PROJECT
(2019-20)

Automatic India Car License Reader

Institute of Engineering & Technology

Team Members
Kashish Upadhyay(171500158)
Hritik Singh(171500141)
Nomaan Husain(171500191)

Supervised By
Name **Mr. Vaibhav Diwan**
Department of Computer Engineering & Applications

Department of computer Engineering and Applications
GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuhan,

Declaration

I hereby declare that the work which is being presented in the Mini Project **"Automatic India Car License Reader(ALPR)",** in partial fulfillment of the requirements for Mini project viva voce, is an authentic record of my own work carried under the supervision of ***Mr. Vaibhav Diwan*** **,** Technical trainer, Department of Computer Engineering & Applications**.**

Kashish Upadhyay(171500158)
Hritik Singh(171500141)
Nomaan Husain(171500191)

Course: B.Tech(CSE)
Year: 3rd
Semester: VI

PREFACE

This report documents is the work done for mini project at GLA University , on the Automatic India Car License Reader(ALPR), under the supervision of **Mr. Vaibhav Diwan**. The report will give an overview of the tasks completed during the period of two months with technical details. Then the results obtained are discussed and analyzed. we have tried our best to keep report simple yet technically correct. we hope we succeed in our attempt.

Kashish Upadhyay
Hritik Singh
Nomaan Husain

ACKNOWLEDGEMENT

It is our pleasure to acknowledge the assistance of a number of people without whose help this project would not have been possible.
First and foremost, we would like to express our gratitude to **Mr. Vaibhav Diwan** , our project guide, for providing invaluable Encouragement, guidance and assistance.we would like to thank the institute staff for the operation extended to us throughout the project. After doing this project we can confidently say that this experience has not only enriched us with technical knowledge but also has unparsed the maturity of thought and vision. The attributes required being a successful professional.

Kashish Upadhyay
Hritik Singh
Nomaan Husain

ABSTRACT

Traffic control and vehicle owner identification has become major problem in every country. Sometimes it becomes difficult to identify vehicle owner who violates traffic rules and drives too fast. Therefore, it is not possible to catch and punish those kinds of people because the traffic personal might not be able to retrieve vehicle License from the moving vehicle because of the speed of the vehicle. Therefore, there is a need to develop Automatic License Plate Recognition (ALPR) system as a one of the solutions to this problem. There are numerous ALPR systems available today. These systems are based on different methodologies but still it is really challenging task as some of the factors like high speed of vehicle, non-uniform vehicle License plate, language of vehicle License Plate and different lighting conditions can affect a lot in the overall recognition rate. Most of the systems work under these limitations. In this project different approaches of ALPR are discussed by considering image size, success rate and processing time as parameters.

Introduction:

The project Automatic India Car License Reader(ALPR) is a Machine learning based python application which is used to identify the License plate of vehicle and give dedicated plate image and plate number as an output.The main objective of this project is to ease the manual effort by use of computer technology and machine learning techniques. ALPR application works on the opencv which is the open computer vision library of machine learning technology.The automation of identification/tracking of vehicles using images can make it simple for officials to track crimes/stolen vehicles across the globe. This application just needs a .jpg image of a vehicle with a license plate clearly visible in the image then it uses various modules like PIL, OpenCV and pytesseract to identify the license plate.

Features:

The functions used in the application gives different outcomes which are then combined to give a integrated result these outcomes can be categorized as features of the ALPR application which are as follows:-

- **License Plate Detection:** The image to be used is detected by the program by reading the image in the code using cv2 package and suitable methods.
- **Preprocessing the image:** The image which is now available in the code is preprocessed, which means we ready it for further processing it consist manipulations like using gaussianblur and resizing the image which is then returned to be worked upon.
- **Extracting Contours:** This technique is used to extract the boundary of the digital image. This is also know as boundary/border following.
- **Detecting the text:** Character Segmentation and Character Recognition are two main aspects of this section after which the desired result is generated using the machine learning modules and algorithms.
- **Cleaning License plate:** The license plate image file is then cropped and final colour and texture touch is given to the image which is the cleaning of image.

