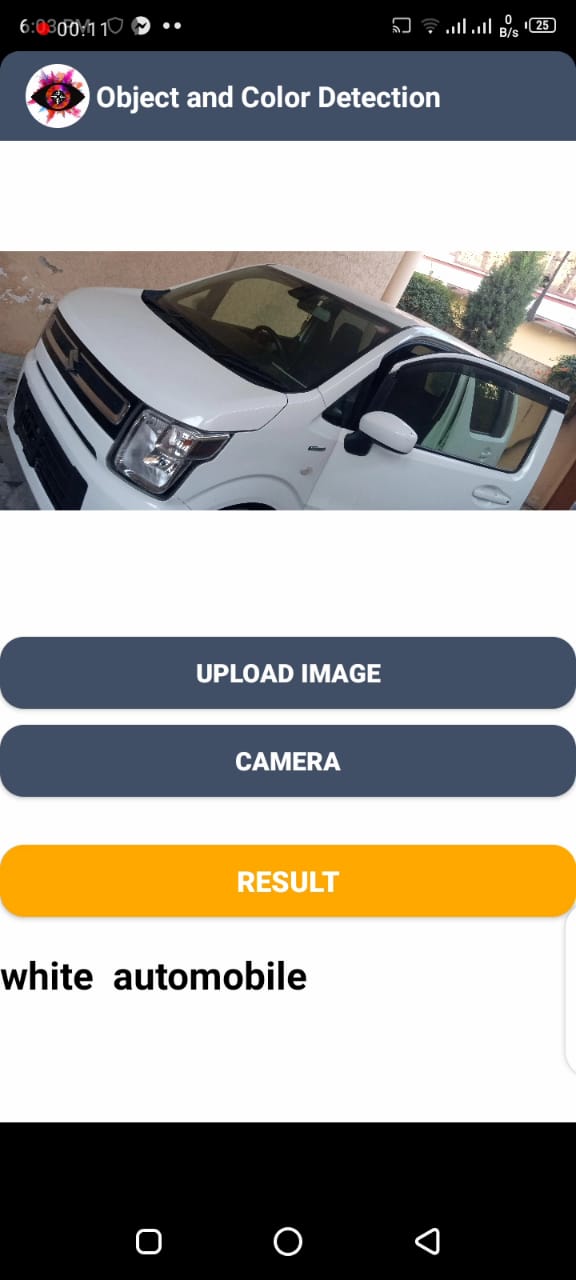
Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. We use a flask framework for the connection. The user will simply upload images to the server using a mobile device and that will be classified by model so the device itself doesn't have to run inference.

For connection, the HTTP requests and reactions are taken care of in Android utilizing OkHttp. The android application simply makes a HTTP POST directive for sending a message. The Flask server gets these requests and answers with an affirmation message that the association is successful. From that point forward, the Android application is altered to transfer a picture to the Flask server. A picture is chosen from android stockpiling through an Intent. The chose picture is changed over into a byte cluster which is then gotten at the Flask worker utilizing its attachment data (for example IPv4 address and port number). The got byte exhibit is then changed over into a picture for additional working. At the point when clicked result button, a callback strategy named connectServer() is called, which builds up the OkHttp association. Beneath this Button, there's a TextView for showing the condition of the association with the worker. On the off chance that the association fizzles, a message mirroring the disappointment is printed, and the other way around for an effective association. Before utilizing OkHttp inside the action class, the task should uphold utilizing it. To do that, we add the dependencies in part of the build.gradle (application module). At the point when Gradle builds the project, the OkHttp library will be downloaded and will utilize it. This way the image will be sent to the server for classification and the result of it will be sent first to the mobile application and then will be displayed to the user.

