**CHAPTER 1**

**INTRODUCTION**

**1.1 IMAGE PROCESSING**

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristic features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too. Image processing is a rapidly growing area of computer science. Its growth has been fueled by technological advances in digital imaging, computer processors and mass storage devices. Fields which traditionally used analog imaging are now switching to digital systems, for their flexibility and affordability. Important examples are medicine, film and video production, photography, remote sensing, and security monitoring. These and other sources produce huge volumes of digital image data every day, more than could ever be examined manually.

In the last few decades, the development of image and video processing algorithms and approaches has received significant attention from the scientific community. This development is a consequence of the increased hardware and sensing capabilities and the increased quality of the digitally recorded materials. The increase in quality of the recorded materials and the increased computing capabilities have allowed the development of more complex and more powerful approaches. The large interest in the development of such algorithms is mainly caused by the large number of potential applications that could use them. The number of industries that could use the advancements of image processing and machine learning for the purpose of object recognition or segmentation from images is quite large. Some of the prospect applications are medical applications,

1

agriculture, ecology, quality inspection in automated production for machine parts, the military industry and many others. The ultimate goal is to build applications that could use machine learning and vision for process automation that could increase the productivity and ease the burden of the everyday life.

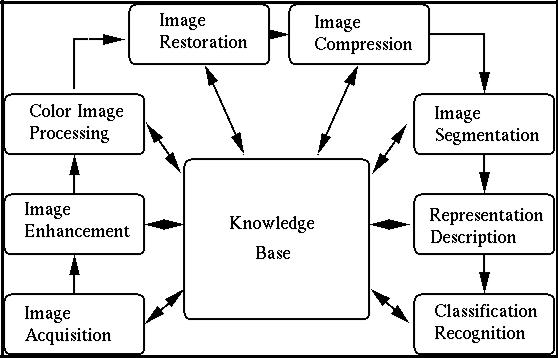
There are multiple tasks at hand that could use image processing and machine vision. Some of them are: Object recognition in images, image segmentation to extract regions of interest, image database search etc. All of these tasks are equally important for the development of the computer sciences and especially for the development of robotics and intelligent systems where these approaches could be beneficial

Digital image processing is concerned primarily with extracting useful information from images. Ideally, this is done by computers, with little or no human intervention. Image processing algorithms may be placed at three levels. At the lowest level are those techniques which deal directly with the raw, possibly noisy pixel values, with de-noising and edge detection being good examples. In the middle are algorithms which utilize low level results for further means, such as segmentation and edge linking. At the highest level are those methods which attempt to extract semantic meaning from the information provided by the lower levels, for example, handwriting recognition.

Image processing basically includes the following three steps - Importing the image via image acquisition tools, analyzing and manipulating the image and output which can be an altered image or a report that is based on image analysis. There are two types of methods used for image processing namely, analog and digital image processing. Analog image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers.

2

The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction. The preprocessing techniques are designed to enhance selected features and eliminate irrelevant data. Some image preprocessing methods are Smooth, Background Subtraction (Flat field), Dilate, Erode etc. Image enhancement is the process of improving the quality of a digitally stored image by manipulating the images with software. Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. Image classification refers to the process of categorizing all pixels in a digital image into one of several land cover classes.

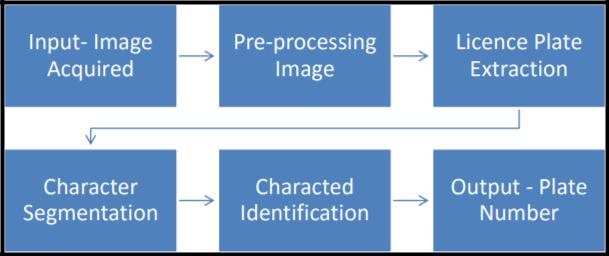


**Figure 1.1 Processes involved in Image Processing.**

Some of the major fields in which digital image processing is widely used are mentioned below:

3

* Medical field.
* Remote sensing.
* Machine/Robot vision.
* Pattern recognition.
* Video processing.



**Figure 1.2 Flowchart of image processing**

**technique.**

Modern digital technology has made it possible to manipulate multi-dimension signals with systems that range from simple digital circuits to advanced parallel computers. The goal of this manipulation can be divided into three categories:

* Image Processing image in → image out
* Image Analysis image in → measurements out
* Image Understanding image in → high-level description out.

Image processing has many operations which can be done on an image. These operations are done to uncover details and information from the image. Among many other image processing operations are:

* Euclidean geometry transformations such as enlargement, reduction, and rotation.

4

* Color corrections such as brightness and contrast adjustments, color mapping, color balancing, quantization, or color translation to a different color space.
* Digital compositing or optical compositing (combination of two or more images), Used in film-making to make a "matte".
* Interpolation, and recovery of a full image from a raw image format using a Bayer filter pattern.
* Image registration, the alignment of two or more images • Image differencing and morphing.
* Image recognition, for example, extract the text from the image using optical character recognition or checkbox and bubble values using optical mark recognition.
* Image segmentation.
* High dynamic range imaging by combining multiple images.
* Geometric hashing for 2-D object recognition with affine invariance.

**1.2 SYSTEM OVERVIEW**

There has been a rampant increase in the number of vehicles and as increasing vehicles are violating the traffic rules, high number of accidents has been recorded in today’s world . This often leads to damage of number plate of vehicles which makes it difficult for the police to identify the owner of the vehicle. For a vehicle to be recognized, vehicle license plate detection plays a crucial role. This involves five major steps: Preprocessing of captured image, Enhancing the image by removing noise and brightening the image to identify the key features, Segmentation and Character Recognition of license plate and finally verifying the obtained number in the database. In pre-processing, the required

5

image of the damaged number plate is taken through the digital camera, brightness of image is adjusted and noise removal using filters and image is converted to gray scale. Extractions of license plate region consist of finding the edges in the image where exact location of licenses plate is located and crop it into rectangular frame. Segmentation plays a vital role in vehicle licenses plate recognition; the damaged characters of the number plate are segmented and compared with the trained data set of characters of different fonts by template matching. Finally, recognition of each character is done and verified with the existing database to identify the owner of the vehicle.

**1.3 SCOPE OF THE PROJECT**

The disfigured number plate identification system identifies the damaged characters and important role in detecting damaged number plate and any security threat. The system uses a series of image processing techniques and identifies the vehicle from the database stored in the PC. The system is implemented using python and its performance is tested on real images. The system robustness and speed is improved. Distinctive filtering procedures can be acquainted with the reducing of noise to a more noteworthy degree, so the image processing can be more productive. In future, the recognition of damaged number plate should be possible from the video processing as well.

6

**CHAPTER 2**

**LITERATURE SURVEY**

1. **Vehicle License Plate Tracking and Owner Details Identification. Author: S.Jayapradha, Dr.S.P.Manikandan. Year:**

**April 2017**

**Transaction: International Journal of Informative & Futuristic Research.**

Jayapradha & Manikandan have designed the License plate Tracking System to provide high speed security as well as monitoring system. The system also recognizes and identifies the car plate automatically. Car license plate identification plays a very important role in the Data Driven Intelligent Transportation System. Each car has a unique license plate to identify them. Jayapradha & Manikandan designed the system where the license number of the car has been checked against the database and only the authorized vehicles would be allowed to enter the restricted areas. Character extraction and database matching played a vital role in their Identification process. The performance of the system depends on the Feature extraction. Their system has been developed in the manner to extract relevant features for improvising the performance and accuracy of the whole system. The successful recognition of a moving vehicle has been analysed in their system.

1. **Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications**

**Author: Rahim Panahi, Member, IEEE, and Iman Gholampour, Member, IEEE**

**Year: 2016**

**Transaction: IEEE transactions on intelligent transportation systems**

Rahim and Iman drafted an online highly accurate system for ANPR that could be used as a basis for many real-world ITS applications. Their system have been designed to deal with unclear vehicle plates, variations in weather and lighting

7

conditions, different traffic situations, and high-speed vehicles. They have addressed various issues by presenting proper hardware platforms along with real-time, robust, and innovative algorithms. The data sets have images that have been captured from cross- roads, streets, and highways, in day and night, various weather conditions, and different plate clarities. Over these data sets, their system achieved a very good successful accuracy rates for plate detection, character segmentation, and plate recognition, respectively. The false alarm rate in plate detection have been analyzed to be slightly less. The overall accuracy on the dirty plate’s portion of their data sets have been victorious. The algorithms that have been used by them for each part of the system that has been robust to lighting changes. Their system has been also tested on three other Iranian data sets and has achieved perfect accuracy in both detection and recognition parts.