TECHNOLOGY USED

Python

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code. It consist of various machine learning libraries, modules and packages which are very helpful in creating a machine learning or deep learning project.

Machine Learning

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence and stated that "it gives computers the ability to learn without being explicitly programmed".

Classification of Machine Learning

Machine learning implementations are classified into three major categories, depending on the nature of the learning "signal" or "response" available to a learning system which are as follows:-

- **Supervised learning :** When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning.
- **Unsupervised learning :** when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

- **Reinforcement learning :** When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which is connected to applications for which the

algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error. Errors help you learn because they have a penalty added (cost, loss of time, regret, pain, and so on), teaching you that a certain course of action is less likely to succeed than others. An interesting example of reinforcement learning occurs when computers learn to play video games by themselves.

- **Semi-supervised learning :** where an incomplete training signal is given: a training set with some of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

- **Classification :** When inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email messages and the classes are “spam” and “not spam”.

- **Regression :** Which is also a supervised problem, A case when the outputs are continuous rather than discrete.

- **Clustering :** When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

Python Machine Learning Libraries

- NumPy.
- Scikit-learn
- Theano
- TensorFlow
- Keras
- PyTorch
- Matplotlib

Image-Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.
If we talk about the basic definition of image processing then "Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality".

Digital-Image
An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.
In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.
Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

Image processing basically includes the following three steps

- Importing the image
- Analysing and manipulating the image
- Output in which result can be altered image or report that is based on image analysis.

Reading images in Python

Python supports very powerful tools when comes to image processing. Let's see how to process the images using different libraries
like OpenCV, Matplotlib, PIL etc.

- **Using OpenCV :** OpenCV (Open Source Computer Vision) is a computer vision library that contains various functions to perform operations on pictures or videos. It was originally developed by Intel but was later maintained by Willow Garage and is now maintained by Itseez. This library is cross-platform that is it is available on multiple programming languages such as Python, C++ etc.

- **Using MatplotLib :** Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002. Matplotlib comes with a wide variety of plots. Plots helps to

understand trends, patterns, and to make correlations. They're typically instruments for reasoning about quantitative information.

- **Using PIL :** PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. It was developed by Fredrik Lundh and several other contributors. Pillow is the friendly PIL fork and an easy to use library developed by Alex Clark and other contributors.

NumPy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.
It is the fundamental package for scientific computing with Python.

OpenCV

Computer Vision

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

OpenCV

OpenCV is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To Identify image pattern and its various features we use vector space and perform mathematical operations on these features.

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. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Applications of OpenCV

There are lots of applications which are solved using OpenCV, some of them are listed below

- face recognition
- Automated inspection and surveillance
- number of people – count (foot traffic in a mall, etc)
- Vehicle counting on highways along with their speeds
- Interactive art installations
- Anamoly (defect) detection in the manufacturing process (the odd defective products)
- Street view image stitching
- Video/image search and retrieval
- Robot and driver-less car navigation and control
- object recognition
- Medical image analysis
- Movies – 3D structure from motion
- TV Channels advertisement recognition

OpenCV Functionality

- Image/video I/O, processing, display (core, imgproc, highgui)
- Object/feature detection (objdetect, features2d, nonfree)
- Geometry-based monocular or stereo computer vision (calib3d, stitching, videostab)
- Computational photography (photo, video, superres)
- Machine learning & clustering (ml, flann)
- CUDA acceleration (gpu)

Copy in Python (Deep Copy and Shallow Copy)

