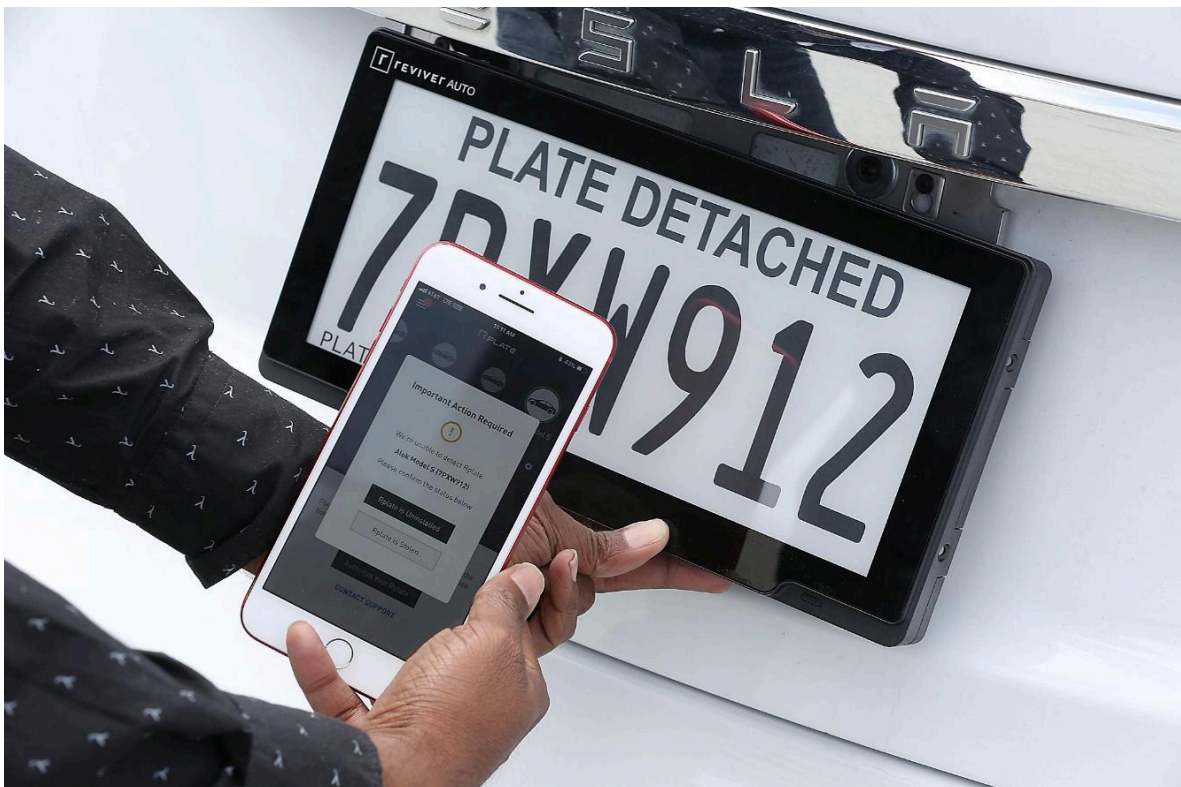


ABSTRACT

Automatic number plate recognition (ANPR) systems are becoming increasingly popular in private properties and government entities, as they can greatly increase productivity and efficiency. These systems use specialized cameras and software to automatically read and record the license plate numbers of vehicles entering or exiting a property.

One major benefit of ANPR systems is that they can significantly reduce the time and effort required to manually record and track vehicle movements. For example, in IIT Madras , an ANPR system can quickly and accurately record the entry and exit times of each vehicle, eliminating the need for a security guard to manually record this information. This can save a significant amount of time and allow security personnel to focus on other tasks.



ANPR systems can also improve security in private properties by providing a record of all vehicles entering and exiting the property. This can be especially useful in cases where there is a need to track the movements of

specific vehicles or individuals, as the system can quickly provide a record of when and where a particular vehicle was present.

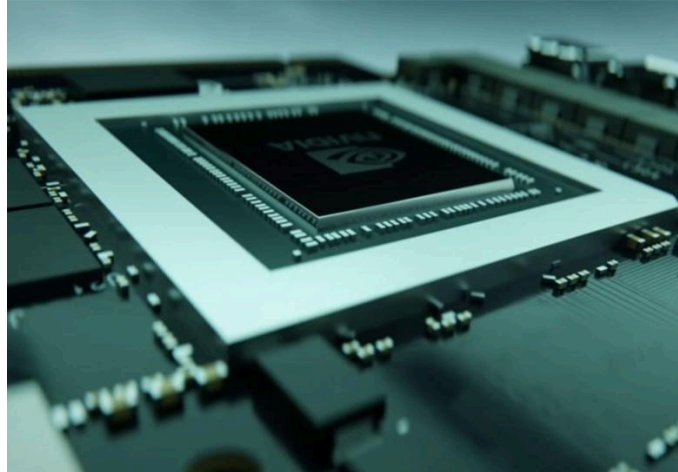
In addition to increasing productivity and improving security, ANPR systems can also provide a range of other benefits. For example, they can be used to collect data on vehicle movements within a property, which can be used to optimize traffic flow and improve the efficiency of parking management. They can also be used to enforce parking rules and fees, as the system can automatically identify and record any vehicles that are in violation of these rules.