1. **Indian Vehicle Number Plate Detection Using Image Processing Author: Prof.Amit Kukreja, Swati Bhandari, Sayali Bhatkar, Jyoti Chavda, Smita Lad Year: April 2017**

**Transaction: International Research Journal of Engineering and Technology**

Vehicle number plate recognition by Amit et al designed the most exciting and challenging research topic from past few years. An efficient less time-consuming vehicle number plate detection method have been projected which have been performed on multifaceted image. By using, Sobel edge detection method they detected the edges and filled the holes less than 8 pixels only. They anticipated algorithm that have been mainly based on Indian automobile number plate system. Extraction of number plate accuracy has been increased for low ambient light image. Number plates of Different shapes, size and color of number plates have been used in different countries. The work has been distributed into several parts: Input raw image, Image binarization, reduce noise using mid-filtering method, enhance contrast using histogram equalizer, Plate localization and Character segmentation. If the number

8

plate being similar to the background then their system has a challenge to identify the location. Illumination and contrast have been changed when the light fall changes according to the morphological operations that have been used to eliminate the contrast feature within the plate.

1. **An Efficient Method for Vehicle License Plate Detection in Complex Scenes. Author: Mahmood Ashoori-Lalimi, Sedigheh Ghofrani Year: 2011 Transaction: IEEE Journal**

Mahmood and Sedigheh designed an efficient method for license plate localization in the images with various situations and complex background. The main goal of their system has to reduce problems such as low quality and low contrast in the vehicle images, image contrast have been enhanced by the two different methods and the best for following has been selected. Later, vertical edges of the enhanced image have been extracted by sobel mask. Then the most of the noise and background edges have been removed by an effective algorithm. The output of this stage have been given to a morphological filtering to extract the candidate regions and finally they used several geometrical features such as area of the regions, aspect ratio and edge density to eliminate the non-plate regions and segment the plate from the input car image. Their System has been performed on some real images that have been captured at the different imaging conditions. Their experimental results have shown that their depicted model that has nearly independent to environmental conditions such as lightening, camera angles and camera distance from the automobile, and license plate rotation.

9

1. **Number Plate Recognition and Document Verification using Feature Extraction OCR Algorithm**

**Author: Bhonsale Tejas, Dhamal Omkar, Dhumal Rutuj, Khedekar Prajakta, Patil Bhakti**

**Year:Nov 2016**

**Transaction: International Journal of Engineering and Computer science.**

The police forces around the world use vehicle number plate for legal vehicle authorization purposes, to check if a vehicle have been registered or licensed. Most of us keep the vehicle papers in the vehicle itself, which have not at all safe in case of theft. In today's world, it has no security to carry the vehicle papers along with the vehicle. Hence Bhonsale et al sketched a system in which it has been not necessary to carry the important documents to each and every place for verification. So, they designed a system which captured the image of the number plate of a vehicle using a camera and the details have been retrieved using the character segmentation which has done by a feature extraction optical character recognition algorithm (OCR). Then the details have been retrieved from the number plate in text format that have been used to extract all the important information of the vehicle like, the name of the owner, address of the owner, date of registration of the vehicle etc. from the database. The police can even verify whether the documents that have been fake or not. According to their system, it have been useful as they do not have to carry their documents to every place with the fear of losing them. Their System have been efficient for being used in practice.

1. **An Efficient Approach for Automatic Number Plate Recognition System under Image Processing.**

**Author: Sarbjit Kaur, Sukhvir Kaur Year: August 2014**

**Transaction: International Journal of Advanced Research in Computer science.**

Sarbjit Kaur and Sukhvir Kaur designed the ANPR system , an image processing

10

technology that identifies the vehicles by tracking their number plate without direct human intervention and an application of computer vision. ANPR have been an important method that has been used in ITS (Intelligent Transportation System). Number Plate Extraction has been the most crucial step in the ANPR system which affected the overall accuracy and efficiency of whole ANPR system. The quality of acquired vehicle image has been a major factor in the success of ANPR. Sarbjit and Sukhvir depicted an efficient approach for ANPR in which the input vehicle image have been pre-processed first using iterative bilateral filter and adaptive histogram equalization and number plate which have been extracted from pre-processed vehicle image using morphological operations, image subtraction, thresholding, sobel edge detection and boundary box analysis. An efficient approach for number plate extraction has been described in their system. Their system have been mainly designed for real-time Indian vehicles number plate but it also works well for foreign number plates. This extraction process works well for low resolution, noisy and low contrast images. This model depicted by them have tested number of vehicles images under different weather and illumination conditions i.e. daytime, night time, sunny, cloudy, rainy days etc and success rate achieved have been Successful.

1. **Vehicle Number Plates Detection and Recognition using improved Algorithms: A Review with Tanzanian Case study**

**Author: Cosmo H.Munuo, Dr. Michael Kisangiri Year: May 2014**

**Transaction: International Journal of Engineering and Computer Science.**

Number Plates Recognition (NPR), which have been invented in 1976 have been used as a wide commercial application, making its research prospects challenging and scientifically interesting. A complete NPR system functions regularly as localization, sizing and orientation, normalization, character recognitions and geometric analysis. Cosmo and Michael studied various system and gave a review of NPR preliminary stages which explains number plate localization, sizing and orientations as well as normalizations sections of the Number Plates Detection and Recognition. The input

11

incorporated includes front and rear photographic images of vehicles, for proximity and simulation purposes the ample angle of image has 90 degree +-15. The captured image has been converted to gray scale, binarized and edge detection algorithms have been used to enhance edges. The output of this stage provides the input feature extraction, segmentation and recognition. However, it should be noted that a complete NPR system have required effective set of hardware and software components, most preferably efficient infra-red cameras and powerful computers to provide high quality images, which acts as the disadvantage of their system. Canny edge detector has been used by them to have higher accuracy.

1. **A Survey on License Plate Recognition Systems Author: Divya Gilly, Kumudha Raimond, PhD Year: January 2013**

**Transaction: International Research Journal of Engineering and Technology**

License Plate Recognition (LPR), a well-known image processing technology. which captures the image from digital camera that follows pre-processing, character segmentation and character recognition. License plates have been available in different formats in various countries. So, each country develops the LPR system appropriate for the vehicle license plate format. LPR system faces the difficulties of the environmental and non-uniform outdoor illumination conditions. Therefore, most of the systems work under restricted environmental conditions like fixed illumination, limited vehicle speed, designated routes, and stationary backgrounds. Each LPR system use different combination of algorithms. From the papers being surveyed, Divya & kumudha have realized that a good success rate have been obtained by the combination of fuzzy logic for license plate detection and Self Organizing (SO) neural network for character recognition.

12

1. **An Approach towards Detection of Indian Number Plate from Vehicle Author: Sourav Roy, Amitava Choudhury, Joydeep Mukherjee Year:March 2013**

**Transaction: International Journal of Innovative Technology and Exploring Engineering**

Vehicle number plate recognition, the most interesting and challenging research topic for past few years. The numbers plates of different shape and size and also have different color have been available in different countries. In India the most common vehicle number plate used yellow or while as background and black used as foreground color. Sourav et al designed a system for the localization of number plate mainly from the vehicles of West Bengal (India) and segmented the numbers as to identify each number separately. They depicted an approach based on simple and efficient morphological operation and sobel edge detection method. They also drafted a simple approach to segment all the letters and numbers used in the number plate. After reducing noise from the input image, they tried to enhance the contrast of the binarized image using histogram equalization. They mainly concentrated on two steps; to locate the number plate and to segment all the number and letters to identify each number separately.

1. **Text detection and character recognition through fuzzy logic. Author: Mohanad Alata, Mohammad Al-Shabi Year: 2005 Transaction: Journal of ELECTRICAL ENGINEERING**

The investigation by Mohanad and Mohammad presented an algorithm and software to detect and recognize character in an image. Three types of fonts under the investigation, namely, case (I): Verdana, case (II): Arial and case (III): Lucida Console. The font size will be within the range of 17–29 font size. These types have been chosen because the characters have low variance and there has less redundancy in the single character. They designed an algorithm which has assumed that there must be at least three characters of same color in a single line and no merge

13

between group of characters and at last, a single character has only one color. They developed three different algorithms such as detection algorithm, recognition algorithm and fuzzy system which have been implemented differently. The basic methodology used by them have to use the 8-connected component to binarize the image, then to find the characters in the image and recognize them. The negative texts have been considered as a graphic region which may aﬀect the system eﬃciency. The presented approach would take each color alone. This would make the merge between the text and the background unlikely to happen. Also, there would be no negative text in the whole image because the negative texts have been changed into a color as a normal text for another. The results have shown that the eﬃciency of the discussed algorithm has been successful.

14

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

The existing vehicle number plate detection makes use of a digital camera to capture the image of the number plate. The captured image is then preprocessed to correct blurry image, sharpen it and remove noise. It then uses a series of image manipulation techniques to detect, normalize and enhance the image of the number plate .Then optical character recognition (OCR) is done to extract the alpha-numeric of the license plate . This can be done real time or sent to a remote server to perform OCR. This system is used to detect when the vehicle is in most ordinary driving condition.