In Python, Assignment statements do not copy objects, they create bindings between a target and an object. When we use = operator user thinks that this creates a new object; well, it doesn't. It only creates a new variable that shares the reference of the original object. Sometimes a user wants to work with mutable objects, in order to do that user looks for a way to create "real copies" or "clones" of these objects. Or, sometimes a user wants copies that user can modify without automatically modifying the original at the same time, in order to do that we create copies of objects.
A copy is sometimes needed so one can change one copy without changing the other. In Python, there are two ways to create copies :

- Deep copy
- Shallow copy

Deep copy is a process in which the copying process occurs recursively. It means first constructing a new collection object and then recursively populating it with copies of the child objects found in the original. In case of deep copy, a copy of object is copied in other object. It means that any changes made to a copy of object do not reflect in the original object. In python, this is implemented using “deepcopy()” function.

A shallow copy means constructing a new collection object and then populating it with references to the child objects found in the original. The copying process does not recurse and therefore won’t create copies of the child objects themselves. In case of shallow copy, a reference of object is copied in other object. It means that any changes made to a copy of object do reflect in the original object. In python, this is implemented using “copy()” function.

Python PIL | getbands() and getextrema() method

Python PIL library contains Image module in which variety of functions are defined. PIL.Image.Image.getbands()
This method is used to get the mode (bands) present in an image.
PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. It was developed by Fredrik Lundh and several other contributors.
Our goal is to convert a given text image into a string of text, saving it to a file and to hear what is written in the image through audio.

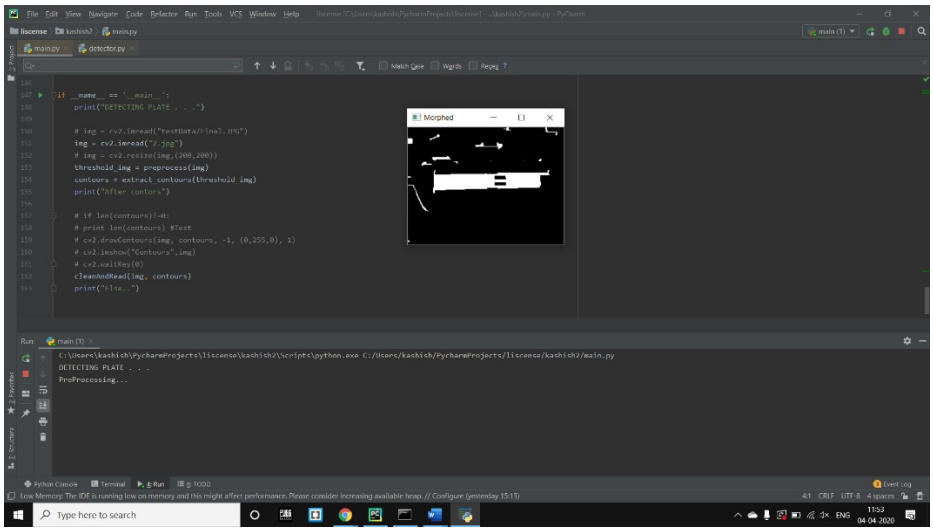
For this, we need to import some Libraries

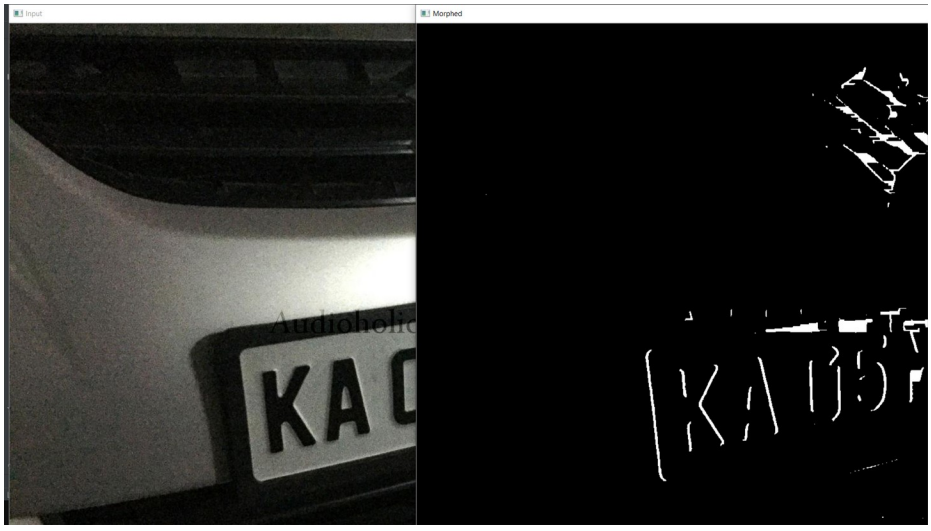
- **Pytesseract(Python-tesseract) :** It is an optical character recognition (OCR) tool for python sponsored by google.
- **Python Imaging Library (PIL) :** It adds image processing capabilities to your Python interpreter