Overall, ANPR systems have the potential to greatly increase productivity and efficiency in private properties, as well as improving security and providing a range of other benefits. As such, they are likely to become an increasingly important tool for property managers and owners looking to optimize the management of their properties.

REQUIREMENTS

HARDWARE

- ☐ Personal Computer with min 4 GB RAM (preferably 8 GB RAM)
- ☐ GPU is preferred for faster computation



SOFTWARE

- ☐ Python 3.8(IDLE)
- ☐ Command Prompt

INSTALLATION OF THE PROJECT

- ☐ To run the project, a PC is required with any Operating System (preferably Window-10/11) and a stable WI-FI connection is also required.
- ☐ Install the latest version of Python (preferably Version 3.0 or above).
 - Go to <https://www.python.org/downloads/>
 - Click on download Python 3.8.10 and proceed with the steps given in the setup to install it.
- ☐ Install the following libraries
 - OpenCV-python
 - datetime
 - easyocr
 - numpy
 - mysql.connector

- ✓ Open Command Prompt and type **pip install <library_name>** and click the Enter key.
- ✓ Do the same for the next 4 libraries, if it is already installed no worries.
- ✓ For installation of easyocr, go to <https://pytorch.org/> and follow the steps for the installation of PyTorch. In Compute Platform choose CUDA(version - according to the GPU) if a GPU is in the PC and then install easyocr in command prompt.

PyTorch Build	Stable (1.10.1)		Preview (Nightly)		LTS (1.8.2)		
Your OS	Linux		Mac		Windows		
Package	Conda		Pip		LibTorch		Source
Language	Python				C++ / Java		
Compute Platform	CUDA 10.2		CUDA 11.3		ROCm 4.2 (beta)		CPU
Run this Command:	pip3 install torch torchvision torchaudio						

```

Microsoft Windows [Version 10.0.22000.376]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Suresh>pip install opencv-python_

```

- ☐ Open Python Interpreter.
- ☐ Copy and paste the Source Code of the Program in the Python Interpreter and save it.
- ☐ Run the program to capture the license plate, recognize it and store it in the MySQL Database.
- ☐ Press 's' to save the image of the license plate.
- ☐ Press 'e' to enter the MySQL queries.
- ☐ Press 'q' to exit the program.

BUILT-IN FUNCTIONS

I. len()

The len() function returns the length (the number of items) of an object. The argument may be a sequence (such as a string, bytes, tuple, list, or range) or a collection (such as a dictionary, set, or frozen set).

II. print()

The print() function prints the specified message to the screen or other standard output device. The message can be a string or any other object, the object will be converted into a string before being written to the screen.

III. str()

The str() function returns the string version of the given object.

The syntax of str() is:

```
str(object, encoding='utf-8', errors='strict')
```

The str() method takes three parameters:

object - The object whose string representation is to be returned. If not provided, returns the empty string.

encoding - Encoding of the given object. Defaults of UTF-8 when not provided.

errors - Response when decoding fails. Defaults to 'strict'.

IV. Exception

Base class for all exceptions

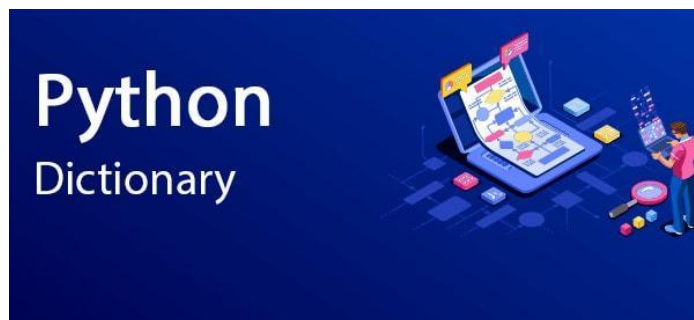
V. ord()

The ord() function returns an integer representing the Unicode character.

DATA STRUCTURES

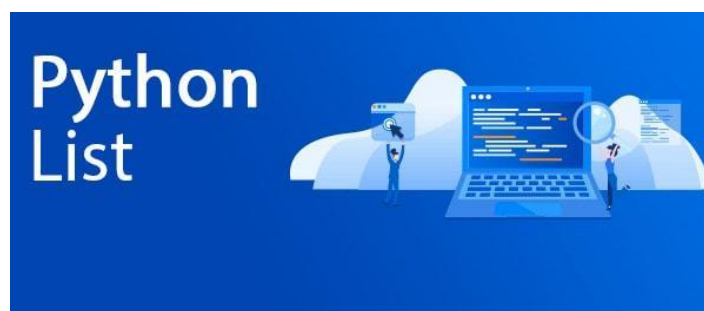
I. DICTIONARY

- Dictionaries are used to store data values in key: value pairs.
- Each key is separated from its value by a colon (:), the items are separated by commas, and the entire dictionary is enclosed in curly braces.
- In this project, it had been used to store key: value pairs of states in India. The state(value) is retrieved by using the key and it is stored in the MySQL database



II. LIST

- A list is a collection of values or an ordered sequence of values/items.
- Elements of a list are enclosed in square brackets [], separated by commas.
- In this project, it had been used to store similar regions of an image while checking the image similarity, and it had also been used to store the cropped license plate.



