B.Tech Minor Project Report

COT-415

On

VISION: Android application for identifying vehicle ownership based on vehicle number plate recognition

By

Suresh Paudel(11530015)

Sandeep Bhandari(11530016)

Prashant Kumar(11530017)

Group No: 29

Under the Supervision of

Dr. Mantosh Biswas

DEPARTMENT OF COMPUTER ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY KURUKSHETRA

136119,HARYANA (INDIA)

July-Dec,2018

CERTIFICATE

I hereby certify that the work which is being presented in the B.Tech Minor Project (COT-415) report entitled "**VISION:Android application for identifying vehicle ownership based on vehicle number plate recognition**", in partial fulfillment of the requirements for the award of the **Bachelor of Technology in Computer Engineering** is an authentic record of my own work carried out during a period from July 2018 to December 2018 under the supervision of Dr. Mantosh Biswas, Asst. Professor, Computer Engineering Department.

The matter presented in this project report has not been submitted for the award of any other degree elsewhere.

SURESH PAUDEL (11530015)

SANDEEP BHANDARI (11530016)

PRASHANT KUMAR (11530017)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Date:

Dr. Mantosh Biswas

Asst.Prof. NIT Kurukshetra

TABLE OF CONTENTS

ABSTRACT

Currently there is no any image processing tool that is available on the standard Android mobile phone. So Vehicle Number Plate Recognition using the android application is going to be implemented. Android platform provides many advantages over traditional PC based software. The accuracy can be increased with less resource consumption and less computational complexities. Moreover android application is portable, user friendly with attractive graphical user interface. Such application is easy to install once it is available in the play store. After recognizing the number plate of the vehicle, the ownership of that vehicle can be obtained using the application. The vehicle number plate recognition involves three main components: license plate detection, character segmentation and Optical Character Recognition (OCR). For License Plate Detection and character segmentation, we used Python and its image libraries. And for OCR, we used machine learning algorithms. We obtained very good results by using these libraries. We also stored records of license numbers in database and for that purpose SQLite has been used.

I. INTRODUCTION

Android application for identifying vehicle ownership based on vehicle number plate recognition is an application which helps us to know information about the owner of the vehicle. It is an image processing technology where there is the combination of hardware and software to detect the number plate of the vehicle. Based on the number plate of the vehicle we can extract information of the owner. This system is mainly implemented in the entrance of highly secured areas like offices, schools, public areas like malls, parks, etc. For making the system portable, user friendly and easily accessible, we can create an android app and install in the phone. If any individual has to know about the vehicle, he/she should simply capture the image of number plate of the vehicle. The image will be processed and he/she will get the desired information of that vehicle.

The Vehicle number plate recognition system recognizes the characters in the number plate through various techniques and algorithms like image pre processing, object detection and character recognition. It consists of camera to capture the image and processing unit to extract the characters and interpret the pixels to numerically readable characters. The Vehicle number plate recognition system can be used in traffic law enforcement including speed trap, stolen car detection, boarder monitoring, parking systems, etc.

In the recent decade, android platform has gained popularity. Android applications are easy to install in the phone, user friendly and are easy to implement. Though many research works have been performed in the field of automatic number plate recognition, these researches are not mainly focused on the smart phones or android platform. As currently there is no image processing tool to detect the number plate of the vehicle in the standard mobile phone, we are going to implement the Vehicle number plate detection technique by developing an android application. The android application developed will have good graphical user interface so that any people can understand it easily and use it to acquire the required information of the vehicle.

To implement Vehicle Number Plate detection on an Android application, we have provided a user friendly UI and we send the further Image processing part to the backend which handles detecting and recognizing the Number Plate through the image sent. It uses camera to take a picture and extract the number plate character from the snapped image. The android version will make it easier to use and is portable rather than using the traditional dedicated camera. In addition to the recognition part, we are maintaining the database of the vehicle and this information can also be displayed in real time in android device. For dataset, we are using the college vehicle information.

The server on the other hand uses Python frameworks and Machine Learning Algorithms to detect the number plate and recognize characters on it. This is done is various stages. The first stage is to acquire the car image using a camera. The parameters of the camera, such as the type of camera, camera resolution, shutter speed, orientation, and light, have to be considered. The second stage is to extract the license plate from the image based on some features, such as the boundary, or the existence of the characters. The third stage is to segment the license plate and extract the characters by labeling them. The final stage is to recognize the extracted characters using classifiers called SVM.

Then the acquired string from the server is checked against the already present database in the application present inside the handheld during installation of the application. The Vehicle Number on successful check allows us to view the already stored profile of the Owner and we are saved from the repeated input of the ownership details which is commonly seen at guard posts. With a well built database, a significant time is saved using this Application rather than using the traditional register and pen . On the other hand, if the database is not present in the database, we are asked to manually feed it into the database, so that it can be used in the future. With this app, we also aim to increase the accuracy in the amount of clarity in the photo taken from the mobile so as to provide a good sharp pixels to the servers for Number Plate Detection.

Thus, Our Application VISION not only detects Vehicle Number Plates but also store their appropriate information in the database for future use.

II. MOTIVATION

Today's world is gradually shifting from manual to digital. Taking the records of every vehicle entering and exiting the entrance manually is really time consuming and it takes more human effort

In the context of our college, many security officers are deployed to record the information of the vehicles that are entering and exiting through the main gate. The information is recorded in the register manually. This process is time consuming and in the peak office hours we can see heavy traffic near the main entrance of the college. The condition is really annoying. So we came up with the idea to solve the problem faced in the main entrance of the college by developing the android application which can be useful in registering the vehicles and acquiring the information easily. So while developing the android application we are using the database of the vehicles that are registered in the college.

III. LITERATURE SURVEY

Various research work purposed in this process basically consists of different plate localization, character segmentation and character recognition steps. Some of the purposed work made by different researcher is presented below.

G Ravi Kumar, P Prudhvi Kiran [1] In this they purposed of making automatic number plate recognition for android application. They made use of various image process libraries and technologies for plate detection, character segmentation and character recognition. They made use of Tesseract library for character recognition process.

Mutua Simon Mandi, Bernand Shibwabo, Kaibiru Mutua Raphael [2] In this they purposed of making automatic number plate recognition for car park management with the mobile devices. They also made use of basic image processing techniques but for character recognition process they made use of hardware devices like Nokia-Multiscanner, Cam-card, etc.

Anand Sumatilal Jain, Jayshree M. Kundargi [8] In this paper they purposed a novel algorithm for automatic number plate recognition for all-day traffic surveillance environment. A pre-processing step is applied to improve the performance of license plate localization and character segmentation in case of severe imaging conditions .In the last stage artificial neural network was used to categorize the input character of the license plate.

P.anishiya, prof. S. Mary joans [5] focused a number plate localization and recognition system for vehicles in Tamilnadu (India) is proposed. This system is developed based on digital images and can be easily applied to commercial car park systems for the use of documenting access of parking services, secure usage of parking houses and also to prevent car theft issues. The proposed algorithm is based on a combination of morphological operation with area criteria tests for number plate localization. Segmentation of the plate characters was achieved by the application of edge detectors, labelling and fill hole approach. The character recognition was accomplished with the aid of optical characters by the process of Template matching.