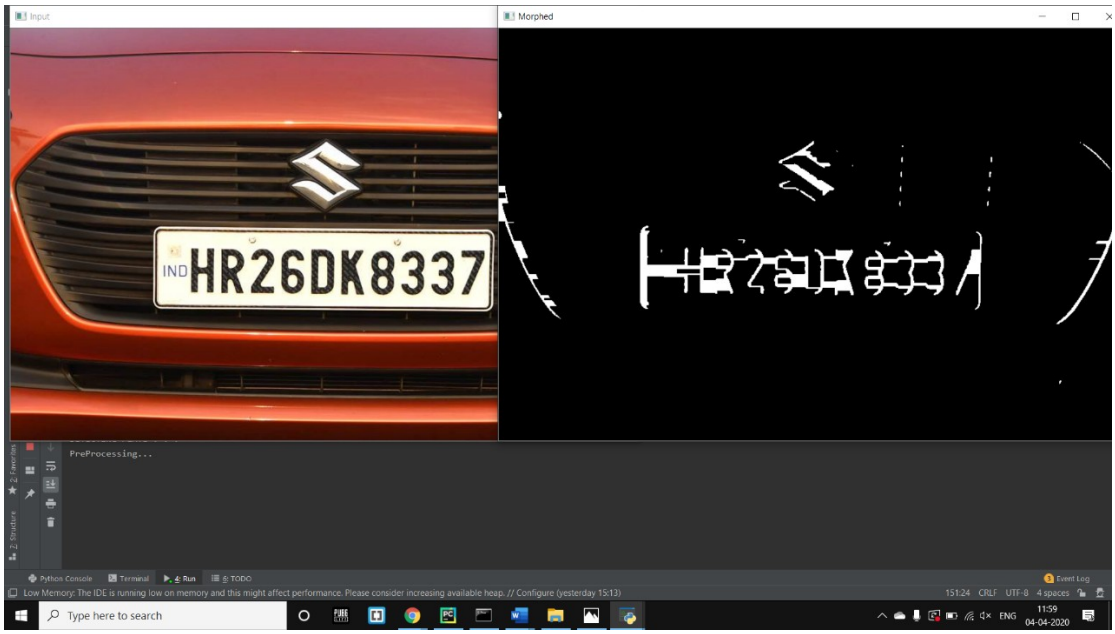
Implementation:-

Automatic License Plate Recognition system is an application of computer vision and image processing technology that takes photograph of vehicles as input image and by extracting their License plate from whole vehicle image , it display the License plate information into text. Mainly the ALPR system consists of 4 phases: - Acquisition of Vehicle Image and Pre-Processing, Extraction of License Plate Area, Character Segmentation and Character Recognition. The overall accuracy and efficiency of whole ALPR system depends on License plate extraction phase as character segmentation and character recognition phases also depend on the output of this phase. Further the accuracy of License Plate Extraction phase depends on the quality of captured vehicle image. Higher be the quality of captured input vehicle image more will be the chances of proper extraction of vehicle License plate area. The existing methods of ALPR works well for dark and bright/light categories image but it does not work well for Low Contrast, Blurred and Noisy images and the detection of exact License plate area by using the existing ALPR approach is not successful even after applying existing filtering and enhancement technique for these types of images. Due to wrong extraction of License plate area, the character segmentation