LIBRARIES

I. os

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system dependent functionality. The `*os*` and `*os.path*` modules include many functions to interact with the file system. `os.listdir` was used to get the list of all files and directories in the specified directory. If the directory is not specified, then the list of files and directories in the current working directory will be returned.

II. OpenCV (cv2)



OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. The functions that have been used in this project are `cv2.ORB_create()`, `cv2.BFMatcher()`, `cv2.NORM_HAMMING`, `cv2.imwrite()`, `cv2.imread()`, `cv2.CascadeClassifier`, `cv2.VideoCapture()`, `cv2.cvtColor()`, `cv2.destroyAllWindows()`, `cv2.rectangle()`, `cv2.putText()`, `cv2.imshow()`, `cv2.waitKey()`.

`cv2.ORB_create()` is a fusion of the FAST keypoint detector and BRIEF descriptor with some added features to improve the performance. `cv2.BFMatcher()` is used for matching the features of the first image

with another image, it takes one descriptor of the first image and matches to all the descriptors of the second image, and then it goes to the second descriptor of the first image and matches to all the descriptor of the second image and so on.

`cv2.NORM_HAMMING` is used for binary string-based descriptors like ORB, BRIEF, BRISK, etc which uses Hamming distance as measurement.

`cv2.imwrite()` method is used to save an image to any storage device.

`cv2.imread()` method loads an image from the specified file. If the image cannot be read (because of the missing file, improper permissions, unsupported or invalid format) then this method returns an empty matrix.

`cv2.CascadeClassifier()` is a method for combining increasingly more complex classifiers like AdaBoost in a cascade that allows negative input (non-license plate) to be quickly discarded while spending more computation on promising regions (license plate). It significantly reduces the computation time and makes the process more efficient.

`cv2.VideoCapture()` to get a video capture object from the camera.

`cv2.cvtColor()` method is used to convert an image from one color space to another.

`cv2.rectangle()` method is used to draw a rectangle on any image.

`cv2.putText()` method is used to draw a text string on any image.

`cv2.imshow()` method is used to display an image in a window. The window automatically fits the image size.

`cv2.waitKey()` function of Python OpenCV allows users to display a window for given milliseconds or until any key is pressed. It takes time in milliseconds as a parameter and waits for the given time to destroy the window, if 0 is passed in the argument it waits till any key is pressed.

`destroyAllWindows()` function allows users to destroy all windows at any time. It doesn't take any parameters and doesn't return anything.

III. Datetime

datetime library is used to manipulate date and time in Python.

now() function will return the current local date and time and it had been used as a name for the permanent image.

IV. Easyocr

easyocr is a Python package used to perform Optical Character Recognition. easyocr is implemented using Python and the PyTorch library. If there is a CUDA-capable GPU, the underlying PyTorch deep learning library can speed up the text detection and OCR speed tremendously. easyocr can OCR text in 58 languages



V. mysql.connector

mysql.connector provides connectivity to the MySQL server, enables Python programs to access MySQL databases and to implement MySQL queries.

SOURCE CODE

<https://pythonsandbox.dev/embed/dio7gxh93kip?file=main.py>

```
import os

import cv2

from datetime import datetime as dt

import easyocr

import mysql.connector

count=0                                #initialized for saving temp images


states={"AN":"Andaman and Nicobar", "AP":"Andhra Pradesh",
"AR":"Arunachal Pradesh", "AS":"Assam", "BR":"Bihar", "CH":"Chandigarh",
"DN":"Dadra and Nagar Haveli", "DD":"Daman and Diu", "DL":"Delhi",
"GA":"Goa", "GJ":"Gujarat", "HR":"Haryana", "HP":"Himachal Pradesh",
"JK":"Jammu and Kashmir", "KA":"Karnataka", "KL":"Kerala",
"LD":"Lakshadweep", "MP":"Madhya Pradesh", "MH":"Maharashtra",
"MN":"Manipur", "ML":"Meghalaya", "MZ":"Mizoram", "NL":"Nagaland",
"OD":"Odissa", "PY":"Pondicherry", "PN":"Punjab", "RJ":"Rajasthan",
"SK":"Sikkim", "TN":"TamilNadu", "TR":"Tripura", "UP":"Uttar Pradesh",
"WB":"West Bengal", "CG":"Chhattisgarh", "TS":"Telangana",
"JH":"Jharkhand", "UK":"Uttarakhand"}


def orb_sim(img1, img2):                #func for checking similarity of images

    orb = cv2.ORB_create()

    kp_a, desc_a = orb.detectAndCompute(img1, None)

    kp_b, desc_b = orb.detectAndCompute(img2, None)

    bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)

    matches = bf.match(desc_a, desc_b)    #perform matches.
```

```

similar_regions = [i for i in matches if i.distance < 50]

if len(matches) == 0:

    return 0

return len(similar_regions) / len(matches)

def checkperm():

    is_similar = False

    directory = os.listdir("D:\\Image_perm")

    if len(directory) == 0:

        image_path = "D:\\Number_plates\\Image"+str(count)+".jpg"

        img_save= cv2.imread(image_path)

        cv2.imwrite("D:\\Image_perm\\"+str_rep+".jpg",img_save)

    else:

        for images in os.listdir("D:\\Image_perm"):

            img1=cv2.imread("D:\\Number_plates\\Image"+str(count)+".jpg")

            img2=cv2.imread("D:/Image_perm/"+ images)

            similarity = orb_sim(img1,img2)

            if similarity <= 0.60:

                pass

            else:

                is_similar = True

                break

    return is_similar

def ocr(c):

    img = cv2.imread('D:\\Number_plates\\Image'+str(c)+' .jpg')