IV. CONTRIBUTION

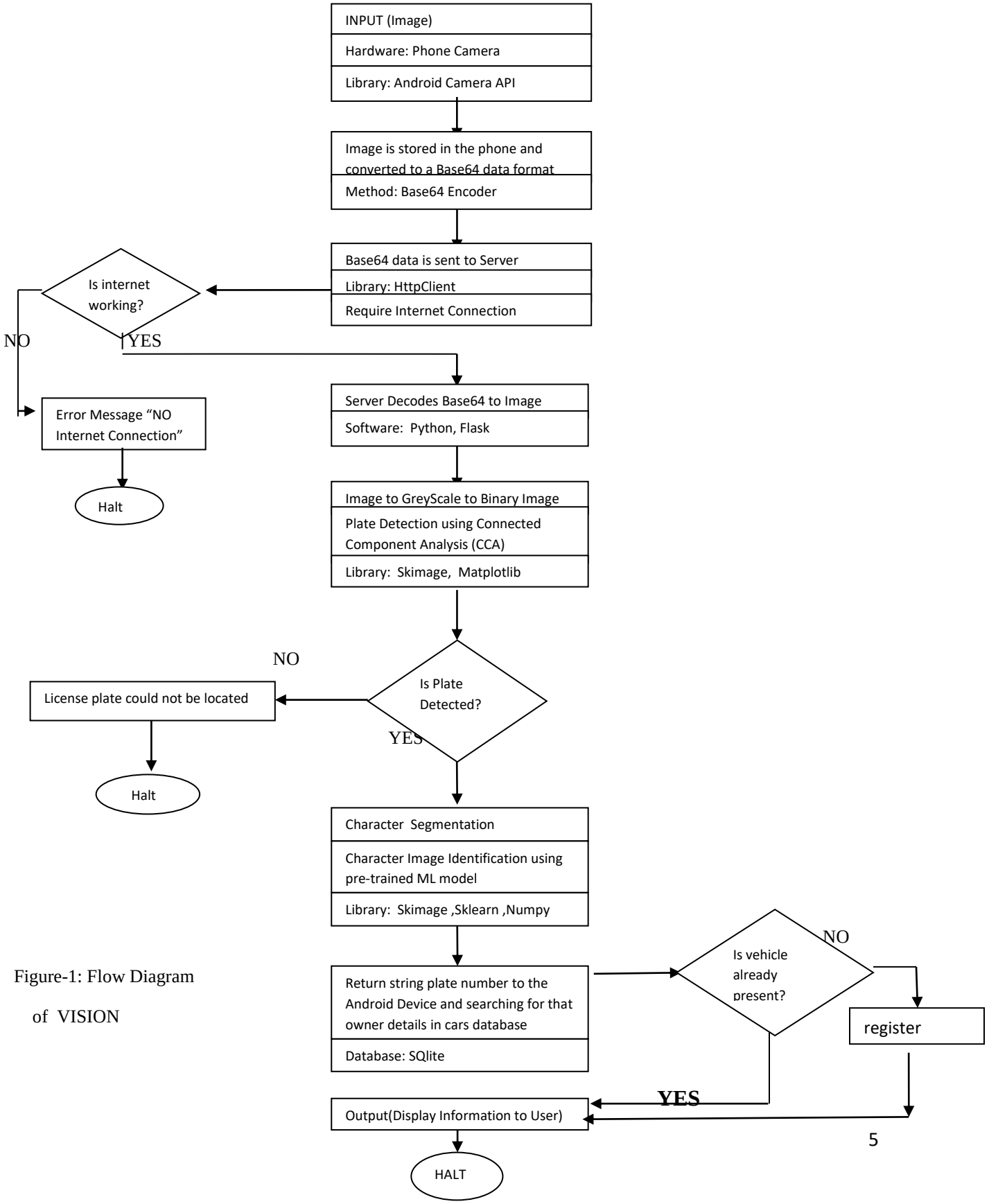


Figure-1: Flow Diagram of VISION

Our Project consist of two parts: Client Part and Server Part. The Client part consist of Android Application and the Server part consist of Vehicle Number Plate Recognition part.

IV.1. CLIENT PART

Android Platform:

Android platform is made for mobile devices such as smart phone and other handheld devices. The android platform consists of Linux kernel-based OS, a swift UI, frameworks, multimedia support etc.

Our Android Application is called V.I.S.I.O.N (Vehicle Identification System Including Ownership).This application is an integral part of this project as the user gets to interact with the application and make proper use of the features installed. The main motivation of our project was to provide the user with an easy interface with tons of features which would have a significant impact on the User Experience. We have tried our best to make this application bug-free and swiftly designed. Although, some bugs may appear in the future due to regular Android updates and Server Failure. Our Project starts with this very app and we get the results of queries in the application itself. In this short span of 6 months, we have tried to add numerous features and modules that a user may and will use in the future.

Proposed Design of Android Application

To implement Vehicle Number Plate detection on an Android application, we have provided a user friendly UI and we send the further Image processing part to the backend which handles detecting and recognizing the Number Plate through the image sent. Here, we are going to learn the basics of our Android Application and its UI components to which users interact on a daily basis.

Hardware and Software Environment Development

The android smart phone used in this research is Redmi Note 4 which is cost-friendly high-end smart phone with Android Oreo version(8.1).The smart phone has a 5.5 inch screen display with a resolution of 1080 pixels and 2Ghz Octa Processor with 4GB of RAM. The camera present has resolution of 13 mega-pixels.

The android application is developed using Android Studio 3.2 IDE using Java Development Kit(JDK). Although our present smart phone is high-end with latest android version, this app is compatible with minimum Android API 16(Android 4.1 version), i.e. Jellybean so that low-end smart phone users can also use this app for themselves.

System Design

In order to develop any application or system, the developer needs the guidance of how the application should work in order to make it work and deliver a good application. The design phase helps the developer to do this. The design leads the developer to the creation of various representation of the system design which server as a guide for the development of the app. The design can be divided into Interface Design and Process Design.

Process Design

The Process Design of the VISION starts with the Home-Page and then server and the Database helps us to retrieve the appropriate information to the user. The given figure shows the whole process of Information retrieval through VISION in this Project.

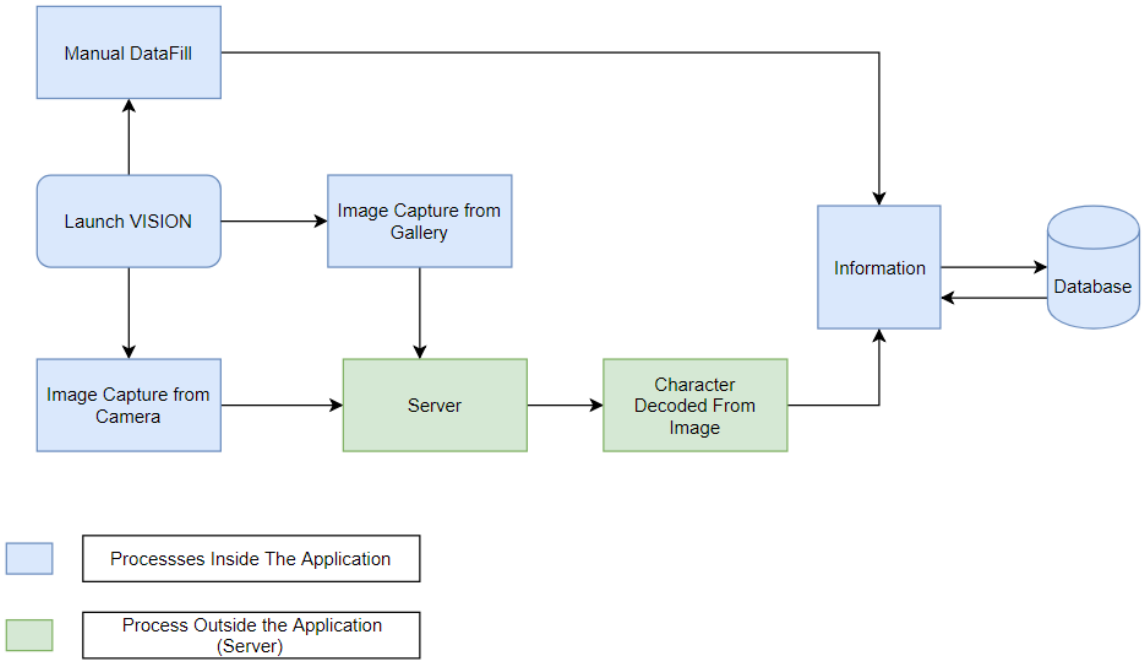


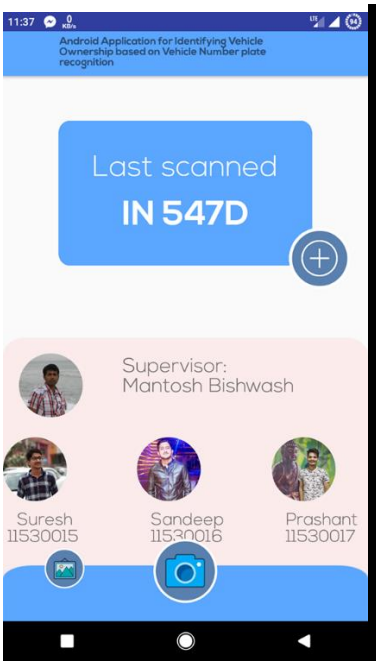
Figure-2: Flow Chart of Android Application in Android Smartphone and Python Server

Inside the android application, the Process of Vehicle Number Detection starts with Home Page app where we have the option of capturing image from gallery or capture vehicle image at real time with the Android Camera. We can also use the Manual DataFill Button to feed manually the data into the database where the information is stored. Apart from that, when sending the image containing the number plate to the server , we get the decoded characters from the server using Machine Learning Algorithms used in the Server Part. The characters obtained is used then to

decide whether it is already present in the database or not. If not, it is stored into the database, with additional information otherwise, the already present information surfaces out to the user.

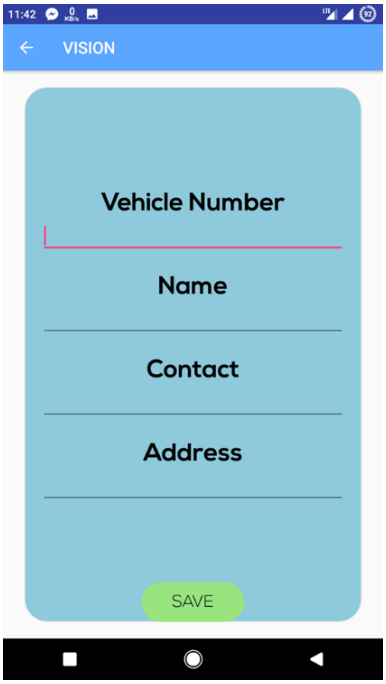
Graphical User Interface Design

User Interface design is an important part of an Android Application. It attracts users to its functions. The UI shows how an android application or any other application present itself to the users. It is very important that its easy, bug-free and nice to look at. Nowadays, Material is quickly replacing all traditional methods UI design. The Material Design Elements looks best, works fluently and most of all, feels awesome. We also have incorporated some material design elements in our app to provide the users, an awesome experience with our android application. In the given VISION app, we are showing some of the layouts of the app. The application include Home Page Window, Gallery access, manual input of information activity and pre-server activity which is shown in the following figure.