**3.1.1 DISADVANTAGES**

* **Object Obscure**: This happens when certain object or substance sticks on tothe plate or a part of the plate, for example, dirt or tow bar wiring which causes complications in taking a clear image of the Licence Plate..
* **Poor Lighting and Low Contrast Images**: This situation arises mostly onaccount of overexposure in the surroundings, reflection of various objects in the environment or shadows projecting on the licence plate.
* **Trouble caused on account of different fonts used**: A different font is usedon the Licence Plate of the Vanity Vans, which enhances the problem. However, various countries have completely boycotted this which has resulted in eliminating the trouble caused due to the employment of these dissimilar fonts.

15

**3.2 PROPOSED SYSTEM**

The high inclination in the number of accidents due to violation of traffic rules and regulation calls for speedy recognition of the details of the vehicle. The existing system fails to identify disrupted number plate as it is difficult to identify the disfigured characters. The feasible solution is to train the system with damaged character to help identify the character after feature extraction and segmentation.

The architecture is designed to first capture the image of the car. After preprocessing and enhancing the captured image it is then extracted to obtain the ends of the rectangular number plate. Each character is segmented and the character is identified. The K nearest neighbor algorithm is used and the system is trained with dataset to identify damaged characters. After the number is obtained it is verified with the existing database created using MS SQL server to procure the car details and show the result in a web page.

**3.2.1 ADVANTAGES**

* More efficient and scalability is improved.
* It can identify even damaged characters.

**3.3** **REQUIREMENTS SPECIFICATION**

**3.3.1 Hardware Requirements**

* Hard Disk: 40GB and above
* RAM: 512MB and above
* Processor: Pentium IV and above

**3.3.2 Software Requirements**

* Windows operating system XP and above.

16

**3.4** **LANGUAGE SPECIFICATION**

**3.4.1 PYTHON**

Python is an interpreted high level programming language for general purpose programming. Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Python is used in many applications:

* Web and Internet Development
* Scientific and Numeric
* Education
* Desktop GUIs
* Business Applications

Python is mostly preferred for image processing than MATLAB. The scientific Python ecosystem is maturing fast and Python is an appealing alternative, because it’s free, open source, and becoming ever more powerful. Python code tends to be more compact and readable than matlab. Matlab is expensive but python is free to use. The common differences between the two tools are ,

* **Easy programming language**. Python was created to be a generic language

17

that is easy to read, while Matlab started as a matrix manipulation package to which a programming language was added.

* **Powerful**. It’s easier than other languages to transform the ideas into code.

Further, Python comes with extensive standard libraries, and has a powerful datatypes such as lists, sets and dictionaries. These really help to organize the data.

* **String manipulation**. This is incredibly easy in Python. This line of code

which return straight justified line of thirty characters: "I code in Matlab".replace('Matlab','Python').rjust(30)

* **Portability**. Because Python is for free, the code can run everywhere. Further,it works on Windows, Linux, and OS X.
* **Class and function definitions**. Functions and classes can be definedanywhere. In one file as many functions and classes can be designed.
* **Great GUI toolkits**. With Python a front-end for the application that looksgood and works well can be created. Any of the major GUI toolkits like Wx or Qt. Pyzo comes with PySide (a wrapper for Qt) can be chosed.

**3.4.2 OPENCV**

OpenCV was started at Intel in 1999 by Gary Bradsky and the first release came out in 2000. Vadim Pisarevsky joined Gary Bradsky to manage Intel’s Russian software OpenCV team. In 2005, OpenCV was used on Stanley, the vehicle who won 2005 DARPA Grand Challenge. Later its active development continued under the support of Willow Garage, with Gary Bradsky and Vadim Pisarevsky leading the project. Right now, OpenCV supports a lot of algorithms related to Computer Vision and Machine Learning and it is expanding day-by-day.

Currently OpenCV supports a wide variety of programming languages like C++, Python, Java etc and is available on different platforms including Windows, Linux, OS X, Android, iOS etc. Also, interfaces based on CUDA and OpenCL are also under active development for high-speed GPU operations. OpenCV-Python is the Python API of OpenCV. It combines the best qualities of OpenCV C++ API and Python language. OpenCV is the leading open source library for computer

18

vision, image processing and machine learning, and now features GPU acceleration for real-time operation.

OpenCV is released under a BSD license and hence it’s free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 6 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics.

**APPLICATIONS:**

The applications of opencv includes ,

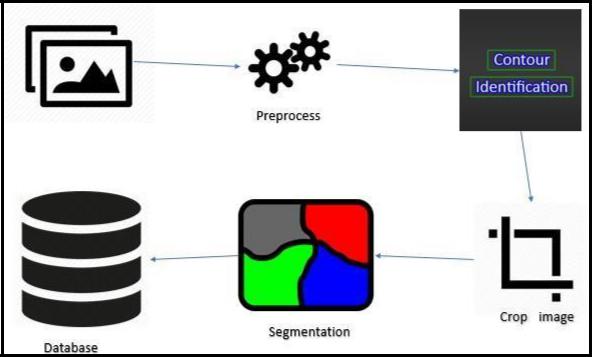
* Street view image stitching
* Automated inspection and surveillance
* Robot and driver-less car navigation and control
* Medical image analysis
* Video/image search and retrieval

19

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 SYSTEM ARCHITECTURE:**



**Figure 4.1 System Architecture of Proposed System**

In Figure 4.1, a vehicle’s Image is taken as an input. The vehicle’s image contains a damaged number plate in which the characters are not clear enough. The number plate is preprocessed using preprocessing algorithms which is then undergoes contour extraction. Later the number plate alone is cropped from the image which undergoes the process of dilation and erosion. Then the process of segmentation of Characters takes places which recognizes the number plate. The detected number plate’s number is then compared with the database that is provided and brings out the details of the owner of the corresponding vehicle. Thus the disfigured number plate’s number could be detected by this system.

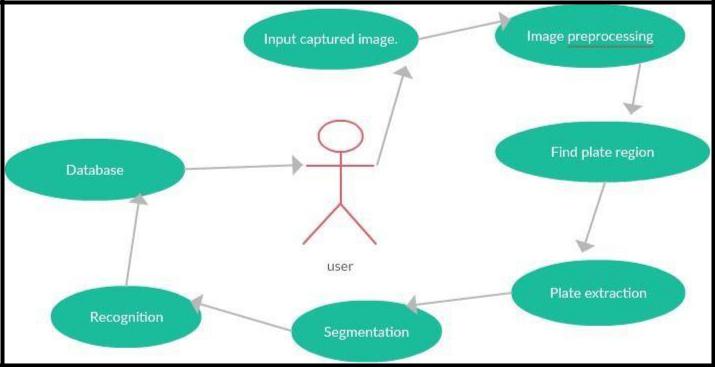
20

**4.2 USE CASE DIAGRAM**

In the Unified Modeling Language (UML), a use case diagram can summarize the details of the system's users (also known as actors) and their interactions with the system. To build one, a set of specialized symbols and connectors are used. An effective use case diagram can help the team discuss and represent:

* Scenarios in which the system or application interacts with people, organizations, or external systems.
* Goals that the system or application helps those entities (known as actors) achieve.
* The scope of the system.

Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.



**Figure 4.2 Use Case Diagram of Proposed System**

21

In Figure 4.2, the Use Case of the Proposed System is given as above. The Input Captured Image is preprocessed and plate is extracted and segmented, later the text is recognized and then identified by matching it with data’s that are present in the database and details will be revealed.

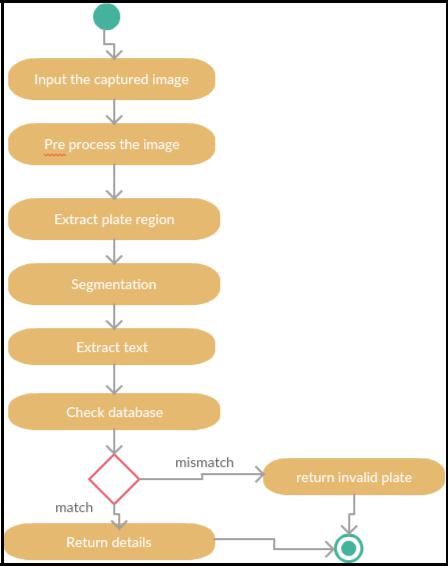
**4.3 ACTIVITY DIAGRAM**

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. An activity diagram is a behavioral diagram i.e. it depicts the behavior of a system.

The purpose of an activity diagram can be described as :

* Describe the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

22



**Figure 4.3 Activity Diagram of Proposed System**

In Figure 4.3, the Activity flow of the system is explained. The input image is given for preprocessing .Then the extraction of contour takes places. Later the cropped image is retrieved and the text of the number plate is detected which is compared with the database to bring out the details of the vehicle using the proposed system.