```

```

reader = easyocr.Reader(['en'])

result = reader.readtext(img)

text = result[0][-2]

text= ".join(e for e in text if e.isalnum())

text_cap = text.upper()

print(text_cap)

stat = text_cap[0:2]

try:

    state = states[stat]

    print('Car Belongs to',state)

except:

    state = "SNR"

    print('State not recognised!!')

myCursor.execute("INSERT INTO ANPR (PLATE_CHAR, STATE) VALUES
(%s, %s)", (text_cap, state))

mycon.commit()

try:

mycon=mysql.connector.connect(host='localhost',user='root',passwd='V')

myCursor=mycon.cursor()

myCursor.execute("CREATE DATABASE IF NOT EXISTS ANPR_SYS")

myCursor.execute("USE ANPR_SYS")

myCursor.execute("CREATE TABLE IF NOT EXISTS ANPR (DATE_TIME
TIMESTAMP NOT NULL DEFAULT current_timestamp on update
current_timestamp,PLATE_CHAR VARCHAR(15),STATE VARCHAR(30))")

except Exception as e:

```

```

print(e)

plateCascade = cv2.CascadeClassifier(cv2.data.harcascades +
"haarcascade_russian_plate_number.xml")

minArea = 500

cap =cv2.VideoCapture(0)

while True:

    success , img = cap.read()

    imgGray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    numberPlates = plateCascade.detectMultiScale(imgGray, 1.1, 4)
#number plate recognition

    for (x, y, w, h) in numberPlates:

        area = w*h

        if area > minArea:

            cv2.rectangle(img, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.putText(img,"NumberPlate",(x,y-5),cv2.FONT_HERSHEY_COMPLEX,1,(0,
0,255),2)

            imgRoi = img[y:y+h,x:x+w]                #crop number plate

            cv2.imshow("ROI",imgRoi)

cv2.imshow("Result",img)

if cv2.waitKey(1) & 0xFF ==ord('s'):#press s to save image in temp director

    cv2.imwrite("D:\\Number_plates\\Image"+str(count)+".jpg",imgRoi)

    c = checkperm()

    if c == False:

        now = dt.now()

        str_now = str(now)

```

```

    str_rep = str_now.replace(':', '.')

    image_path = "D:\\Number_plates\\Image"+str(count)+".jpg"

    img_save= cv2.imread(image_path)

    cv2.imwrite("D:\\Image_perm\\"+str_rep+".jpg",img_save)

    print("image saved")

    cv2.rectangle(img,(0,200),(640,300),(0,255,0),cv2.FILLED)

    cv2.putText(img,"Scan
Saved",(15,265),cv2.FONT_HERSHEY_COMPLEX,2,(0,0,255),2)

    cv2.imshow("Result",img)

    cv2.waitKey(500)

    ocr(count)

    count+=1

if cv2.waitKey(1) & 0xFF ==ord('e'):          #press e to write queries

    try:

        qry = input("ENTER SELECT TABLE QUERY:")

        myCursor.execute(qry)

        myr = myCursor.fetchall()

        for x in myr:

            print(x)

    except Exception as e:

        print(e)

if cv2.waitKey(1) & 0xFF ==ord('q'):#press q to quit and close the windows

    break

cap.release()

cv2.destroyAllWindows()