(a)

(b)



(c)

(d)

Figure-3: GUI, (a)Home Page Window

(b)Pre-server window

(c)Manual-Input Window

(d)Gallery Access windows

IV.2. SERVER PART

For making the server we used Flask framework of Python. Flask is the micro web framework of Python. It has the tools that easily enables us to link our client software which is android to our server side application.

Our server gets image in the base64 data format in the JSON content from the android device which is decoded using Base64 and ByteIO packages in python and finally converted into normal image format using Image function from PIL package. Finally our server is run using host port address 0.0.0.0.

Code snippet for the server:

```
application = Flask(__name__)
@app.route("/hello",methods=['POST'])
def predict():
        content=request.get_json()
        im=Image.open(BytesIO(base64.b64decode(content["image"])))
        im.save("required.jpg")
        /*Algorithm Code*/
if __name__ == '__main__':
        application.run(host='0.0.0.0')
```

Now the image processed through the server goes to our image processing and algorithm section.

Vehicle Number Plate Recognition Algorithm

In this section, implementation of vehicle license plate recognition algorithm on Android platform is presented. This algorithm basically consists of following steps which are explained below in detail.

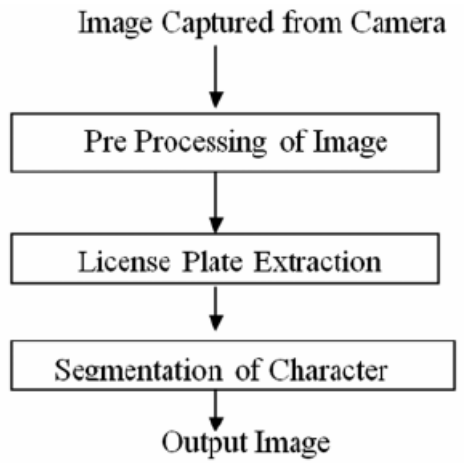


Figure 4: Block Diagram of implementation steps

1) Image Acquisition

This is the first step of implementing the algorithm. Image required for the further processing was captured through an android phone camera. One constraint to be noted while capturing the picture of the vehicle is that it should clearly capture the rear and back view of the vehicle with clearly visible number plate. Original picture which was captured was of RGB (Red, Green, Blue) color model. Our captured image may consists of various noise or distortion problem which was eliminated in our next step of image pre-processing.

Figure 5: Image Captured Using Android Camera

2) Image Pre-Processing

This is the next step to be performed which helps in removal of noise and contrast enhancement between the character and its background. Thus the end image after this was processed for license plate detection. At first captured RGB image was read as

grayscale image. This was achieved by using skimage library in Python and by multiplying each pixel in our image with 255. This will make our image pixel to be in the range of grey scale which is between 0 and 255. Then algorithm checks for the size of the image and resize it to maintain a feasible aspect ratio of 600*600 pixel value. Then grey scale image was converted in to binary image .This process is called image binarization which was achieved using threshold_otsu filter package of skimage library. Otsu method calculates appropriate threshold value for each tile of grayscale image and performs thresholding operation using the value resulting in binary image.

Pseudo Code For Image Pre-Processing

1. Read Image as grayscale
2. If size.width > 600
3. Resize image to feasible aspect ratio
4. else
5. Find the threshold value using Otsu method
6. Convert to binary image using threshold value

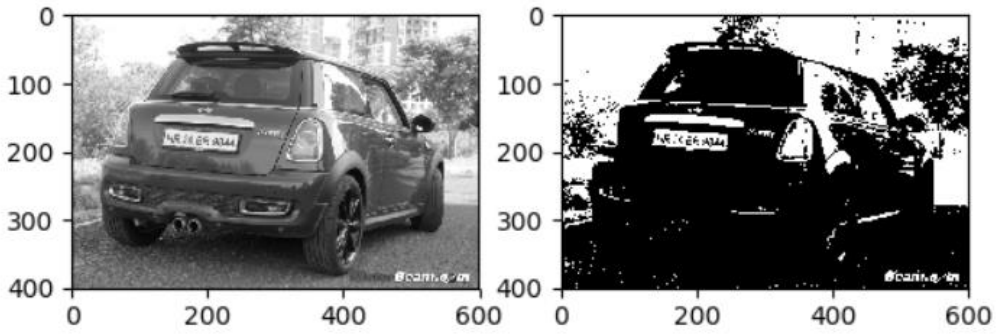


Figure 6: Grayscale to binary image conversion

3) License Plate Extraction

This is the third step of implementation of algorithm which helps us to detect number plate position in the image. It involves the following steps:

i. Connected Component Analysis

In this step we find the connected component patches from the image. Connected component is that component in which adjacent pixels are having the same value. Here we used measure package from skimage library to label all the connected component in the image. Then we made the assumption of plate dimension and extracted same dimensions component using the region props package from skimage library. Finally, we drew rectangle boundary to our plate like component and stored in the array of object.

ii. Vertical Projection

Plate component object obtained till now may also contain object of stickers or headlight. This objects need to be eliminated for which vertical projection is used. So, at first color inversion was done to make the background of the component to black and the letter to white. Finally, we count the white pixel value in the individual components and find the component which is having the largest average count which is finally our required plate component.

iii. Color Inversion Method

This is the process of inverting the color of the image. It is basically known as a negative image. It invert the color of the number plate and the character in the plate. Inversion is done by deducting each RGB color value from the maximum possible value which is ordinarily 255.

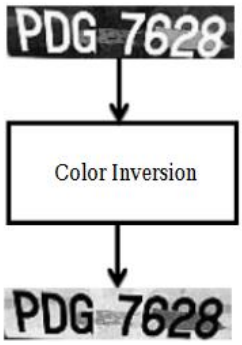


Figure 7: Color Inversion

Figure 8: Rectangular boundary around the detected plate

4) Character Segmentation

This is the fourth step for implementing the algorithm. Generally, in this step we are converting our license plate image in to many sub images each containing the individual character which can later be used for character recognition. Fixed pitch detection and chopping method is used for the same. The fixed pitch method tests the number plate's text lines to determine if they are fixed pitch. Fixed pitch is basically the character dimension that we have supposed. If fixed pitch text found in number plate, the fixed pitch method chops the number plate into characters using the pitch. So, after finding individual text we draw a rectangular boundary with the help of rectangle patch from skimage library. Then we resize the character image into 20*20 pixels for easier recognition later. We finally store those characters image object in to a character array for later steps of character recognition.

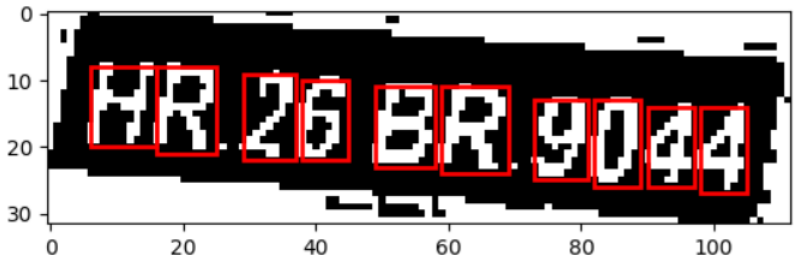


Figure 9: Character Segmentation using rectangular patches boundary

5) Character Recognition

This is the final step for the implementation of the algorithm for recognizing the character in the number plate. Machine Learning is used for the same to map the character image in to corresponding letters A's, B's, etc. There are two types of machine learning which are Supervised and Unsupervised Leaning. We used supervised learning for the implementation. In supervised learning, we train our model using the already labeled data. Supervised Learning is of two types: Classification and Regression. We used classifier for our experiment to classify our character image in to their corresponding letters or numbers. SVM (Support Vector Classifier) was used for the same since it was performing best for our model than any other classifier.

i. **Training Model** : For training, we set the directory path for our training data. Our training data consist of 10 images for each letter from A-Z and 0-9. Then we read the data and store it in to two array named image_data and target_data containing corresponding training data image and their target letter. We fit this two array in to our SVM model using kernel as linear and we finally save our model in a pickle file named finalized_model.sav . We can also use 4-fold cross validation concept to measure our accuracy. It indicates we are dividing our data in to 4 parts of which 1/4 is used for testing and ¾ for training.

Pseudo Coode for Training phase

1. Set path for training data directory
2. Read image and target data
3. Apply SVM model to the above data
4. Cross Validate the model
5. Save the model in pickle file

ii. Testing Phase :

For the above 4 fold cross-validation, we got result accuracy of 95% , 96% ,95%, and 99%. So, for any input data our recognition step would show the output somewhat like below. This will finally give us the string of license plate number which can later be used for ownership processing in Android.