**4.4 SEQUENCE DIAGRAM**

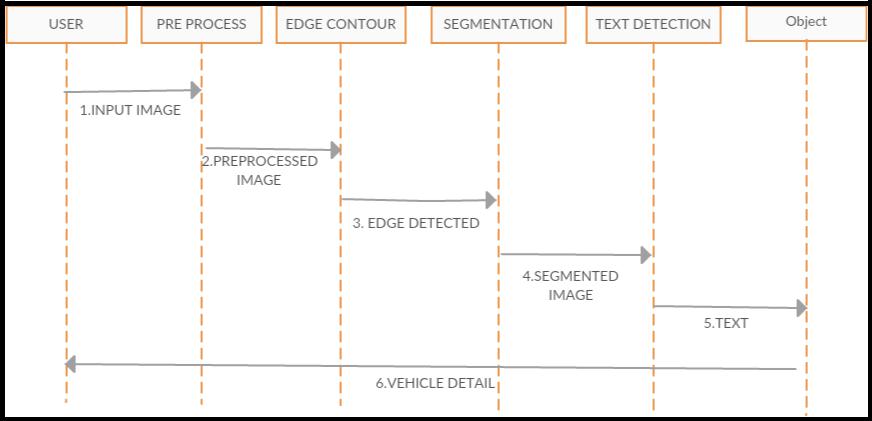
A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Terms like event diagrams or event scenarios can also be used to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These

23

diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

Uses of sequence diagrams:

* Used to model and visualize the logic behind a sophisticated function, operation or procedure.
* They are also used to show details of UML use case diagrams.
* Used to understand the detailed functionality of current or future systems.
* Visualize how messages and tasks move between objects or components in a system.



**Figure 4.4 Sequence Diagram of Proposed System**

In Figure 4.4, the system that is given to explain about the detection of disrupted image from the license plate of the vehicles. The user gives the input image and then it is preprocessed and edge detection takes places. The contours are extracted from the input image. Later the image is cropped. The text is detected from the cropped image which is then compared with the data’s present in the database and gives out the details of the vehicle in the perfect manner.

24

**CHAPTER 5**

**SYSTEM IMPLEMENTATION**

**5.1** **MODULES**

The modules are

1. Pre-processing
2. Extract Contours
3. Crop Image
4. Text Identification
5. Database Checking

**5.1.1 PRE-PROCESSING**

Image pre-processing refers to operations done before a key processing step, such as filtering, color transforms, sub-sampling/scaling, histogram generation, edge detection, segmentation etc. intelligent use of image pre-processing can provide benefits and solve problems that ultimately lead to better local and global feature detection. Image pre-processing has positive effects on the quality of feature extraction and the results of image analysis. Image pre-processing is analogous to the mathematical normalization of a data set, which is the basic step in feature description.

Image processing plays a major role in character recognition. The importance of the pre-processing stage of a character recognition system lies in its ability to remedy some of the problems that may occur due to some of the factors like scanner quality, scan resolution, type of printed documents etc. Intensity transformations are applied to enhance the quality of the image for further processing. The following transformations are applied on the color JPEG image:

25

Conversion of RGB image into a Gray-scale Intensity image , Conversion of Intensity image into a Binary image and Conversion of Gray-scale Image into an edge detected image.

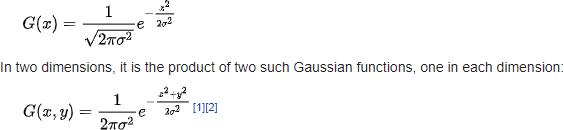
The pre-processing techniques used are

1. Gaussian blur.
2. Grey scale.
3. Edge detection.
4. Threshold.

**5.1.1.1 GAUSSIAN BLUR**

In image processing, a Gaussian blur is the result obtained by blurring an image using a Gaussian function. It has its wide application in image processing to reduce noise. Applying a Gaussian blur to an image is the same as convolving the image with a Gaussian function. It is a type of image-blurring filter that uses a Gaussian function (which also expresses the normal distribution in statistics) for calculating the transformation to apply to each pixel in the image.

The formula of a Gaussian function in one dimension is



Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. This operation can be performed on an image using the Gaussianblur() method of the imgproc class.

26

This method accepts the following parameters –

* src − A Mat object representing the source (input image) for this operation.
* dst − A Mat object representing the destination (output image) for this operation.
* ksize − A Size object representing the size of the kernel.
* sigmaX − A variable of the type double representing the Gaussian kernel standard deviation in X direction.

**5.1.1.2 GREY SCALE**

A greyscale or greyscale image is the one in which the value of each pixel is a single sample representing only the amount of light. It carries only the information about the intensity of the image. Grey scale images, a kind of black- and-white or gray monochrome, are composed exclusively of shades of grey. It can be the result of measuring the intensity of light at each pixel according to a particular weighted combination of frequencies (or wavelengths), and in such cases they are monochromatic proper when only a single frequency is captured. Grey scale images are preferred over coloured ones to simplify mathematics. It is relatively easier to deal with (in terms of mathematics) a single colour channel (shades of white/black) than multiple colour channels.

The image that is acquired from the camera can be an RGB colour image or a Grayscale Intensity image. The algorithm has to check for the RGB image and then has to convert it into a Grayscale image, because all the further processing is done in Grayscale format. Grayscale is chosen because of its simplicity in processing and for its two dimensional matrix nature, and also it contains enough information needed for the actual recognition.

27



**Figure 5.1 Gray Scale conversion**

**5.1.1 EDGE DETECTION**

Edge detection is one of the important pre-processing steps in many of the image processing applications. Edge detection is a pre-processing step in many applications such as object recognition, boundary extraction and segmentation. The basic assumption used in most edge detection is that the edges are characterized by large changes in intensity. Hence, at the location of the edge the first derivative of the intensity function should be a maximum or the second derivative should have a zero crossing.

The Gray scale image is converted into edge detected image. Edge supports different edge-finding methods. Each method has its own advantages and disadvantages and can be used in specific areas. The six different types of edge detection methods are ,

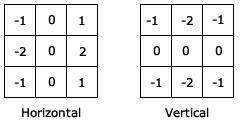
28

* The Sobel method finds edges of an image using the Sobel approximation to the derivative. It produces edges at those points where the gradient of an image is maximum.
* The Prewitt method finds edges using the Prewitt approximation to the derivative. It gives edges at those points where the gradient of an image is maximum.
* The Roberts method finds edges using the Roberts approximation to the derivative. It produces edges at those points where the gradient of an image is maximum.
* The Laplacian of Gaussian method finds edges by looking for zero crossings after filtering an image with a Laplacian of Gaussian filter.
* The zero-cross method produces edges by looking for zero crossings after filtering an image with the filter specified.
* The Canny method finds edges by looking for local maxima of the gradient of an image. The gradient is calculated using the derivative of a Gaussian filter. This technique employs two thresholds, to identify strong and weak edges, and comprises the weak edges in the output only if they are connected to strong edges. This method is therefore less prone than others to be affected by noise, and more likely to detect the weak edges.

Out of all these, Sobel edge detection gives the better results. So Sobel edge detection is taken.