```

OUTPUT

Python

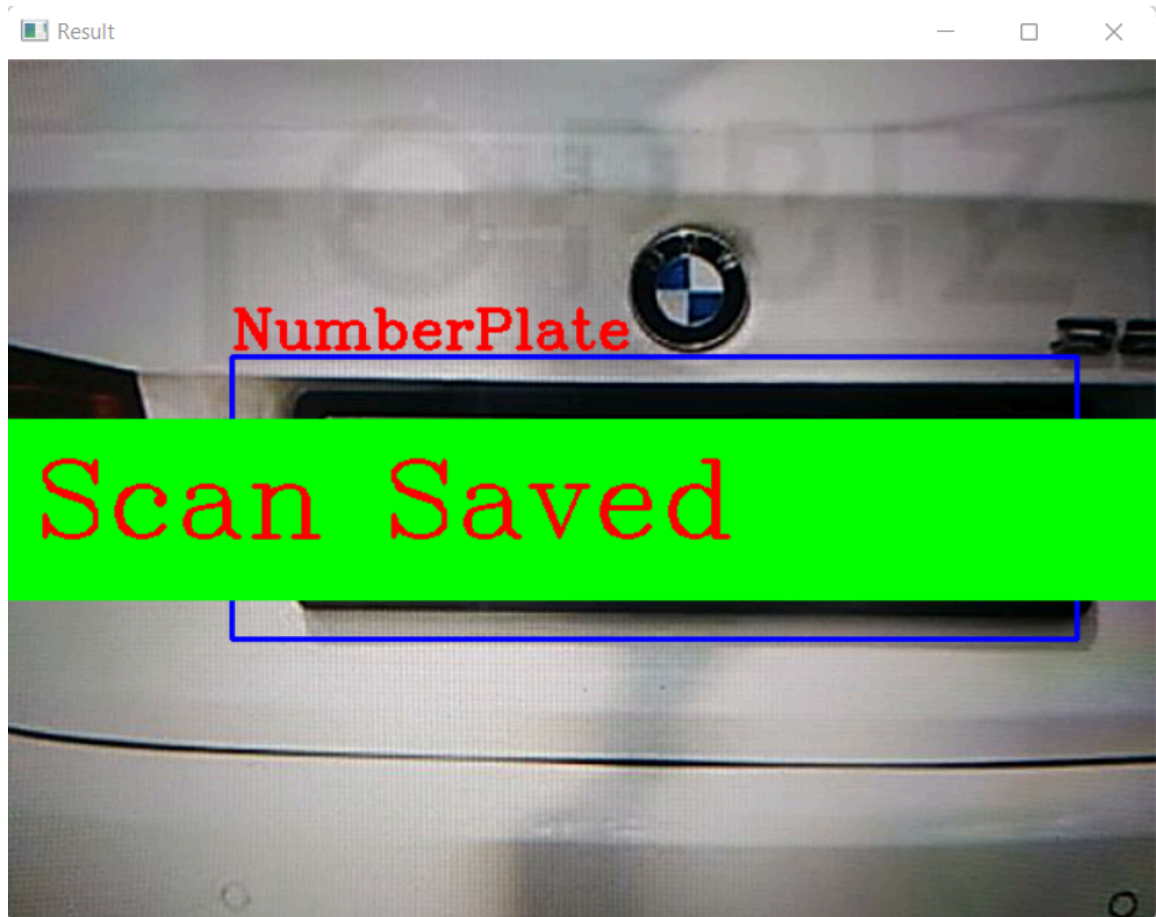
When the program runs and the webcam detects a license plate



The license plate is cropped and displayed



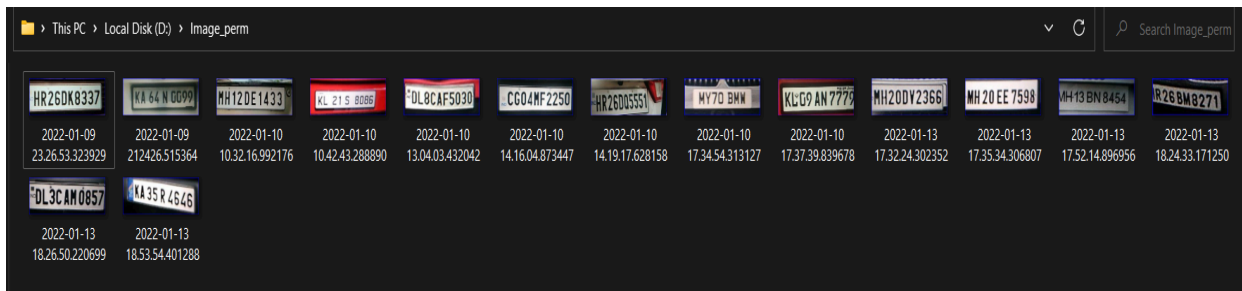
When 's' is pressed the cropped license plate is saved in Number_plates (temp) directory