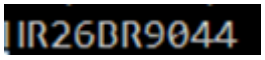


Figure-10: Testing Result

V. DATA FLOW DIAGRAM

A data flow diagram shows the way information flows through a process or system. It includes data inputs and outputs, data stores, and the various sub processes the data moves through. Thus, Data flow diagrams visually represent systems and processes that would be hard to describe in a chunk of text. Here we have used Yourdon and Coad (Y & C) methodology of DFD to discuss deeply about how data is handled in our application.

V.1. Level-0 DFD

Level-0 DFD is also known as context diagram that only contains one process node ("Process 0") that generalizes the relationship between the entire system and external entities.

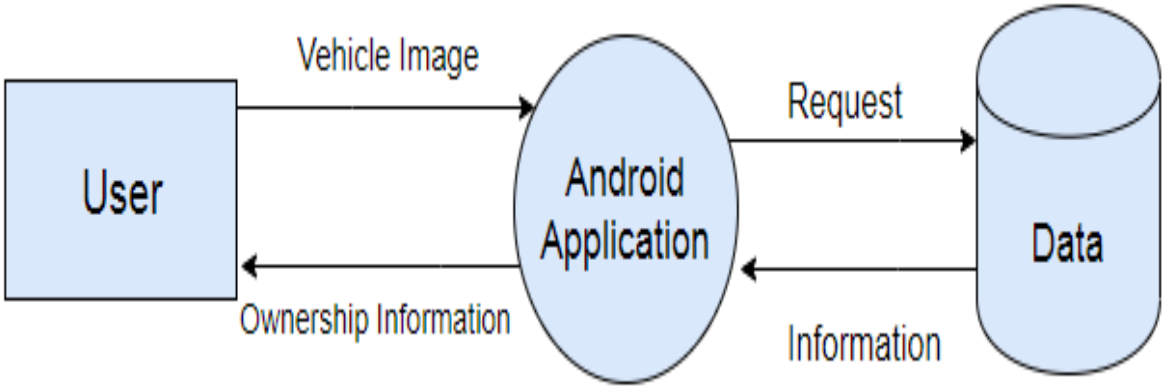


Figure-11: Level-0 DFD

V.2 Level-1 DFD

The Level 1 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the system as a whole.

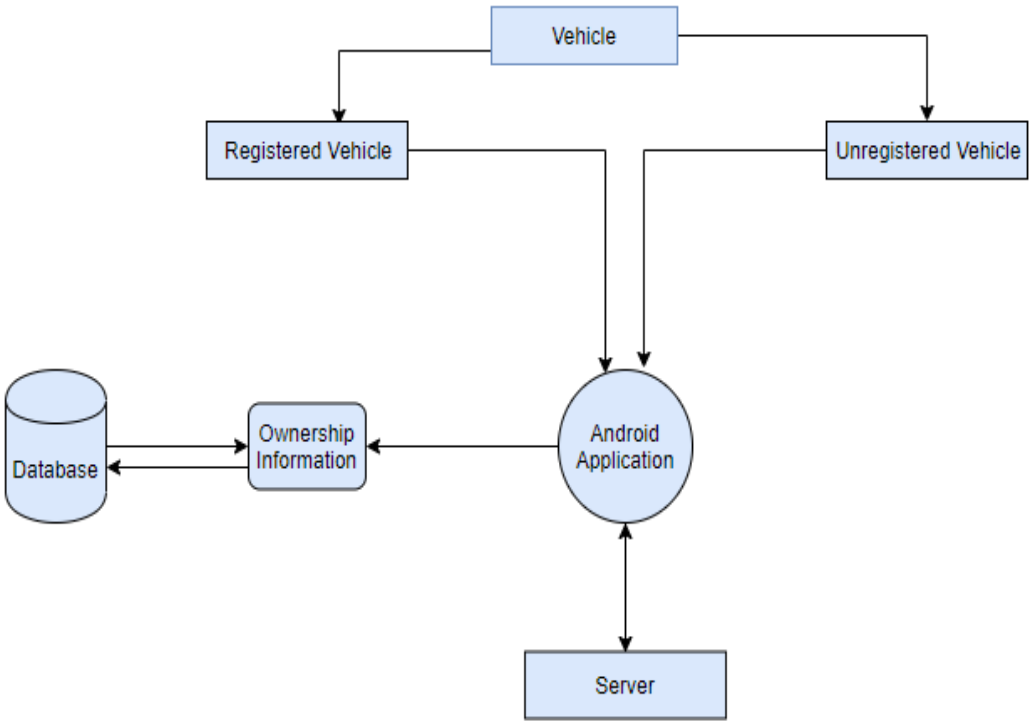


Figure-12: Level-1 DFD

V.3 Level-2 DFD

A level 2 data flow diagram (DFD) offers a more detailed look at the processes that make up an information system than a level 1 DFD does. It can be used to plan or record the specific makeup of a system which can then input the particulars of your own system.