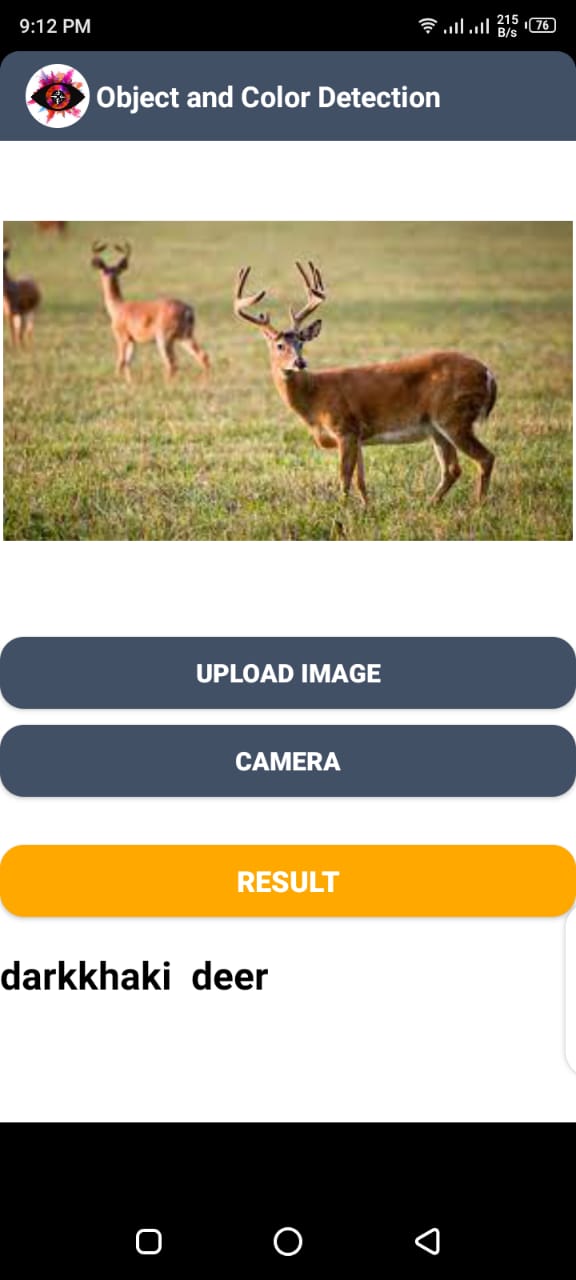


**Figure 5‑11 OkHttp Dependency**

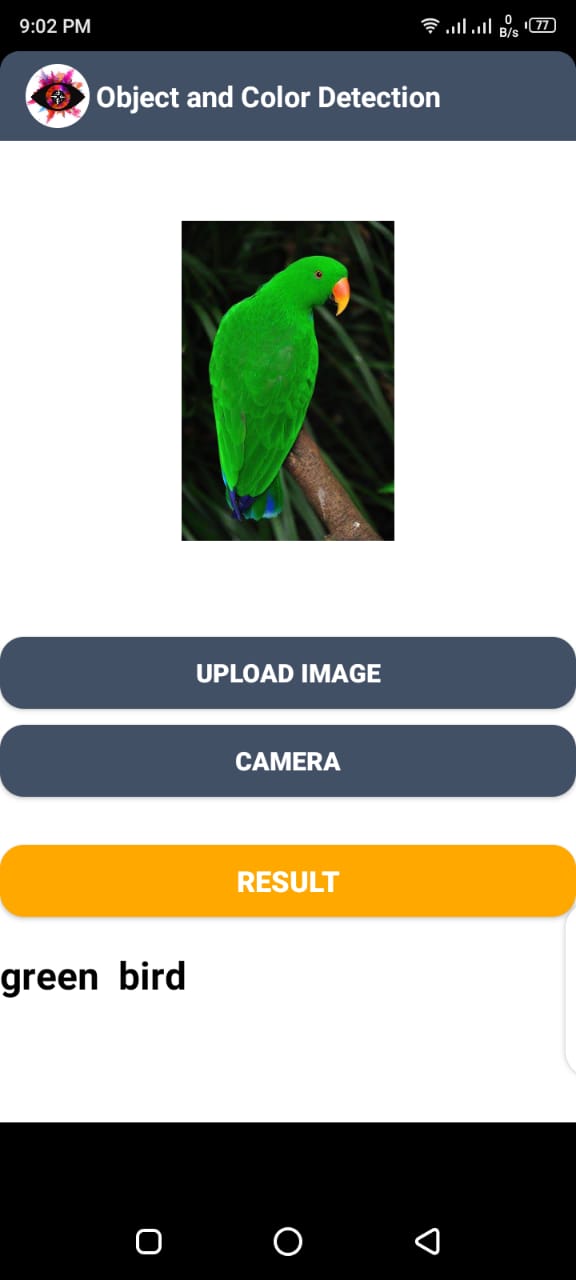
### Classification Examples



**Classification Example 2**

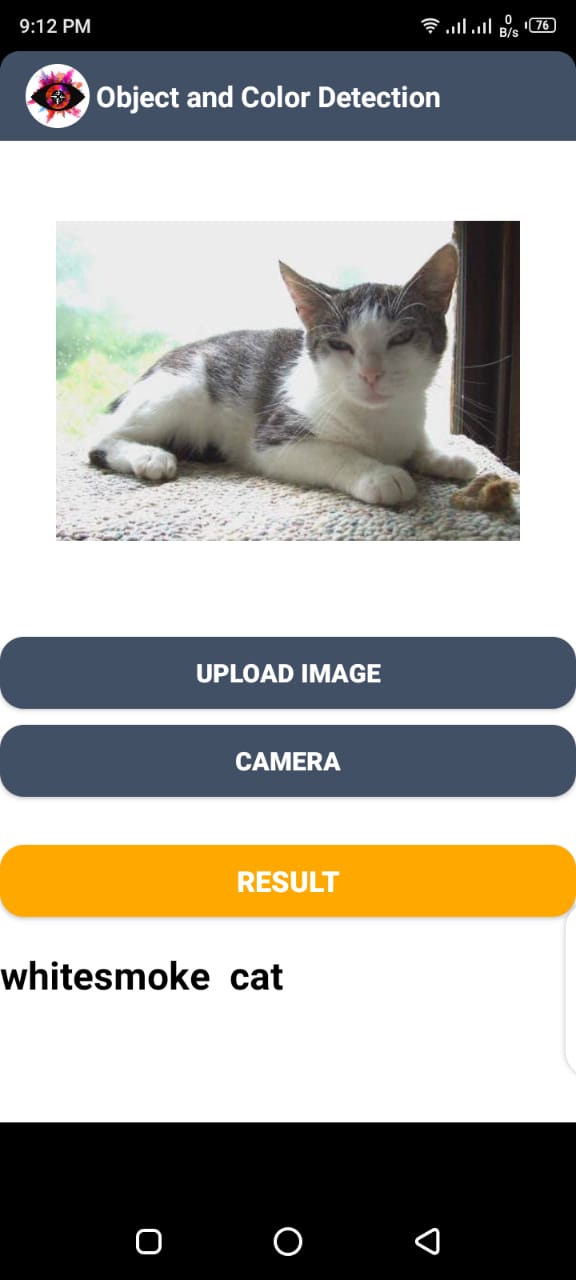


**Classification Example 1**



**Classification Example 4**

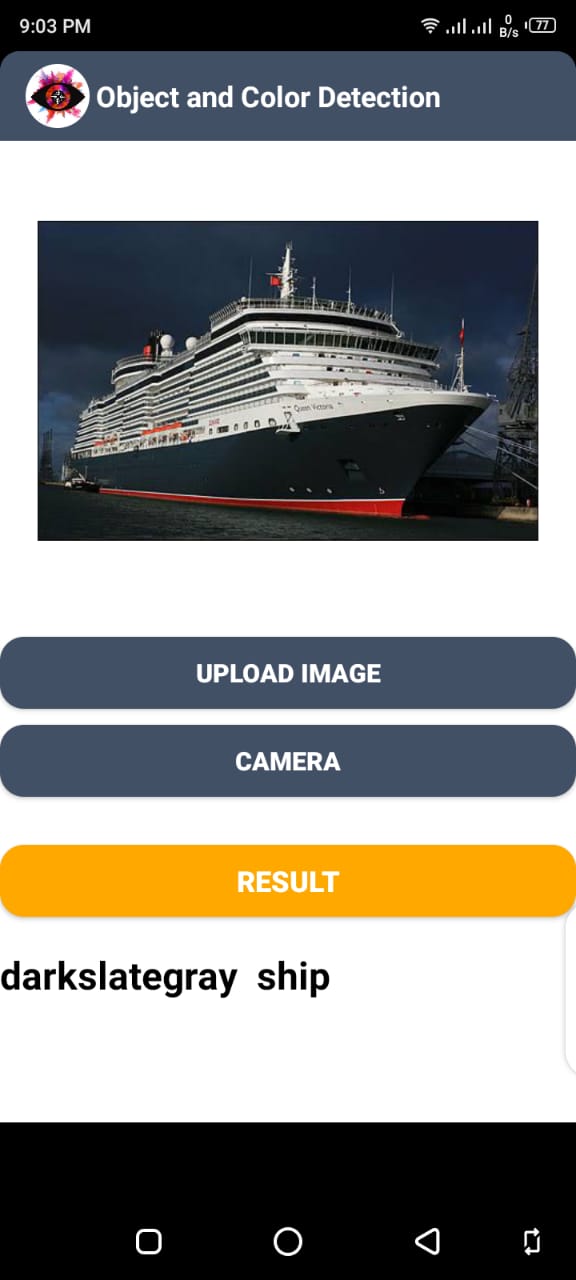
**Classification Example 3**



# 

**Classification Example 6**

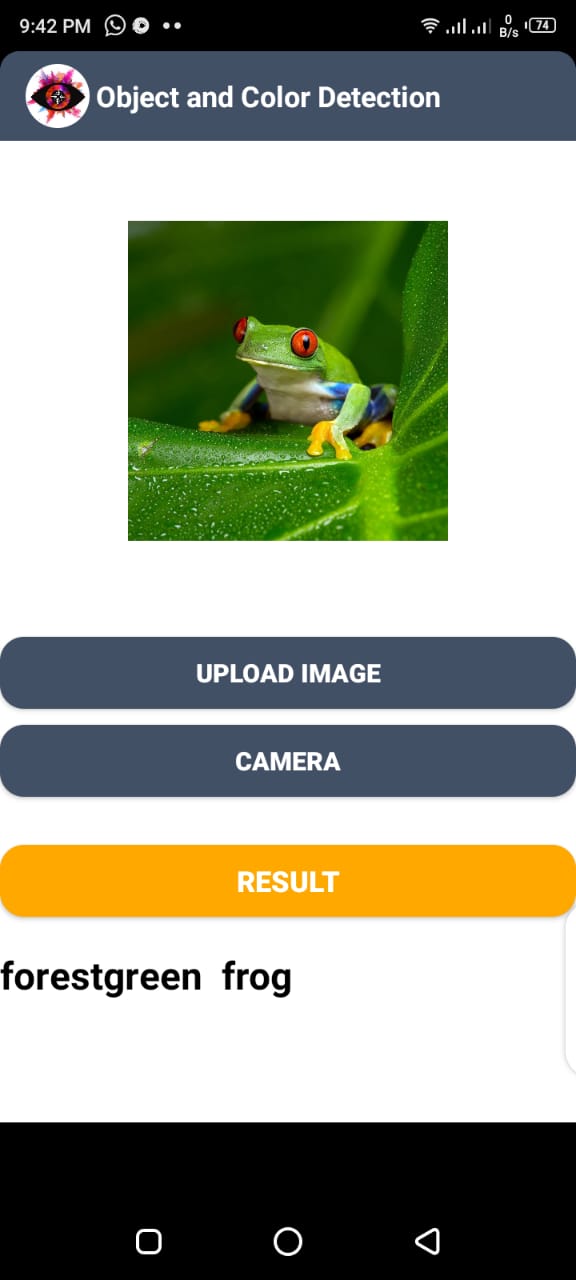
**Classification Example 5**



# 

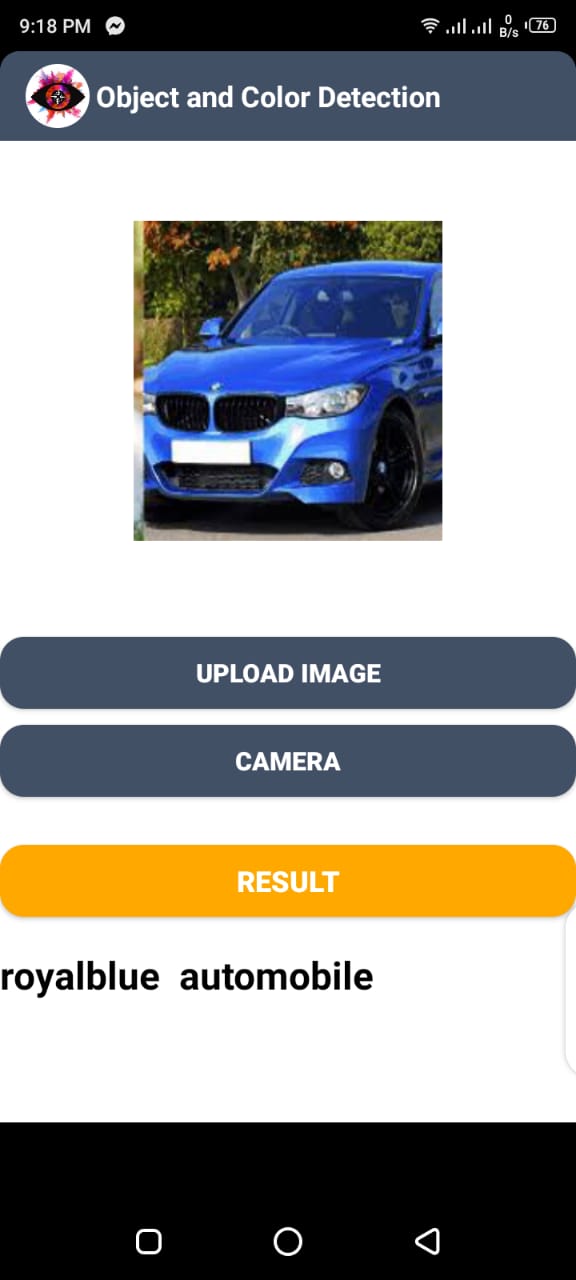
**Classification Example 8**

**Classification Example 7**



# 

**Classification Example 9**



**Classification Example 10**

**Chapter 6**

**TESTING AND EVALUATION**

# Testing and Evaluation

Evaluation is the stage where we analyze the performance of the application by performing different tests to check the