and character recognition are also not successful in this case by using the existing method. To overcome these drawbacks we proposed an efficient approach for ALPR in which the input vehicle image is pre-processed firstly by iterative bilateral filtering , adaptive histogram equalization and number plate is extracted from pre-processed vehicle image using morphological operations, image subtraction, image binarization/thresholding, sobel vertical edge detection and by boundary box analysis. Sometimes the extracted plate area also contains noise, bolts, frames etc. So the extracted plate area is enhanced by using morphological operations to improve the quality of extracted plate so that the segmentation phase gives more successful output. The character segmentation is done by connected component analysis and boundary box analysis and finally in the last character recognition phase, the characters are recognized by the pytesseract executable file just is included in the code and is used to determine the text from a image to a string pattern.

License Plate Detection and Morphing

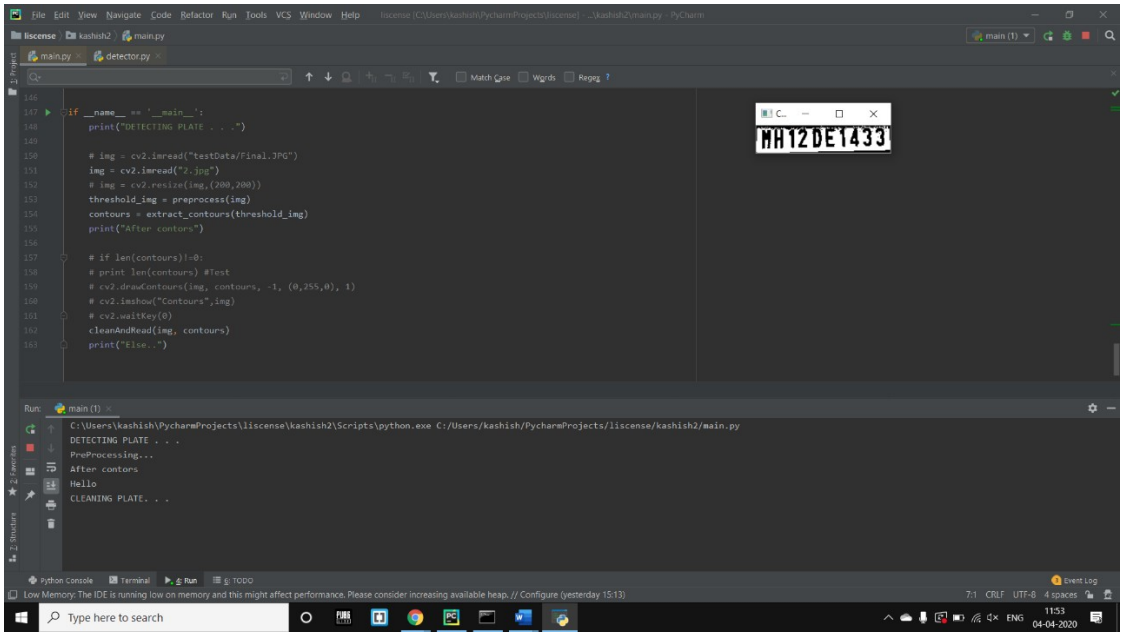






Morphing and license detection which is the preprocessing in this program is done using cv2 and other modules. Gaussianblur, changing the color and using sobel and threshold methods give the following morphed results.