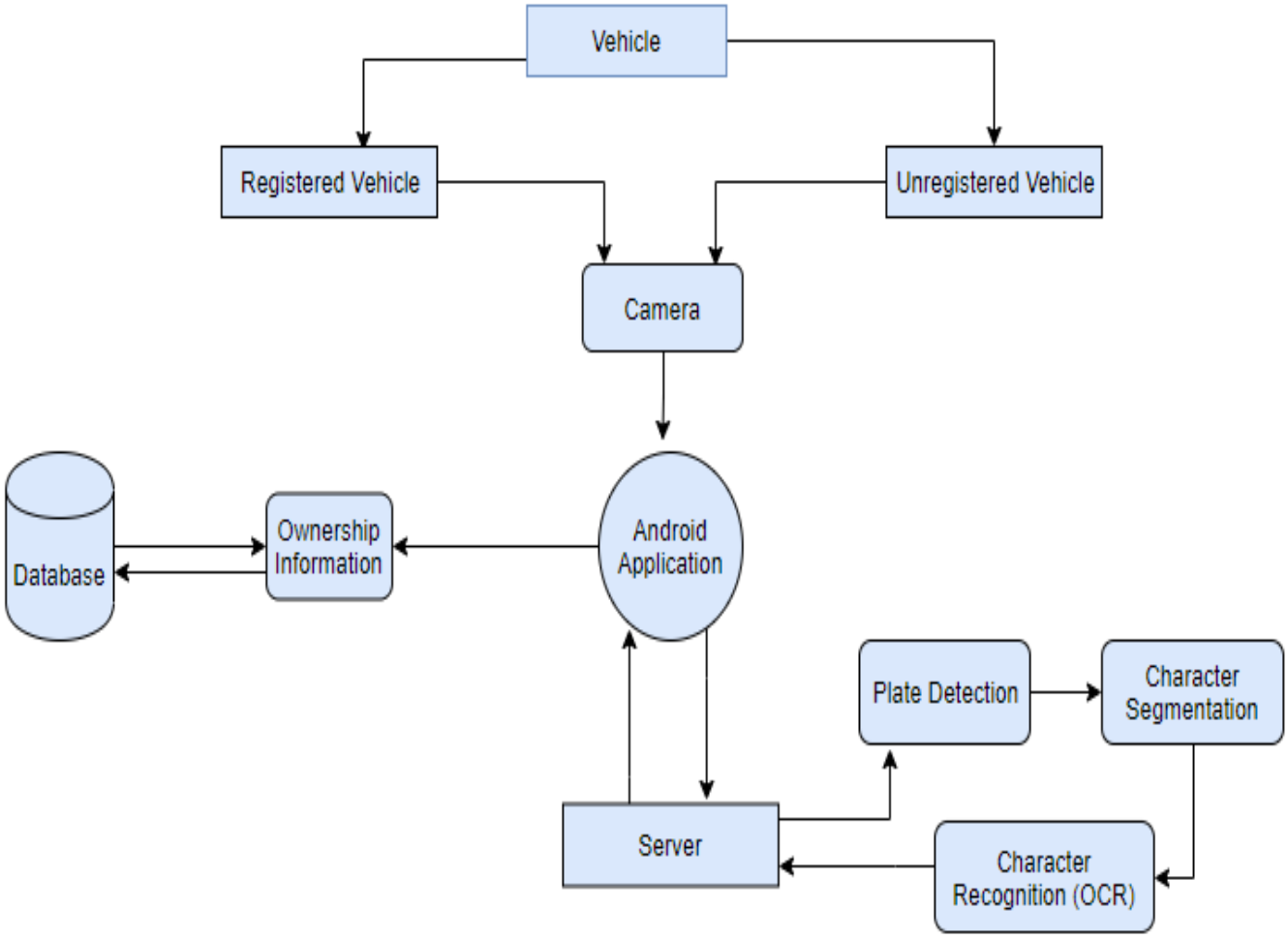


Figure 13: Level-2 DFD

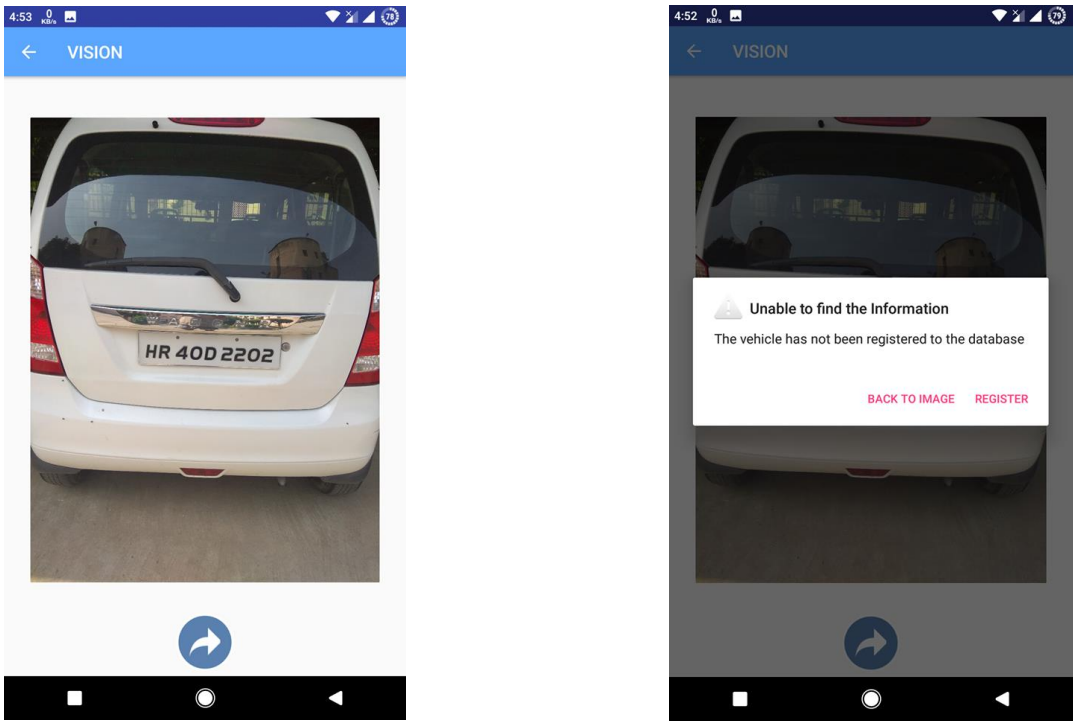
VI. RESULT/OBSERVATION

Result

We have been using the android application for weeks now, and below are some of the screenshots of the application when tested against real world applications.

The application starts with the home page with options provided to the user. Once the user decides to whether capture real time image or select the image from Gallery, the image is sent to the server for Character Decoding from the image. And then, the result is shown,

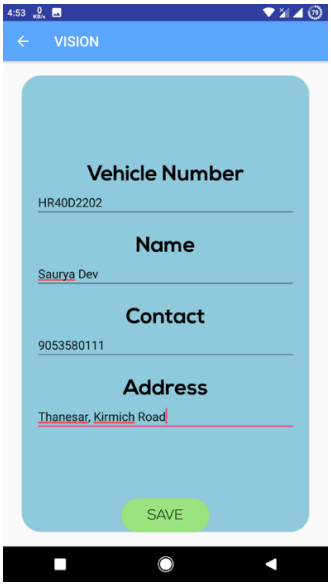
Below are some of the screenshots of the application in use:

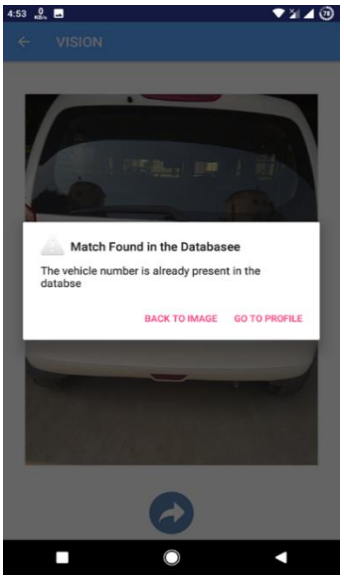


(a) (b)

Figure 14 :(a)Capturing a new Image through Camera (b) Notification on the basis of result obtained from server

At first, we are going to capture image from the android Camera which is linked to our VISION app. Here, we are going to capture a new image from the from the camera, i.e. Number plate whose database has not been recorded yet as in fig(a). Then , On Sending the image to the server, we get the following popup as shown in fig(b). Now, we save the Vehicle Plate Number into the database with additional information as shown in fig(c).Once we click the SAVE button, we save the information on the mobile application.





(c) (d)

Figure 15 :(c) Saving the new database (d)Proof that the data is saved

To check that the database information is saved, we again grab the image from Gallery and we again send the image to the server, same as in fig(a). Now, we can see the popup appears “Match already Found in the Database”. Hence Proved that we have saved the former image into the database as shown in fig(d). Now, if we click on PROFILE, we can view the database present on the database and view the database as shown in the given fig(e).

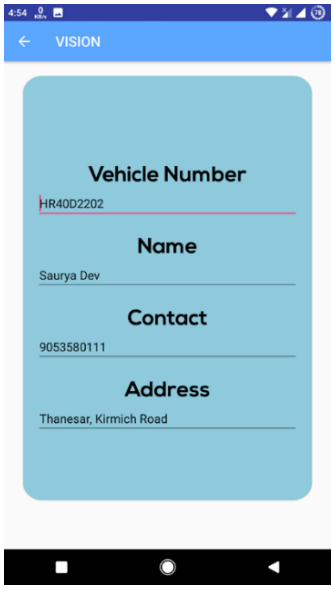


Figure 16: (e)Saved database

Observation

We tested 10 different Indian number plate and we got the following observation. Red colored number plate in the observation were found inaccurate or not detected.

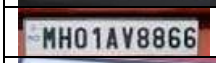
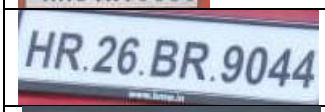
Licence plate	Recognize character	Actual Character
	HR08T7428	HR08T7428
	HR40D2202	HR40D2202
	KA02MK7259	KA02MK7259
	HR07H2074	HR07H2074
	MH0AV8866	MH01AV8866 (Inaccurate)
	HR26BR9044	HR26BR9044
	UP14DD0061	UP14DD0061
	No detected	MP04CC2688 (Inaccurate)
	MH12DE1433	MH12DE1433
	DL2CAY3126	DL2CAY3126

Table-1 : Observation Table

Total Number of License Plate = 10

Number of detected and Recognized Plate =8

Precision = (8/10)*100 %= 80%

So, the accuracy was found to be 80 % using our algorithm.

VII. CONCLUSION AND FUTUREPLAN

Android application for extracting the ownership detail of the vehicle based on number plate recognition has been developed successfully. The application is designed mainly to identify Indian number plates. The system is tested over several number plate images. We have tried our best to make the system accurate and reliable. However it can be further optimized using deep learning. Once the accuracy level is reached as per the expectation, we can implement the system in the college main entrance gate. The camera feature can be improved so that the vehicle number plate can be easily detected even during night time and low light conditions. We can create a cloud server in the college so that the database of all the vehicles including their ownership information can be stored so that it can be accessed from anywhere and by anyone who are authorized.

REFERENCES

[1]G Ravi Kumar,P Prudhvi Kiran,"Automatic Number Plate Recognition using Android Application" International journal of Innovative Research in Computer and Communication Engineering, Vol.5,Issue 4,pp. 7207-7211, April 2017

[2] Mutua Simon Mandi et.al., "An Automatic Number Plate Recognition System for Car Park Management", International Journal of Computer Applications, volume 175-No 7, pp. 36-42, October 2017.

[3] M.M Shidore,S.P. Narote, "Number Plate Recognition for Indian Vehicles" ,International Journal of Computer Science and Network Security,VOL. 11 No.2,pp. 143-145, Feb.2011.

[4] Karwal .H, & Girdhar. A, "Vehicle Number Plate Detection System for Indian Vehicles" , IEEE International Conference on Computational Intelligence & Communication Technology(CICT), 2015, Feb, pp.8-12

[5] P. Anishiya, Prof. S. Mary Joans, "Number Plate Recognition for Indian Cars Using Morphological Dilation and Erosion with the Aid Of Ocrs", International Conference on Information and Network Technology, IPCSIT vol.4 ,pp. 115-119, 2011.

[6] MY Aalsalem and WZ Khan. "Campus Sense-A Smart Vehicle Parking Monitoring and Management System using ANPR Cameras and Android Phones", Journal Editorial Board, Vol.5, Issue 2,pp.809-814,March 2016.

[7] Teddy Surya Gunawan, Abdul Mutholib, Mira Kartiwi, "Design of Automatic Number Plate Recognition on Android Smartphone Platform", Indonesian Journal of Electrical Engineering and Computer Science,Vol.5,No.1, January 2017,pp. 99-108.

[8] Anand Sumatilal Jain, Jayshree M.Kundargi, "Automatic Number Plate Recognition Using Artificial Neural Network", International Research Journal of Engineering and Technology(IRJET),Volume: 02, Issue 04, pp.1072-1077, July 2015.