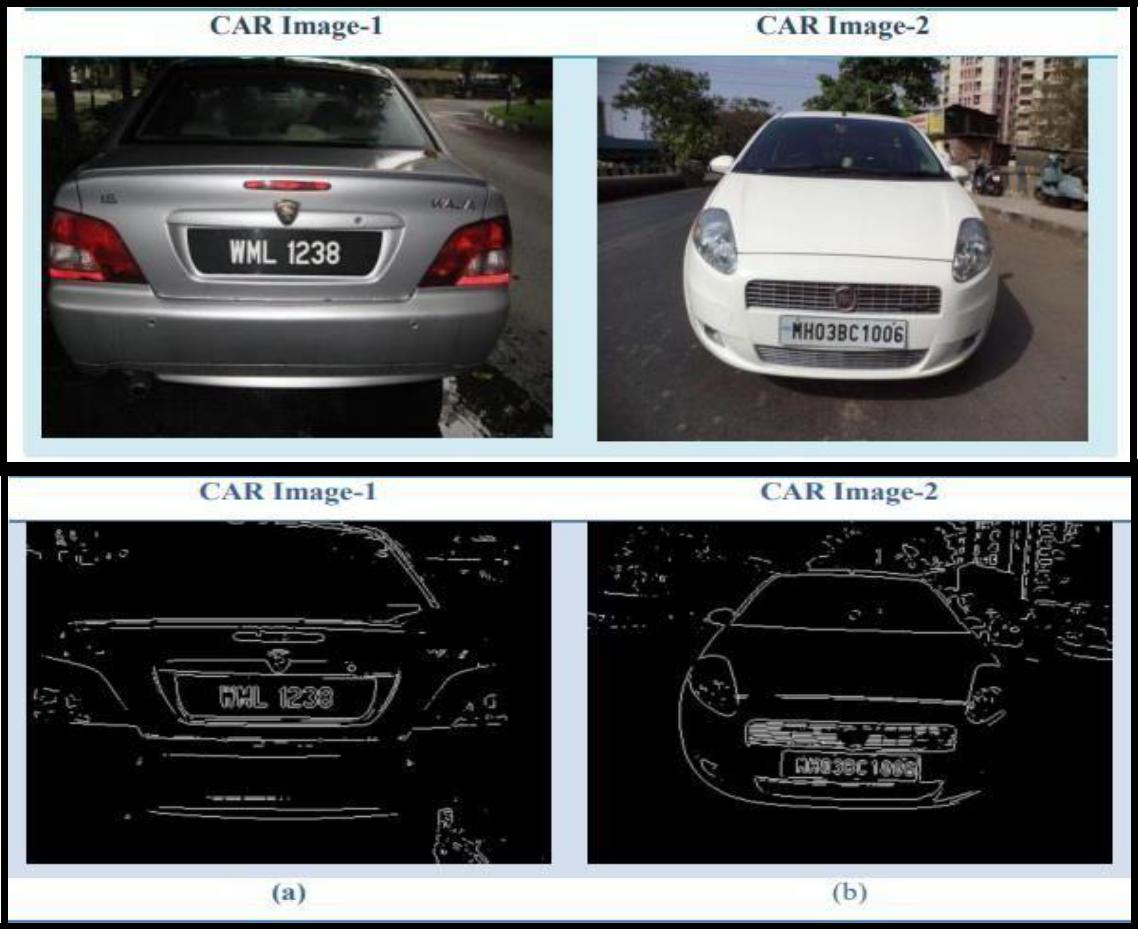
The operator used is sobel operator. It is a gradient based method which works with firs order derivatives. It calculates the first derivatives of the image separately for the X and Y axes. The derivatives are only approximations (because the images are not continuous). To approximate them, the following kernels are used for convolution:

29



Kernels used in the Sobel edge detection. The kernel on the left approximates the derivative along the X axis. The one on the right is for the Y axis. Using this information, the following can be calculated:

* Magnitude or "strength" of the edge.
* Approximate strength.
* The orientation of the edge.



**Figure 5.2 Edge Detection applied to an image.**

30

**5.1.1.4 THRESHOLD**

Thresholding can be defined as a process of converting a grayscale input image to a bi-level image by using an optimal threshold. Its purpose is to extract those pixels from some image which represent an object which can be either text or other line image data such as graphs, maps etc. Though the information is binary the pixels represent a range of intensities. Thus the objective of binarization is to mark pixels that belong to true foreground regions with a single intensity and background regions with different intensities. It is one of the simplest method to separate regions which are higher than the set threshold.

This transformation, also known as Image Quantization, produces a binary image from the intensity image by comparing pixel intensities with a threshold. This stage is very critical for the separation of license plate from the acquired car image. The license plate is assumed to have black characters on a white background and also it is assumed that all the license plates for various cars have approximately the same uniform colour. Taking the advantage of its brighter background area, the license plate can be separated from the relatively darker car image.

The threshold for binarization can be two types static or dynamic. In static general threshold is taken 150 (in gray scale of 0-255). This value reasonably quantizes those pixels, which represents the license plate and any other portions, which has a pixel-value more than the selected threshold. The remaining portions, which have a pixel-value less than the threshold value, are darkened, as shown in Figure 5.3. In dynamic threshold, the threshold taken will be the average of low and median gray scale values.

31



**Figure 5.3 Thresholded Image**

**5.1.2 EXTRACT CONTOUR**

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

* For better accuracy, binary images are used . So before finding contours, apply threshold or canny edge detection.
* Since OpenCV, findcontour() no longer modifies the source image but returns a modified image as the first of three return parameters.
* In OpenCV, finding contours is like finding white object from black background. So object to be found should be white and background should be black

Opening is just another name of erosion followed by dilation. Here we use the function, cv.morphologyEx()

Opening = cv.morphologyEx (img, cv.MORPH\_OPEN, kernel)

32



**Figure 5.4 Result of Opening which is Erosion followed by Dilation**

The Main concept over here is

* Erosion
* Dilation

**5.1.2.1 EROSION**

The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object .The kernel slides through the image . A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is eroded. All the pixels near boundary will be discarded depending upon the size of kernel. So the thickness or size of the foreground object decreases or simply white region decreases in the image. It is useful for removing small white noises, detach two connected objects etc.

The erosion of a binary image f by a structuring element s (denoted f s) creates a new binary image g = f s with ones in all locations (x, y) of a structuring element's origin at which that structuring element s fits the input image f, i.e. g(x, y) = 1 is s fits f and 0 otherwise, repeating for all pixel coordinates (x, y).



**Figure 5.5 Erosion applied to an image.**

33

**Basics of Erosion:**

* Erodes away the boundaries of foreground object
* Used to diminish the features of an image.

**Working of erosion:**

1. A kernel (a matrix of odd size (3, 5, 7) is convolved with the image.
2. A pixel in the original image (either 1 or 0) will be considered 1 only if all the pixels under the kernel is 1, otherwise it is made to zero.
3. All the pixels near boundary will be discarded depending upon the size of kernel.
4. So the thickness or size of the foreground object decreases or simply white region decreases in the image.

**5.1.2.2 DILATION**

It is the opposite of erosion. Here, a pixel element is '1' if at least one pixel under the kernel is '1'. So it increases the white region in the image or size of foreground object increases. Normally, in cases like noise removal, erosion is followed by dilation. Erosion removes white noises, but it also shrinks the object. So it is dilated. Since noise is gone, they won't come back, but the object area increases. It is also useful in joining broken parts of an object.

The dilation of an image f by a structuring element s (denoted f s) produces a new binary image g = f s with ones in all locations (x, y) of a structuring element's origin at which that structuring element s hits the input image f, i.e. g(x, y) = 1 if s hits f and 0 otherwise, repeating for all pixel coordinates (x, y). Dilation has the opposite effect to erosion. It adds a layer of pixels to both the inner and outer boundaries of regions.

34



**Figure 5.6 Dilation applied to an image.**

**Basics of dilation:**

* Increases the object area
* Used to accentuate features

**Working of dilation:**

1. A kernel(a matrix of odd size(3,5,7) is convolved with the image
2. A pixel element in the original image is ‘1’ if atleast one pixel under the kernel is ‘1’.
3. It increases the white region in the image or size of foreground object increases

**5.1.2.3 USES OF EROSION AND DILATION**

* 1. **Erosion:**
* It is useful for removing small white noises.
* Used to detach two connected objects etc.
  1. **Dilation:**
* In cases like noise removal, erosion is followed by dilation. Because, erosion removes white noises, but it also shrinks the object. So it is dilated. Since noise is gone, it won’t come back, but the object area increases.
* It is also useful in joining broken parts of an object.

**5.1.3 CROP IMAGE**

Cropping is the removal of unwanted areas from the periphery of a photographic or illustrated image. Cropping is one of the most basic photo manipulation

35

processes, and it is carried out to remove an unwanted object or irrelevant noise from the periphery of a photograph, to change its aspect ratio, or to improve the overall composition. Cropping an image extracts a rectangular region of interest from the original image. This focuses the viewer's attention on a specific portion of the image and discards areas of the image that contain less useful information.

The crop function in python takes three parameters:

* image\_path – The file path to the file to be cropped.
* coordinates – A 4-element tuple that contains the beginning and end coordinates.
* saved\_location – The file path to save the cropped file.



**Figure 5.7 Examples of Cropped Image.**

**5.1.4 TEXT IDENTIFICATION**

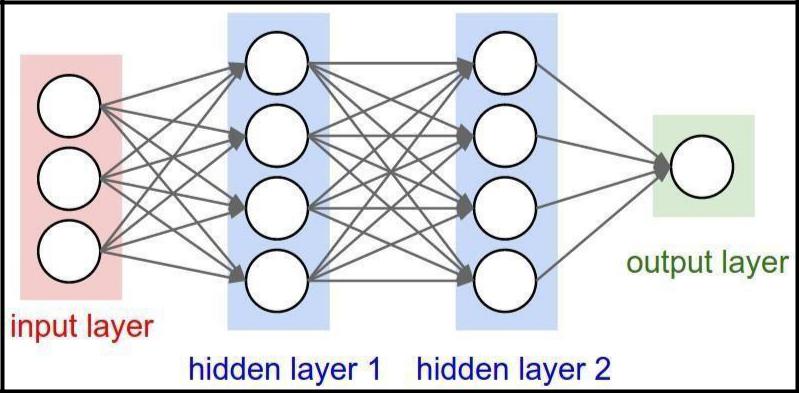
Segmentation part is carried out by the concept of deep learning training data’s in the system and classify them accordingly.