Number_plates (temp) directory



Image_perm directory



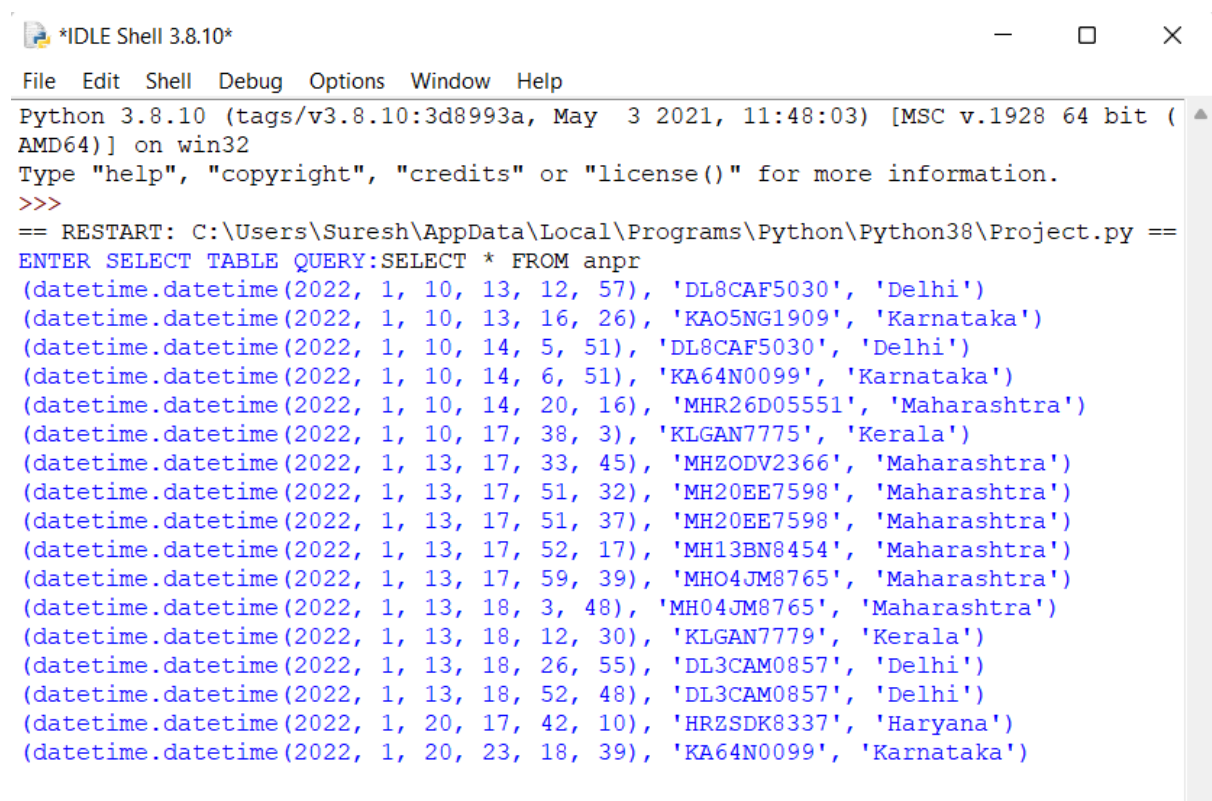
OCR and State of the license plate

```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Suresh\AppData\Local\Programs\Python\Python38\Project.py ==
KA64N0099
Car Belongs to Karnataka
|
```

When 'e' is pressed MySQL Queries can be executed



```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Suresh\AppData\Local\Programs\Python\Python38\Project.py ==
ENTER SELECT TABLE QUERY:
```



```
*IDLE Shell 3.8.10*
File Edit Shell Debug Options Window Help
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Suresh\AppData\Local\Programs\Python\Python38\Project.py ==
ENTER SELECT TABLE QUERY:SELECT * FROM anpr
(datetime.datetime(2022, 1, 10, 13, 12, 57), 'DL8CAF5030', 'Delhi')
(datetime.datetime(2022, 1, 10, 13, 16, 26), 'KAO5NG1909', 'Karnataka')
(datetime.datetime(2022, 1, 10, 14, 5, 51), 'DL8CAF5030', 'Delhi')
(datetime.datetime(2022, 1, 10, 14, 6, 51), 'KA64N0099', 'Karnataka')
(datetime.datetime(2022, 1, 10, 14, 20, 16), 'MHR26D05551', 'Maharashtra')
(datetime.datetime(2022, 1, 10, 17, 38, 3), 'KLGAN7775', 'Kerala')
(datetime.datetime(2022, 1, 13, 17, 33, 45), 'MHZODV2366', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 51, 32), 'MH20EE7598', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 51, 37), 'MH20EE7598', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 52, 17), 'MH13BN8454', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 59, 39), 'MHO4JM8765', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 18, 3, 48), 'MHO4JM8765', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 18, 12, 30), 'KLGAN7779', 'Kerala')
(datetime.datetime(2022, 1, 13, 18, 26, 55), 'DL3CAM0857', 'Delhi')
(datetime.datetime(2022, 1, 13, 18, 52, 48), 'DL3CAM0857', 'Delhi')
(datetime.datetime(2022, 1, 20, 17, 42, 10), 'HRZSDK8337', 'Haryana')
(datetime.datetime(2022, 1, 20, 23, 18, 39), 'KA64N0099', 'Karnataka')
```