effectiveness. Testing is an important phase for any software system. In this phase, we test the functionality of each module of the system in an organized way and test that our system is according to the user's requirement or not.

## Manual Testing

In manual testing, there is no use of any automated tool and test cases are executed manually. It helps to find out visible and hidden defects in a system. Firstly, the documentation is observed by the tester to know about testing areas of the system. The tester analyses the requirement document to fulfill all needs. Each line of code is examined and then the functionality of each module is checked.

### System Testing

In system testing, the complete system is evaluated on basis of the functionality. The Graphical User Interface (GUI) will be tested in the system testing. The input will be given to the system through the GUI and the outputs will be checked. It includes both functional and non-functional testing. The images are provided to the system and classification results are examined.

### Unit Testing

In unit testing, we test each unit of our system and check whether the system is according to the user requirements or not. This testing can be performed in each unit. Unit testing is mentioned for the whole system.

#### Upload Image

Image uploading is done by clicking the button name “Upload Image” it opens the image gallery of the mobile of the user. The user can select any image from it and that image will be displayed inside the label at interface or the user can the open camera to take the image to upload by clicking on “Camera” button.

#### Preprocessing

After sending the image to the server using the rest APIs, the image will be preprocess by scaling it to feed into the model for the object and color classification.

#### Classification

For classification, first, we made a CNN model that has been trained on our dataset. Our pre-trained model is saved in (.h5) extension which we gave to our system and test the given image, the input image is compared with pre-trained CNN load model. Then it detects the object image and figures out its color then our system displays the result.

### Functional Testing

In functional testing, we test the functionality of the system that the functions are working according to the requirements or not. For functional testing, we generate the test cases. They are performed to check the usefulness of the application from the user point of view.