**Deep Learning**: A subset of Machine Learning Algorithms that is very good atrecognizing patterns but typically requires a large number of data. Deep learning excels in recognizing objects in images as its implemented using 3 or more layers of artificial neural networks where each layer is responsible for extracting one or more feature of the image.

**Neural Network**: A computational model that works in a similar way to theneurons in the human brain. Each neuron takes an input, performs some operations

36

then passes the output to the next neuron.



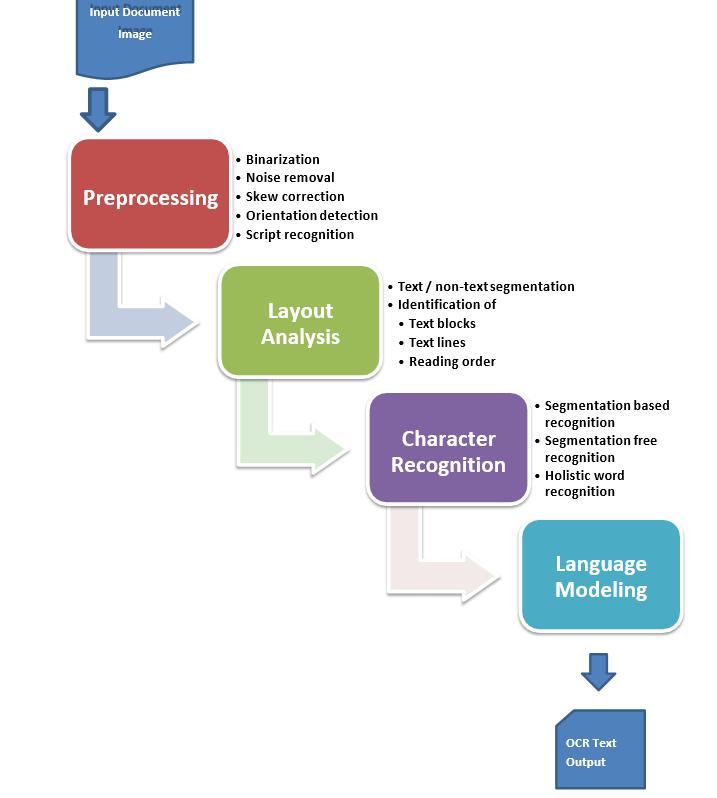
**Figure 5.8 Structure of Neural Network**

**OCR**

OCR technology is widely used in many other fields like mail sorting, education, finance, and government or private offices. It automates the reading of addresses on letters and parcels for efficient mail disbursement. It facilitates in digital archiving of conference proceeding and journals to make them available for online access. Invoice imaging tools help in many businesses to keep track of financial records. In offices, it simplifies the collection of data from printed documents for analysis and further usage. In short, OCR technology has revolutionized the document management process in a wide range of industries by turning a scanned document image into a computer readable text document.

OCR systems transform a two-dimensional image of text, that could contain machine printed or handwritten text, ideally in any script, from its image representation into machine-readable text. OCR systems usually work in a pipeline and there are several steps before actual text recognition takes place. A typical OCR system may comprise preprocessing, layout analysis, character recognition, and language modeling.

37



**Figure 5.9 Optical character recognition.**

It is a widespread technology to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of images containing written text (typed, handwritten or printed) into machine-readable text data.

OCR is often used as a “hidden” technology, powering many well-known systems and services in our daily life. Less known, but as important, use cases for OCR technology include data entry automation, indexing documents for search engines, automatic number plate recognition, as well as assisting blind and visually impaired persons.

**5.1.5 DATABASE VERIFICATION**

Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network. The records of the number plate are saved in MS SQL server. The database contains the details

38

regarding the vehicle. The details include model name , owner name , colour , area and mobile number. This data is fetched via a webpage created using .net .

ASP.NET is an open-source server-side web application framework designed for web development to produce dynamic web pages. It was developed by Microsoft to allow programmers to build dynamic web sites, web applications and web services.

Web Pages is one of many programming models for creating ASP.NET web sites and web applications.

Web Pages provides an easy way to combine HTML, CSS, and server code:

* Easy to learn, understand, and use
* Uses SPA application model (Single Page Application)
  + Similar to PHP and Classic ASP
* VB (Visual Basic) or C# (C sharp) scripting languages

In addition, Web Pages applications are easily extendable with programmable helpers for databases, videos, graphics, social networking and more. The report of the matched number plate is shown as output in the designed webpage.

39

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

The disfigured number plate has been detected using image processing and the data of the vehicle owner is obtained by matching it with the database. Original image of a car is captured and some scratches were done step by step to detect the characters. The preprocessing step is accomplished by means of gaussian blur , grey scale , edge detection and thresholding. Contour extraction is then applied to the preprocessed images to extract features. Then the cropped image of the rectangular region of the number plate is obtained. Optical character recognition is used to identify the damaged characters after erosion and dilation. The detected text which is saved in a text file, matched with the records in the database created using MS SQL server. The matched detail is displayed as the output in a webpage. The result obtained is the details of the owner of the car. The system is able to detect upto level of scratches introduced in the number plate

**6.2 FUTURE ENHANCEMENT**

As future work, we are trying to improve the detection for more level of damage and also to include a camera to capture an image in real time and identify the number plate characters and obtain the details of the vehicle. Further we are trying to identify the details with only partial obtained characters of the number plate.

40

**APPENDIX 1**

**SAMPLE CODING**

import numpy as np

import cv2

from copy import deepcopy

from PIL import Image

#from Knn\_predict import Knn\_predict

import imutils

import pytesseract as tess

import pytesseract

import time

from sklearn.svm import SVC

#from ml\_config import MachineLearningConfig #from ml\_validation import AccuracyValidation import cv2

import numpy as np

pytesseract.pytesseract.tesseract\_cmd =

r"C:\Program Files (x86)\Tesseract-

OCR\tesseract.exe"

def preprocess(img):

cv2.imshow("Input",img)

imgBlurred = cv2.GaussianBlur(img, (5,5), 0)

cv2.imshow("blurred",imgBlurred)

gray = cv2.cvtColor(imgBlurred,

cv2.COLOR\_BGR2GRAY)

cv2.imshow("gray",gray)

41

cv2.waitKey(1)

sobelx = cv2.Sobel(gray,cv2.CV\_8U,1,0,ksize=3)

cv2.imshow("Sobel",sobelx)

cv2.waitKey(0)

ret2,threshold\_img =

cv2.threshold(sobelx,0,255,cv2.THRESH\_BINARY

+cv2.THRESH\_OTSU)

cv2.imshow("Threshold",threshold\_img)

cv2.waitKey(0)

* cv2.destroyAllWindows() return threshold\_img

def cleanPlate(plate):

print ("CLEANING PLATE. . .")

#0gray = cv2.cvtColor(plate,

cv2.COLOR\_BGR2GRAY)

#kernel =

cv2.getStructuringElement(cv2.MORPH\_CROSS,

(3, 3))

thresh= cv2.dilate(gray, kernel, iterations=1)

\_, thresh = cv2.threshold(gray, 127, 255, cv2.THRESH\_BINARY)

im1,contours,hierarchy =

cv2.findContours(thresh.copy(),cv2.RETR\_EXTER

NAL, cv2.CHAIN\_APPROX\_NONE)

42

if contours:

areas = [cv2.contourArea(c) for c in contours]

max\_index = np.argmax(areas)

max\_cnt = contours[max\_index]

max\_cntArea = areas[max\_index]

x,y,w,h = cv2.boundingRect(max\_cnt)

if not ratioCheck(max\_cntArea,w,h):

return plate,None

cleaned\_final = thresh[y:y+h, x:x+w]

#cv2.imshow("Function

Test",cleaned\_final)

return cleaned\_final,[x,y,w,h]

else:

return plate,None

def extract\_contours(threshold\_img):

element =

cv2.getStructuringElement(shape=cv2.MORPH\_RE

CT, ksize=(17, 3))

morph\_img\_threshold = threshold\_img.copy()

cv2.morphologyEx(src=threshold\_img,

43

op=cv2.MORPH\_CLOSE, kernel=element, dst=morph\_img\_threshold)

cv2.imshow("Morphed",morph\_img\_threshold)

cv2.waitKey(0)

* cv2.destroyAllwindows()

cv2.destroyAllWindows()

im2,contours,aaa =

cv2.findContours(morph\_img\_threshold,mode=cv2. RETR\_EXTERNAL,method=cv2.CHAIN\_APPROX \_NONE)

return contours

def ratioCheck(area, width, height):

ratio = float(width) / float(height)

if ratio < 1:

ratio = 1 / ratio

aspect = 4.7272

#aspect = 2

min = 15\*aspect\*15 # minimum area

max = 125\*aspect\*125 # maximum area

#min = 15\*aspect\*15

#max = 80\*aspect\*80

rmin = 3

rmax = 6

44

if (area < min or area > max) or (ratio < rmin or ratio > rmax):

return False

return True

def isMaxWhite(plate):

avg = np.mean(plate)