When 'q' is pressed all windows are closed and the program gets killed



```
Python 3.8.10 (tags/v3.8.10:3d8993a, May 3 2021, 11:48:03) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:\Users\Suresh\AppData\Local\Programs\Python\Python38\Project.py ==
ENTER SELECT TABLE QUERY:SELECT * FROM anpr
(datetime.datetime(2022, 1, 10, 13, 12, 57), 'DL8CAF5030', 'Delhi')
(datetime.datetime(2022, 1, 10, 13, 16, 26), 'KAO5NG1909', 'Karnataka')
(datetime.datetime(2022, 1, 10, 14, 5, 51), 'DL8CAF5030', 'Delhi')
(datetime.datetime(2022, 1, 10, 14, 6, 51), 'KA64N0099', 'Karnataka')
(datetime.datetime(2022, 1, 10, 14, 20, 16), 'MHR26D05551', 'Maharashtra')
(datetime.datetime(2022, 1, 10, 17, 38, 3), 'KLGAN7775', 'Kerala')
(datetime.datetime(2022, 1, 13, 17, 33, 45), 'MHZODV2366', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 51, 32), 'MH20EE7598', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 51, 37), 'MH20EE7598', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 52, 17), 'MH13BN8454', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 17, 59, 39), 'MHO4JM8765', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 18, 3, 48), 'MHO4JM8765', 'Maharashtra')
(datetime.datetime(2022, 1, 13, 18, 12, 30), 'KLGAN7779', 'Kerala')
(datetime.datetime(2022, 1, 13, 18, 26, 55), 'DL3CAM0857', 'Delhi')
(datetime.datetime(2022, 1, 13, 18, 52, 48), 'DL3CAM0857', 'Delhi')
(datetime.datetime(2022, 1, 20, 17, 42, 10), 'HRZSDK8337', 'Haryana')
(datetime.datetime(2022, 1, 20, 23, 18, 39), 'KA64N0099', 'Karnataka')
>>> |
```

MySQL

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| anpr_sys |
| mysql |
| test |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> USE anpr_sys;
Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_anpr_sys |
+-----+
| anpr |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT * FROM anpr;
```

DATE_TIME	PLATE_CHAR	STATE
2022-01-10 13:12:57	DL8CAF5030	Delhi
2022-01-10 13:16:26	KA05NG1909	Karnataka
2022-01-10 14:05:51	DL8CAF5030	Delhi
2022-01-10 14:06:51	KA64N0099	Karnataka
2022-01-10 14:20:16	MHR26D05551	Maharashtra
2022-01-10 17:38:03	KLGAN7775	Kerala
2022-01-13 17:33:45	MHZ0DV2366	Maharashtra
2022-01-13 17:51:32	MH20EE7598	Maharashtra
2022-01-13 17:51:37	MH20EE7598	Maharashtra
2022-01-13 17:52:17	MH13BN8454	Maharashtra
2022-01-13 17:59:39	MH04JM8765	Maharashtra
2022-01-13 18:03:48	MH04JM8765	Maharashtra
2022-01-13 18:12:30	KLGAN7779	Kerala
2022-01-13 18:26:55	DL3CAM0857	Delhi
2022-01-13 18:52:48	DL3CAM0857	Delhi
2022-01-20 17:42:10	HRZSDK8337	Haryana
2022-01-20 23:18:39	KA64N0099	Karnataka