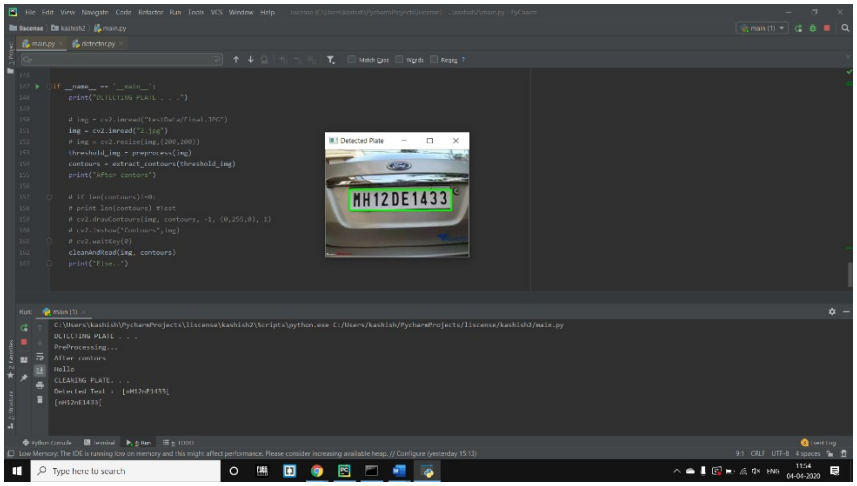
Extracting contour and cleaning license plate

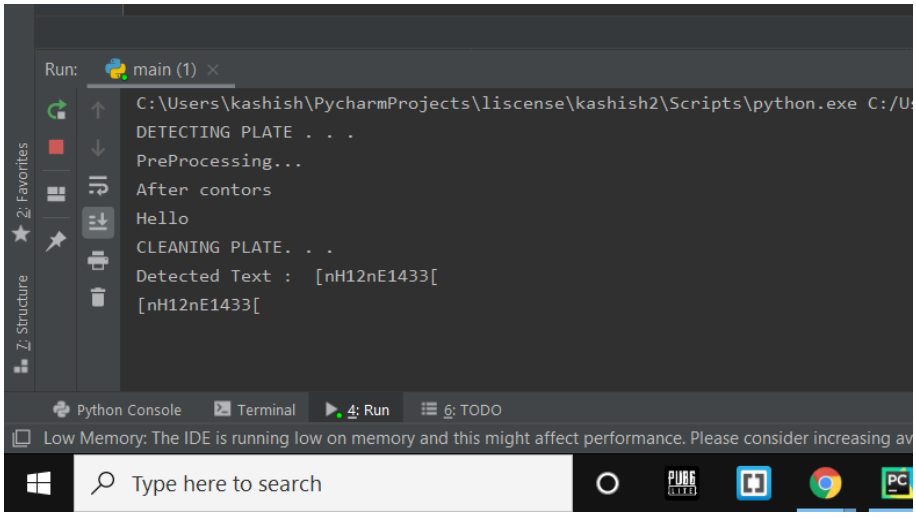


After the preprocessing the morphed image is used for extracting contour which is also known as boundary/border following. After cleaning the plate

image is displayed as cleaned license plate. Which is then used by a suitable algoritm to give the results.

Text detection and final results





The result of the license plate is displayed by an image with an border and the text is detected from the image using the pytesseract executable file which is included in the program and the results are displayed.

Motivation:

The main reason to make this system is to ease the effort required to gain the information by using computational power and machine learning technology. In order to get information like the license plate of a vehicle one should not do it physically rather it is easy to implement and use a computer to do the task. Hence we consider this project as a result of the following motivation

Future Prospects:

The following application can be further combined with databases which can store the record and information of people having a vehicle and hence by using the ALPR application it would become very easy to track/identify a particular vehicle along with the information of the owner which can help a lot in many scenarios hence there are various extension possible for the following Application. Which will further reduce human effort to do all the vehicle related task efficiently.

Requirements:

Hardware Components:

a) Processor – p4
b) Hard Disk – 5 GB
c) Memory – 1GB RAM

Software Requirements:

* python
* packages and extension for ML(opencv,PIL,numpy)
* pytesseract module