APPENDIX

This project consist of both Python(For Server) and Java(For Android Application)

Python

server.py

```
from flask import Flask,request,jsonify
import os
from sklearn.externals import joblib
from PIL import Image
import pickle
import string
import base64
from io import BytesIO
import numpy as np
from skimage.transform import resize
from skimage import measure
from skimage.measure import regionprops
import matplotlib.patches as patches
import matplotlib.pyplot as plt
from skimage.io import imread
from skimage.filters import threshold_otsu
import matplotlib.pyplot as plt
from skimage.transform import resize
from flask import Flask,jsonify,request


from skimage import measure
from skimage.measure import regionprops
import matplotlib.pyplot as plt
import matplotlib.patches as patches
```

```
app = Flask(__name__)

def resize_required(resize_image):
    h, w = resize_image.shape
    ratio = float(w) / h
    # if the image is too big, resize by making the reference size 600 unit
    if w > 600:
        w = 600
        h = round(w / ratio)
        return resize(resize_image, (h, w))

    return resize_image

def verify_plate(validobject,reqobject):
    for each_object in validobject:
        h, w = each_object.shape
        each_candidate = threshold_invert(each_object)
        highavg = 0
        sumofwhitepixel = 0
        for column in range(w):
            sumofwhitepixel += sum(each_object[:, column])

        average = float(sumofwhitepixel) / w
        if average >= highavg:
            reqobject = each_object
            highavg = average

    return reqobject



def threshold_invert(grayimage):
    value = threshold_otsu(grayimage)
    return grayimage > value
```

```
# app is made using flask framework in python and method used is POST to get the image from the android device.
@app.route("/hello",methods=['POST'])
def hello():
                    # load the model
                print("Starting model")
                content=request.get_json()
                im=Image.open(BytesIO(base64.b64decode(content["image"])))
                im.save("image1.jpg")
                car_image = imread("image1.jpg", as_gray=True)
                print(car_image.shape)
                car_image=resize_required(car_image)
                grayscaleimage = car_image * 255
                fig, (ax1, ax2) = plt.subplots(1, 2)
                ax1.imshow(grayscaleimage, cmap="gray")
                value = threshold_otsu(grayscaleimage)
                binaryimage = grayscaleimage > value
                # print(binaryimage)
                ax2.imshow(binaryimage, cmap="gray")
                plt.show()

                # using connected component analysis of the binary image to find the connected component
                imagelabel = measure.label(binaryimage)
                # assumption is made for max and min height and weight of the licence plate dimension
                plate_dimensions = (0.03*imagelabel.shape[0], 0.08*imagelabel.shape[0], 0.15*imagelabel.shape[1], 0.3*imagelabel.shape[1])
                plate_dimensions2 = (0.08*imagelabel.shape[0], 0.2*imagelabel.shape[0], 0.15*imagelabel.shape[1], 0.4*imagelabel.shape[1])
                minh, maxh, minw, maxw = plate_dimensions
                platecordinates = []
                plateobjects = []
                acc_licence = []

                fig, (ax1) = plt.subplots(1)
                ax1.imshow(grayscaleimage, cmap="gray")
                flag =0
```

```
            # regionprops package in skimage makes a list of properties of the connected
image
            for region in regionprops(label_image):
                if region.area < 100:
                    #if the region area is very small less than 100 pixel area then we continue
and not consider it as plate
                    continue
                    # the bounding box coordinates
                minr, mincol, maxr, maxcol = region.bbox

                regionh = maxr - minr
                regionw = maxcol - mincol

                # checking that the region identified satisfies the condition of a typical license
plate
                if regionh >= minh and regionh <= maxh and regionw >= minw and regionw
<= maxw and regionw > regionh:
                    flag = 1
                    plateobjects.append(binaryimage[minr:maxr,
                                        mincol:maxcol])
                    platecordinates.append((minr, mincol,
                                            maxr, maxcol))
                    rectBorder = patches.Rectangle((mincol, minr), maxcol - mincol, maxr -
minr, edgecolor="red",
                                                   linewidth=2, fill=False)
                    ax1.add_patch(rectBorder)
            if(flag == 1):
                plt.show()




            if(flag==0):
                minh, maxht, minw, maxw = plate_dimensions2
                platecordinates = []
                plateobjects = []

                fig, (ax1) = plt.subplots(1)
                ax1.imshow(grayscaleimage, cmap="gray")
```

```
                # regionprops package in skimage makes a list of properties of the connected
image
                for region in regionprops(label_image):
                    if region.area < 50:
                        #if the region area is very small less than 100 pixel area then we continue
and not consider it as plate
                        continue
                        # the bounding box coordinates
                    minr, mincol, maxr, maxcol = region.bbox

                    regionh = maxr - minr
                    regionw = maxcol - mincol

                    # checking that the region identified satisfies the condition of a typical
license plate
                    if regionh >= minh and regionh <= maxh and regionw >= minw and regionw
<= maxw and regionw > regionh:
                        plateobjects.append(binaryimage[minr:maxr,
                                            mincol:maxcol])
                        platecordinates.append((minr, mincol,
                                                maxr, maxcol))
                        rectBorder = patches.Rectangle((mincol, minr), maxcol - mincol, maxr -
minr, edgecolor="red",
                                                       linewidth=2, fill=False)
                        ax1.add_patch(rectBorder)


                plt.show()
            number_of_candidates=len(plate_like_objects)
            print(number_of_candidates)
            if number_of_candidates == 0:
                print("Licence plate could not be located")
            if number_of_candidates == 1:
                acc_licence = threshold_invert(plate_like_objects[0])
            else:
                acc_licence = verify_plate(plate_like_objects,acc_licence)
```

```
            licenseplate = np.invert(acc_licence)

            labelplate = measure.label(license_plate)

            fig, ax1 = plt.subplots(1)
            ax1.imshow(license_plate, cmap="gray")
            # we are making assumption that height of the character is between 35-60% and
weight is between 5-15% of the detected plate component
            character_dimensions = (0.35*licenseplate.shape[0], 0.60*licenseplate.shape[0],
0.05*licenseplate.shape[1], 0.15*licenseplate.shape[1])
            minh, maxh, minw, maxw = character_dimensions

            characters = []
            counter=0
            column_list = []
            for regions in regionprops(labelled_plate):
               y0, x0, y1, x1 = regions.bbox
               regionht = y1 - y0
               regionw = x1 - x0

               if regionh > minh and regionh < maxh and regionw > minw and regionw <
maxw:
                  roi = licenseplate[y0:y1, x0:x1]
                  # below code is to draw rectangular border around the found character
                  rectborder = patches.Rectangle((x0, y0), x1 - x0, y1 - y0, edgecolor="red",
                                       linewidth=2, fill=False)
                  ax1.add_patch(rectborder)
                  # we resize image to 20*20 pixel for making it feasible during training
process
                  resizedchar = resize(roi, (20, 20))
                  characters.append(resizedchar)
                  #  keep track of the arrangement of the characters
                  column_list.append(x0)
            plt.show()



            filename = './finalized_model.sav'
            model = pickle.load(open(filename, 'rb'))
```

```
                print('loading model and predicting character ')
                classification_result = []
                for each_character in characters:
                    # converting image in to 1-D array using reshape and finally predicing using
the model.
                    each_character = each_character.reshape(1, -1);
                    result = model.predict(each_character)
                    classification_result.append(result)

                print('Classification result')
                print(classification_result)

                plate_string = ''
                for eachPredict in classification_result:
                    plate_string += eachPredict[0]

                # it's possible the characters are wrongly arranged
                # since that's a possibility, the column_list will be
                # used to sort the letters in the right order

                column_list_copy = column_list[:]
                column_list.sort()
                rightplate_string = ''
                for each in column_list:
                    rightplate_string += plate_string[column_list_copy.index(each)]
                print('License plate')
                print(rightplate_string)
                return rightplate_string

#below code is to create the server to connect to the local host and android device using the port
ID 0.0.0.0
if __name__ == '__main__':
    app.run(host='0.0.0.0')
```

Train.py

```
import os
import numpy as np
from sklearn.svm import SVC
from sklearn.model_selection import cross_val_score
from sklearn.externals import joblib
from skimage.io import imread
from skimage.filters import threshold_otsu

letters = [
          '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D',
          'E', 'F', 'G', 'H', 'J', 'K', 'L', 'M', 'N', 'P', 'Q', 'R', 'S', 'T',
          'U', 'V', 'W', 'X', 'Y', 'Z'
       ]

def readdata(tradirec):
   image_data = []
   target_data = []
   for eachletter in letters:
      for each in range(10):
         imagepath = os.path.join(tradirec, eachletter, eachletter + '_' + str(each) + '.jpg')
         # Read image of each character
         imgdet = imread(imagepath, as_grey=True)
         # Convert each character image in to binary for training phase
         binaryimage = imgdet < threshold_otsu(imgdet)
         # two dimensional image is converted in to 1D image so that it would be feasible for
training data 1*400
         flatimage = binaryimage.reshape(-1)
         image_data.append(flatimage)
         target_data.append(eachletter)

   return (np.array(image_data), np.array(target_data))

def crossvalid(model, num_of_fold, train_data, train_label):
   accresult = cross_val_score(model, train_data, train_label,
                               cv=num_of_fold)
   print("Validation Result for ", str(num_of_fold), " -fold")
```

```
    print(accuresult * 100)
trainingdir ='C:/Users/user/licence-plate-recognition/lpr/train20X20'
image_data, target_data = read_training_data(trainingdir)
svc_model = SVC(kernel='linear', probability=True)
crossvalid(svc_model, 4, image_data, target_data)
svc_model.fit(image_data, target_data)
import pickle
print("Saving Model")
filename = 'C:/Users/user/finalized_model.sav'
pickle.dump(svc_model, open(filename, 'wb'))
print("Saved Model")
```

Android Application Code

There are various codes present in the Android application.