```
17 rows in set (0.00 sec)
```

CONCLUSION

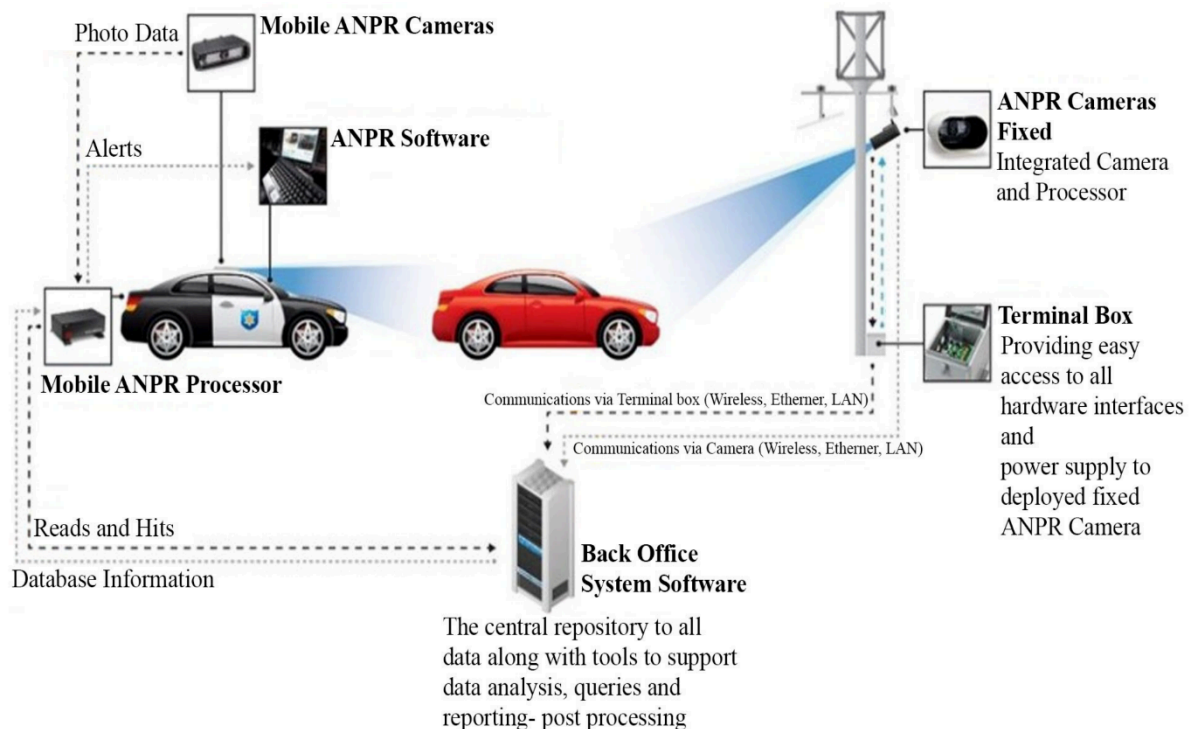
The problem I encountered in this project was saving the image to the permanent directory. At first, I had planned it to be an autonomous process i.e., when it detects the license plate it should save the image of the license plate to the directory but there was an issue. The issue was it saves blurry images of the license plate and this makes it difficult to check for saving in the permanent directory. So, I came up with the solution to save the image manually i.e., by pressing 's' to ensure the clarity of the image.



FUTURE OUTLOOK

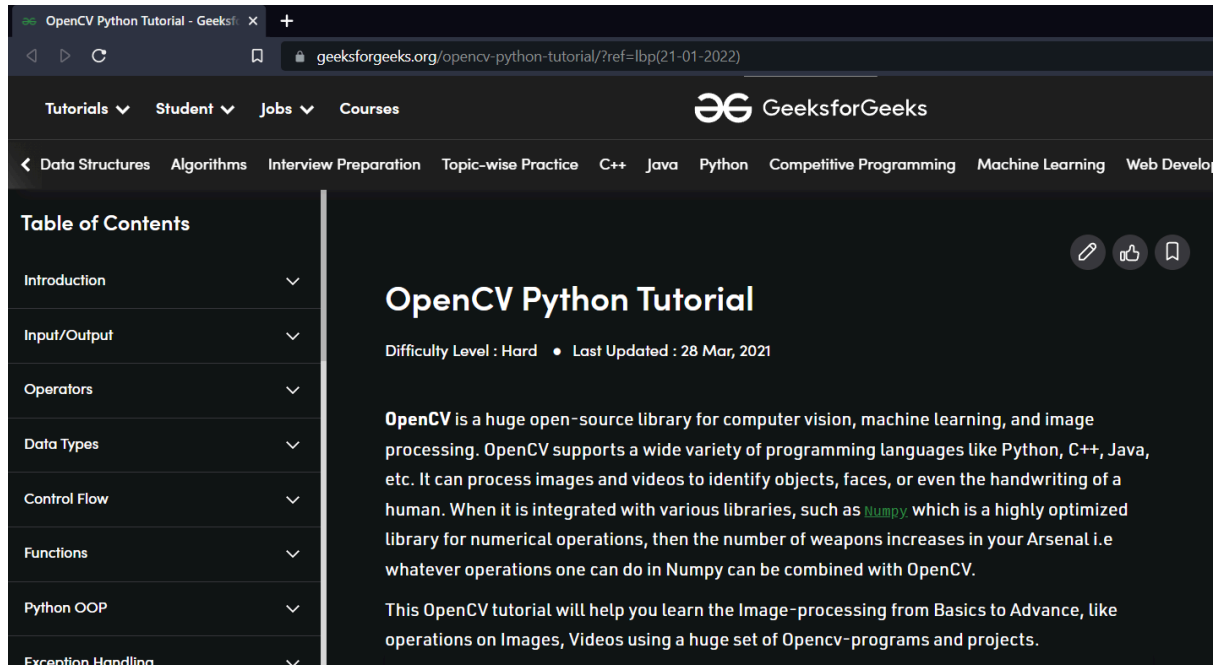
The ANPR system is the next generation of surveillance which will provide analytics, identification, efficiency, and convenience at ease.

Updates that can be made to the ANPR system are to improve the accuracy of detection of the license plate and OCR. Accuracy can be increased by leveraging TensorFlow Object Detection and Machine Learning. A web interface or an app can be created for the ANPR system to monitor the Livestream and to analyse the data that had been collected. Depending on the domain in which it is used, more specific features can be added to the ANPR system. For example, the ANPR system in Traffic management can be used to monitor vehicle acceleration and accidents, etc.



REFERENCES

[https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=lbp\(22-01-2022\)](https://www.geeksforgeeks.org/opencv-python-tutorial/?ref=lbp(22-01-2022))



[https://www.programiz.com/python-programming\(22-01-2022\)](https://www.programiz.com/python-programming(22-01-2022))

[https://www.w3schools.com/python/\(22-01-2022\)](https://www.w3schools.com/python/(22-01-2022))