***Test Case 1***

**Table 6‑1 Test Case 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upload image | The user is required to select the image from gallery to identified object or capture image runtime. | Selected image is ready to be displayed. | Selected image successfully displayed on screen. |

***Test Case 2***

**Table 6‑2 Test Case 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case** | **Attribute and value** | **Expected result** | **Result** |
| 2 | Without uploading image | If user click button of operation. | Error message to be shown. | Error message shown instead of picture. |

***Test Case 3***

**Table 6‑3 Test Case 3**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case** | **Attribute and value** | **Expected result** | **Result** |
| 3 | Prediction | Compare input image with pre trained load model. | Predict the label of image. | Label of image predicted successfully. |

***Test Case 4***

**Table 6‑4 Test Case 4**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case** | **Attribute and value** | **Expected result** | **Result** |
| 4 | Color classify | Dominate color  Will be classify. | Color classify. | Color classify successfully. |

### 

### Integrated Testing

When all modules are integrated and when frontend linked to backend, then the result shown is correct.

**Table 6‑5 Integrated Testing**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No.** | **Test case** | **Attribute and value** | **Expected result** | **Result** |
| 1 | Upload image | The user is required to select image from gallery to know identified object or capture image run time. | Selected image is ready to be shown. | Selected image successfully shown on screen. |
| 2 | Without uploading image | If user click button of operation. | Error message to be shown. | Error message shown instead of picture. |
| 3 | Classification | Compare input image with pre trained load model. | Predict the label of image. | Label of image predicted successfully. |
| 4 | Color classify | Dominate color  Will be classify. | Color classify | Color classify  Successful. |

## Tools

**Table 6‑6 Tools**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool Name** | **Tool Description** | **Applied on [list of related tests cases /**  **FR / NFR]** | **Results** |
| PyCharm | JetBrains PyCharm  Community Edition  2021.1 | All test cases. | Code is written over it. |
| Anaconda  distribution | Anaconda3-2020.11-  Windows-x86\_64 | All test cases | It supports libraries |
| Android Studio | Android Studio 4.1.1 | Used for designing interface. | It displays frontend of system. |
| Microsoft Word | WORD 2013 | Used for documentation. | The documentation is made. |
| Microsoft  PowerPoint | POWERPOINT 2013 | Used for presentation slides. | The presentation slides are made. |

**Chapter 7**

**CONCLUSION AND FUTURE WORK**

# 

# Conclusion and Future Work

## Conclusion

Object recognition is a largely forthcoming scope for detecting objects using machine learning methods that are used for social and industrial applications. Color is a feature that has been heavily exploited in digital image processing, as it is a powerful tool that often facilitates the classification and identification of objects. A lot of machine learning work has been done for object detection. There is a lot of work, which can incorporate in the future. This is just the beginning, not the ending point. It was experimentally found that Convolution Neural Network generates promising results and they are considered as state of art in computer vision. In this project, we implemented the CNN-based system which classifies the object and then identifies its color using the image processing technique. We achieved 90% accuracy on the training dataset and 87.94% accuracy for testing data. Ten object airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck are used for detection. All implementation is done using the machine learning frameworks mentioned previously. This project will help to improve each day lives of color blind and people with low vision and hearing.

## Future work

Following can be implemented as future work:

* Future work will involve, the improvement of classification results and overall accuracy.
* Multiple objects exist and need detection, with more diverse dataset, all the object can be detected.

**Chapter 8**

**REFERENCES**

# References

1. <https://courses.cs.ut.ee/MTAT.03.291/2015_spring/uploads/Main/Artificial%20neural%20network%20for%20image%20classification.pdf>
2. <https://www.andrew.cmu.edu/user/kdagrawa/documents/10601report.pdf>
3. <https://arxiv.org/pdf/1412.6806.pdf>
4. <https://www.cs.toronto.edu/~kriz/cifar.html>
5. <https://www.sciencedirect.com/topics/computer-science/android-application#:~:text=Android%20is%20an%20open%2Dsource,customize%20to%20differentiate%20their%20products>.
6. <https://www.techopedia.com/definition/25099/android-app>
7. <https://medium.com/mindorks/detection-on-android-using-tensorflow-a3f6fe423349>
8. <https://www.cronj.com/blog/color-space-image-processing/#:~:text=In%20Digital%20Image%20Processing%2C%20the,with%20colors%20as%20humans%20interpret>.
9. <https://heartbeat.fritz.ai/uploading-images-from-android-to-a-python-based-flask-server-691e4092a95e>
10. <https://en.wikipedia.org/wiki/Flask_(web_framework)>