Java Code:

Mainactivity.java

```
package com.example.prax.vis;

import android.content.Context;
import android.content.Intent;
import android.graphics.Bitmap;
import android.net.Uri;
import android.os.Handler;
import android.provider.MediaStore;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
```

```
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;

import java.io.ByteArrayOutputStream;


public class MainActivity extends AppCompatActivity {

   Mydatabase db;

   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      showActionBar();
         db  = new Mydatabase(this);
      Button gallery = (Button) findViewById(R.id.btngallery);
      Button cam = (Button) findViewById(R.id.btncam);
      Button add = (Button)findViewById(R.id.addmore);

      //db=new Mydatabase(this);
      if(!db.searchforserver("MM0AV8866"))
      for(int i=0;i<5;i++)
         db.addData();


      gallery.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            Intent intent = new Intent();
            intent.setType("image/*");
            intent.setAction(Intent.ACTION_GET_CONTENT);
            startActivityForResult(Intent.createChooser(intent, "Select Picture"), 1);
         }
      });
      cam.setOnClickListener(new View.OnClickListener() {
                        @Override
                        public void onClick(View v) {
                           Intent intent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
                           startActivityForResult(intent, 0);
```

```
                        }
        });
         //add button for adding new data
        add.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(MainActivity.this, DataFill.class);
                startActivity(intent);
            }
        });
}
    @Override
    protected void onActivityResult(int requestcode, int resultcode, Intent data){
        super.onActivityResult(requestcode,resultcode,data);
        if (requestcode == 1) {
            Uri uri = data.getData();
            Intent intent = new Intent(MainActivity.this, ImageCaptureGal.class);
            intent.putExtra("imageURI", uri.toString());
            startActivity(intent);
            finish();
        } else {
            Bitmap bMap = (Bitmap) data.getExtras().get("data");
            ByteArrayOutputStream bStream = new ByteArrayOutputStream();
            bMap.compress(Bitmap.CompressFormat.PNG, 100, bStream);
            byte[] byteArray = bStream.toByteArray();
            Intent anotherIntent = new Intent(this, ImageCapture.class);
            anotherIntent.putExtra("image", byteArray);
            startActivity(anotherIntent);
            finish();
        }
    }

    private void showActionBar(){
        LayoutInflater inflater =(LayoutInflater)
this.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View v =inflater.inflate(R.layout.actionbar_layout,null);
        final ActionBar bar=getSupportActionBar();
        bar.setDisplayHomeAsUpEnabled(false);
        bar.setDisplayShowHomeEnabled(false);
        bar.setDisplayShowCustomEnabled(true);
        bar.setDisplayShowTitleEnabled(false);
```

```
        bar.setCustomView(v);
    }
    boolean doubleBackToExitPressedOnce = false;
    @Override
    public void onBackPressed() {
        if (doubleBackToExitPressedOnce) {
            super.onBackPressed();
            return;
        }

        this.doubleBackToExitPressedOnce = true;
        Toast.makeText(this, "Please click back again to exit!", Toast.LENGTH_SHORT).show();

        new Handler().postDelayed(new Runnable() {

            @Override
            public void run() {
                doubleBackToExitPressedOnce = false;
            }
        }, 3000);
    }


}
```

MyDatabase.java

```
package com.example.prax.vis;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class Mydatabase extends SQLiteOpenHelper {
    public static final String DATABASE_NAME = "Informations.db";
    public static final String TABLE_NAME = "Information";
    public static final String colvehicleid = "id";
    public static final String colname = "name";
```

```
public static final String coladdress = "address";

public static final String colcontact = "contact";


public Mydatabase(Context context) {
   super(context, DATABASE_NAME, null, 1);

}

@Override
public void onCreate(SQLiteDatabase db) {
   String createTable = "CREATE TABLE IF NOT EXISTS "
         + TABLE_NAME +
         " ("+colvehicleid+" VARCHAR NOT NULL, "+ colname + " VARCHAR, "
          + colcontact + " VARCHAR, " + coladdress + " VARCHAR  )";
   db.execSQL(createTable);
}

@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
   db.execSQL("DROP TABLE IF EXISTS " + TABLE_NAME);
   onCreate(db);
}
public void addData(){
   SQLiteDatabase db = this.getWritableDatabase();
   ContentValues cv1 = new ContentValues();

   cv1.put(colvehicleid,"MM0AV8866");
   cv1.put(colname,"Prashant Kumar");
   cv1.put(coladdress,"Rajbiraj-4,Saptari, Nepal");
   cv1.put(colcontact,"9053580166");

   db.insert(TABLE_NAME, null, cv1);

   ContentValues cv2 = new ContentValues();

   cv2.put(colvehicleid,"NA 123456");
   cv2.put(colname,"Sandeep Bhandari");
   cv2.put(coladdress,"Butwal, Nepal");
   cv2.put(colcontact,"9053580167");
```

```
        db.insert(TABLE_NAME, null, cv2);

        ContentValues cv3 = new ContentValues();

        cv3.put(colvehicleid,"NA 789123");
        cv3.put(colname,"Suresh Paudel");
        cv3.put(coladdress,"Parbat, Nepal");
        cv3.put(colcontact,"90535801678");
        db.insert(TABLE_NAME, null, cv3);

    }

    public void savenewItem(String id,String name,String contact,String address){
        SQLiteDatabase db = this.getWritableDatabase();
        ContentValues cv3 = new ContentValues();

        cv3.put(colvehicleid,id);
        cv3.put(colname,name);
        cv3.put(coladdress,address);
        cv3.put(colcontact,contact);
        db.insert(TABLE_NAME, null, cv3);
    }

    public boolean searchforserver(String vehid){
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor data = db.rawQuery("SELECT * FROM "+ TABLE_NAME +" WHERE id LIKE
'%"+vehid+"%'",null);
        if(data.getCount()<=0)
        {
            data.close();
            return false;

        }
        data.close();
        return true;
    }
    public Cursor getListContents(String vehid){
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor data = db.rawQuery("SELECT * FROM "+ TABLE_NAME +" WHERE id LIKE
'%"+vehid+"%'",null);
        return data;
```

```
    }



}
```

DataFill.java

```
package com.example.prax.vis;

import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.support.v7.app.ActionBar;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RelativeLayout;
import android.widget.TextView;
import android.widget.Toast;

import java.lang.reflect.Array;
import java.util.ArrayList;

public class DataFill extends AppCompatActivity {

    String number;
    Mydatabase db;
    EditText addressid;
    EditText contactid;
    EditText nameid;


   public void savedmsg(){
      Toast.makeText(this, "Saved", Toast.LENGTH_SHORT).show();
```

```
    Intent i=new Intent(getApplicationContext(), MainActivity.class);
    i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    startActivity(i);
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_data_fill);
    showActionBar();
    final EditText vehid= (EditText)findViewById(R.id.labelvehidedittext);
     nameid= (EditText)findViewById(R.id.labelnameideditttext);
    contactid= (EditText)findViewById(R.id.labelcontactideditttext);
   addressid= (EditText)findViewById(R.id.labeladdressideditttext);
    Button  savebtn = (Button)findViewById(R.id.savebutton);

    db = new Mydatabase(this);


    number = getIntent().getStringExtra("number");

    vehid.setText(number);

    Boolean pres=db.searchforserver(number);

    if(pres){
        savebtn.setVisibility(View.GONE);
        ArrayList<String> list = new ArrayList<>();

        Cursor data= db.getListContents(number);

        if(data.getCount()==0){
            Toast.makeText(this,"NOT IN DB",Toast.LENGTH_LONG).show();
        }
        else{
            int i=0;
            while(data.moveToNext() && i<4){
                list.add(data.getString(i));
                Log.i("COl",Integer.toString(i));
                i++;
            }
```

```
        Log.i("LIST",list.toString());
        vehid.setText(list.get(0));
        nameid.setText(list.get(1));
        contactid.setText(list.get(2));
        addressid.setText(list.get(3));
      }
    }



    savebtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
        String newnumber = vehid.getText().toString();
        String name=nameid.getText().toString();
        String contact = contactid.getText().toString();
        String address = addressid.getText().toString();

        if(name.length()>0 && contact.length()>0 && address.length()>0){
          for(int i=0;i<5;i++)
            db.savenewItem(newnumber,name,contact,address);
          Log.i("Data","Saved");
          savedmsg();

        }
      }
    });









}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
  switch (item.getItemId()) {
```

```
        case android.R.id.home:
           // todo: goto back activity from here

           Intent intent = new Intent(this, MainActivity.class);
           intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP | Intent.FLAG_ACTIVITY_NEW_TASK);
           startActivity(intent);
           finish();
           return true;




        default:
           return super.onOptionsItemSelected(item);
     }
  }


  @Override
  public void onBackPressed() {
     super.onBackPressed();
     startActivity(new Intent(DataFill.this,MainActivity.class));
  }
  private void showActionBar(){
     final ActionBar bar=getSupportActionBar();
     bar.setDisplayHomeAsUpEnabled(true);
     bar.setDisplayShowHomeEnabled(true);
  }

}
```