#if((avg>=150)and(avg<=500)):

if(avg>=115):

return True

else:

return False

def validateRotationAndRatio(rect):

(x, y), (width, height), rect\_angle = rect

if(width>height):

angle = -rect\_angle

else:

angle = 90 + rect\_angle

if angle>15:

return False

if height == 0 or width == 0:

45

return False

area = height\*width

if not ratioCheck(area,width,height):

return False

else:

return True

def cleanAndRead(img,contours):

for i,cnt in enumerate(contours):

min\_rect = cv2.minAreaRect(cnt)

if validateRotationAndRatio(min\_rect):

x,y,w,h = cv2.boundingRect(cnt)

plate\_img = img[y:y+h,x:x+w]

if(isMaxWhite(plate\_img)):

clean\_plate, rect =

cleanPlate(plate\_img)

if rect:

x1,y1,w1,h1 = rect

x,y,w,h =

x+x1,y+y1,w1,h1

46

cv2.imshow("Cleaned

Plate",clean\_plate)

#cv2.waitKey(0)

plate\_im =

Image.fromarray(clean\_plate)

img =

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

cv2.imshow("Detected

Plate",img)

cv2.destroyAllWindows()

def crop\_img(img,countours):

for i,cnt in enumerate(contours):

min\_rect = cv2.minAreaRect(cnt)

if validateRotationAndRatio(min\_rect):

x,y,w,h = cv2.boundingRect(cnt)

plate\_img = img[y:y+h,x:x+w]

grayplate = cv2.cvtColor(plate\_img,

cv2.COLOR\_BGR2GRAY)

cv2.imshow("croppped Plate",grayplate)

cv2.waitKey(0)

cv2.destroyAllWindows()

cv2.imwrite('cropped.png',grayplate)

cv2.imshow("croppped Plate",grayplate)

cv2.waitKey(1)

47

cv2.destroyAllWindows()

print ("Detected Text : ")

ret,thresh2 =

cv2.threshold(grayplate,50,255,cv2.THRESH\_BINA

RY\_INV)

kernel = np.ones((3,3),np.uint8)

thresh2 = cv2.dilate(thresh2,kernel,iterations

= 1)

kernel = np.ones((1,1),np.uint8)

thresh2 = cv2.erode(thresh2,kernel,iterations

= 1)

return grayplate,thresh2

if \_\_name\_\_ == '\_\_main\_\_':

print("DETECTING PLATE . . .")

img = cv2.imread("image/3.jpg") threshold\_img = preprocess(img) contours= extract\_contours(threshold\_img) grayplate,thresh2= crop\_img(img,contours) cv2.imwrite('inputimage.jpg',grayplate) im2,contours,hierarchy=

cv2.findContours(thresh2,mode=cv2.RETR\_EXTER NAL,method=cv2.CHAIN\_APPROX\_NONE)

contours1 = contours[0].reshape(-1,2)

img1 = thresh2.copy()

(h, w) = img1.shape[:2]

48

image\_size = h\*w

mser = cv2.MSER\_create()

mser.setMaxArea(int(image\_size/2))

mser.setMinArea(10)

regions, rects = mser.detectRegions(thresh2)

ii = 1

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

color=(255, 0, 0), thickness=1)

crop\_img = img1[y:y+h, x:x+w]

I1 = cv2.resize(crop\_img,(20,20))

testImgDet = I1.reshape(1, -1)

text =

pytesseract.image\_to\_string(Image.open('inputimage.

jpg'))

print(text)

cv2.imshow(" binary croppped Plate",img1)

cv2.waitKey(0)

cv2.destroyAllWindows()

file =

open('G:/Damaged\_Number\_plate\_Baseline/sample.t

xt','w')

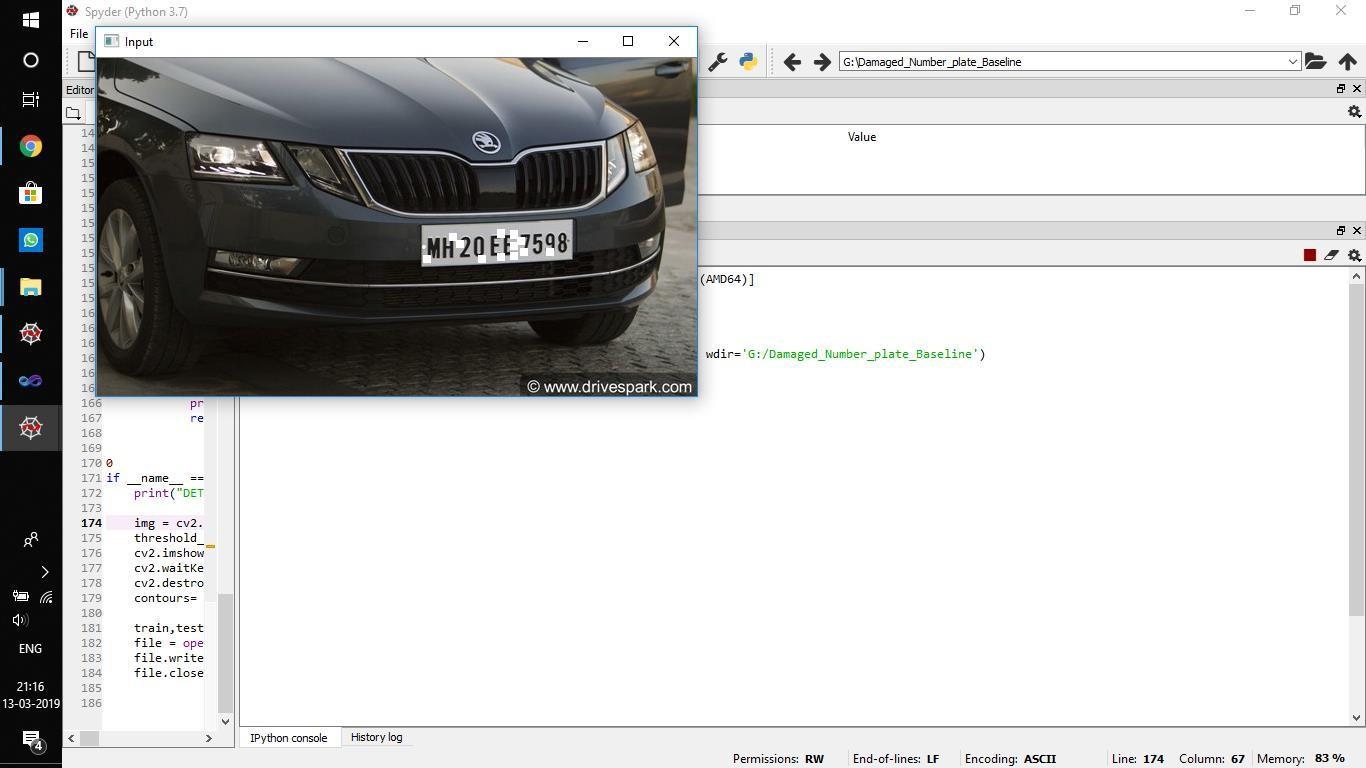
file.write(text)

file.close()