XML Code:

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
```

```
tools:context=".MainActivity"
tools:layout_editor_absoluteY="81dp">

<RelativeLayout
   android:layout_width="match_parent"
   android:layout_height="240dp"
   android:layout_alignParentTop="true"
   android:layout_marginTop="27dp">

   <RelativeLayout
      android:layout_width="318dp"
      android:layout_height="181dp"
      android:layout_centerInParent="true"
      android:layout_marginTop="23dp"
      android:background="@drawable/lastscannedrectangle">

      <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_centerHorizontal="true"
         android:layout_marginTop="35dp"
         android:text="Last scanned"
         android:textColor="#FFFFFF"
         android:textSize="37dp"
         app:fontFamily="@font/nexa_l" />

      <TextView
         android:id="@+id/lastscanned"
         android:layout_width="wrap_content"
         android:layout_height="50dp"
         android:layout_alignParentBottom="true"
         android:layout_centerHorizontal="true"
         android:layout_marginBottom="41dp"
         android:fontFamily="@font/nexa_b"
         android:text="IN 547D"
         android:textColor="#FFFF"
         android:textSize="45sp" />
   </RelativeLayout>

   <RelativeLayout
      android:layout_width="wrap_content"
```

```
      android:layout_height="wrap_content"
      android:layout_alignParentRight="true"
      android:layout_alignParentBottom="true"
      android:layout_marginRight="22dp"
      android:background="@drawable/add_circ">

      <Button
        android:id="@+id/addmore"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:layout_centerInParent="true"
        android:background="@drawable/add" />
    </RelativeLayout>
</RelativeLayout>



<RelativeLayout
    android:layout_marginTop="265dp"
    android:layout_width="match_parent"
    android:layout_height="355dp"
    android:layout_alignParentBottom="true"
    android:background="@drawable/rectangle3"
    >
    <RelativeLayout
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_alignParentTop="true"
      android:id="@+id/supervisor">

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Supervisor:\nMantosh Bishwash"
        android:textSize="25dp"
        android:layout_marginLeft="150dp"
        android:layout_marginTop="20dp"
        android:fontFamily="@font/nexa_l"
        />
      <RelativeLayout
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true">
        <de.hdodenhof.circleimageview.CircleImageView
            xmlns:app="http://schemas.android.com/apk/res-auto"

            android:layout_width="80dp"
            android:layout_height="80dp"
            android:src="@drawable/mantoshsir"
            app:civ_border_width="0dp"
            android:layout_marginTop="20dp"
            android:layout_marginLeft="20dp"
            app:civ_border_color="#FF000000"
            />


    </RelativeLayout>


</RelativeLayout>
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="25dp"
    android:layout_below="@+id/supervisor"
   >
    <RelativeLayout
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_alignParentLeft="true">
        <de.hdodenhof.circleimageview.CircleImageView
            xmlns:app="http://schemas.android.com/apk/res-auto"

            android:layout_width="80dp"
            android:layout_height="80dp"
            android:src="@drawable/suresh"
            app:civ_border_width="0dp"
            app:civ_border_color="#FF000000"
            android:id="@+id/sureshdata"/>


        <TextView
```

```
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="10dp"
      android:textSize="21dp"
      android:text="   Suresh \n 11530015"
      android:fontFamily="@font/nexa_l"
      android:layout_below="@id/sureshdata"/>

</RelativeLayout>


<RelativeLayout
   android:layout_width="wrap_content"
   android:layout_height="match_parent"
   android:layout_centerInParent="true">
   <de.hdodenhof.circleimageview.CircleImageView
      xmlns:app="http://schemas.android.com/apk/res-auto"

      android:layout_width="80dp"
      android:layout_height="80dp"
      android:src="@drawable/sandeep"
      app:civ_border_width="0dp"
      app:civ_border_color="#FF000000"
      android:id="@+id/sandeepdata"/>

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginTop="10dp"
      android:textSize="21dp"
      android:text="   Sandeep \n   11530016"
      android:fontFamily="@font/nexa_l"
      android:layout_below="@id/sandeepdata"/>

</RelativeLayout>


<RelativeLayout
   android:layout_width="wrap_content"
   android:layout_height="match_parent"
   android:layout_alignParentRight="true">
```

```
        <de.hdodenhof.circleimageview.CircleImageView
            xmlns:app="http://schemas.android.com/apk/res-auto"

            android:layout_width="80dp"
            android:layout_height="80dp"
            android:src="@drawable/prashant"
            app:civ_border_width="0dp"
            app:civ_border_color="#FF000000"
            android:id="@+id/prashantdata"/>
        <!--<ImageView-->
            <!--android:layout_width="70dp"-->
            <!--android:layout_height="65dp"-->
            <!--android:layout_alignParentTop="true"-->
            <!--android:layout_centerHorizontal="true"-->
            <!--android:background="@drawable/blankprofile"-->
            <!--android:id="@+id/prashantdata"-->
            <!--/>-->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginTop="10dp"
            android:textSize="21dp"
            android:text="   Prashant \n   11530017"
            android:fontFamily="@font/nexa_l"
            android:layout_below="@id/prashantdata"/>
      </RelativeLayout>

    </RelativeLayout>
</RelativeLayout>

<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="103dp"
    android:layout_alignParentBottom="true"
    android:layout_marginBottom="0dp"
    app:layout_constraintTop_toTopOf="parent">

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:layout_alignParentBottom="true"
```

```
    android:background="@drawable/pr_rect">

  </RelativeLayout>

  <RelativeLayout
    android:layout_width="80dp"
    android:layout_height="80dp"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:background="@drawable/shutter_circ">

    <Button
      android:id="@+id/btncam"
      android:layout_width="65dp"
      android:layout_height="65dp"
      android:layout_centerInParent="true"
      android:background="@drawable/bluecam" />
  </RelativeLayout>


  <RelativeLayout
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:layout_alignParentStart="true"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginStart="52dp"
    android:layout_marginLeft="52dp"
    android:layout_marginTop="12dp"
    android:background="@drawable/shutter_circ">

    <Button
      android:id="@+id/btngallery"
      android:layout_width="45dp"
      android:layout_height="45dp"
      android:layout_centerInParent="true"
      android:background="@drawable/newgalleryicon" />


  </RelativeLayout>
</RelativeLayout>
```

```
</RelativeLayout>
```

Image_cap_gal.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:id="@+id/mybg"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:layout_editor_absoluteY="25dp">



   <RelativeLayout
      android:id="@+id/frameLayout"
      android:layout_width="match_parent"
      android:layout_height="573dp"
      android:layout_marginTop="30dp">

      <ImageView
         android:id="@+id/img_cap"
         android:layout_width="match_parent"
         android:layout_height="match_parent"
         android:layout_centerHorizontal="true"
         android:layout_marginBottom="10dp"
         android:layout_marginLeft="30dp"
         android:layout_marginRight="30dp" />
   </RelativeLayout>

   <!--<RelativeLayout-->
      <!--android:id="@+id/popup"-->
      <!--android:gravity="center"-->
      <!--android:layout_width="match_parent"-->
      <!--android:layout_height="wrap_content"-->
      <!--android:paddingLeft="30dp"-->
      <!--android:paddingRight="30dp"-->
      <!--android:layout_below="@+id/frameLayout"/>-->
```

```
    <RelativeLayout
        android:id="@+id/relativeLayout3"
        android:layout_alignParentBottom="true"
        android:layout_width="65dp"
        android:layout_height="65dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="6dp"
        android:background="@drawable/shutter_circ"
        android:layout_below="@+id/popup">

        <Button
            android:id="@+id/send"
            android:layout_width="35dp"
            android:layout_height="35dp"
            android:layout_centerHorizontal="true"
            android:layout_centerVertical="true"
            android:background="@drawable/send" />

    </RelativeLayout>

    <com.wang.avi.AVLoadingIndicatorView
        android:id="@+id/avi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        style="@style/AVLoadingIndicatorView.Large"
        android:visibility="visible"
        app:indicatorName="BallClipRotateMultipleIndicator"

        app:indicatorColor="@color/colorAccent"


    />



</RelativeLayout>
```