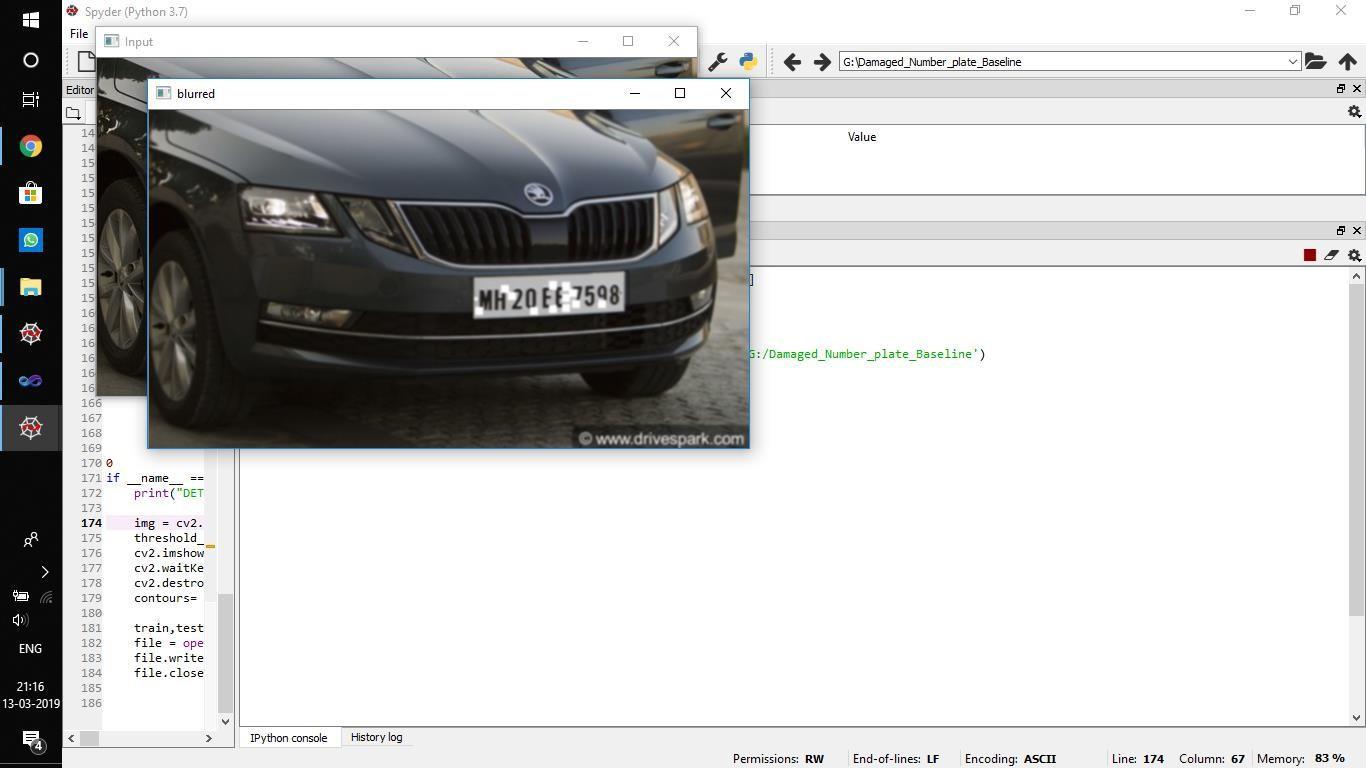
49

**APPENDIX 2**

**SCREENSHOTS**

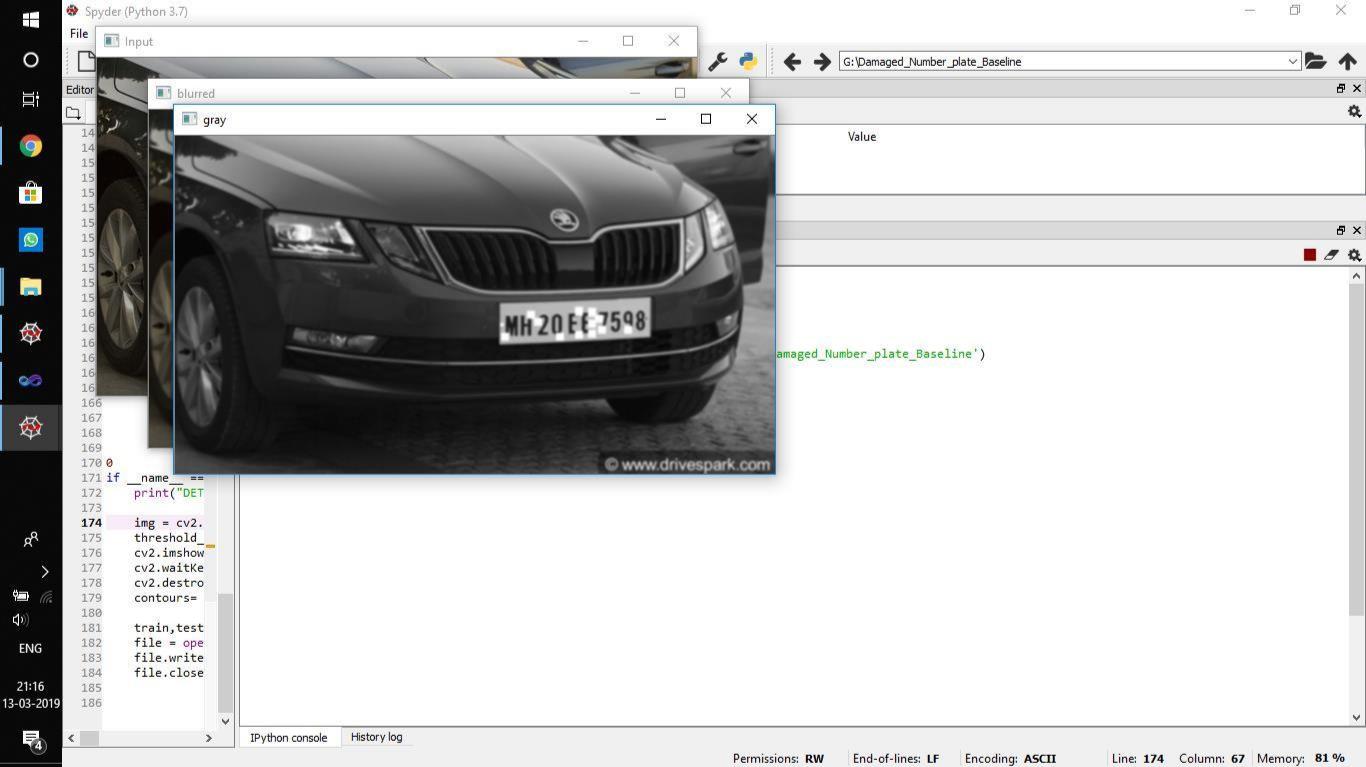


**INPUT IMAGE**

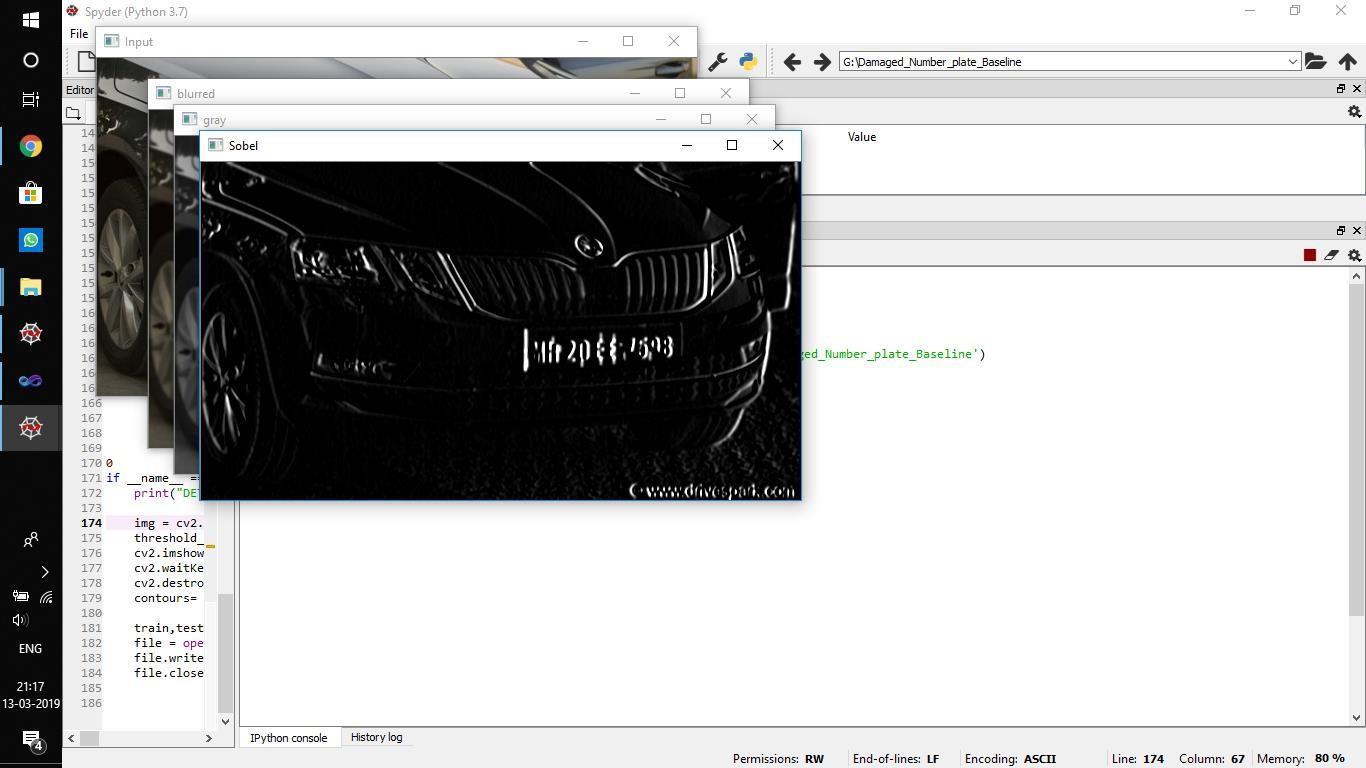


**BLURRED IMAGE OF THE INPUT**

50



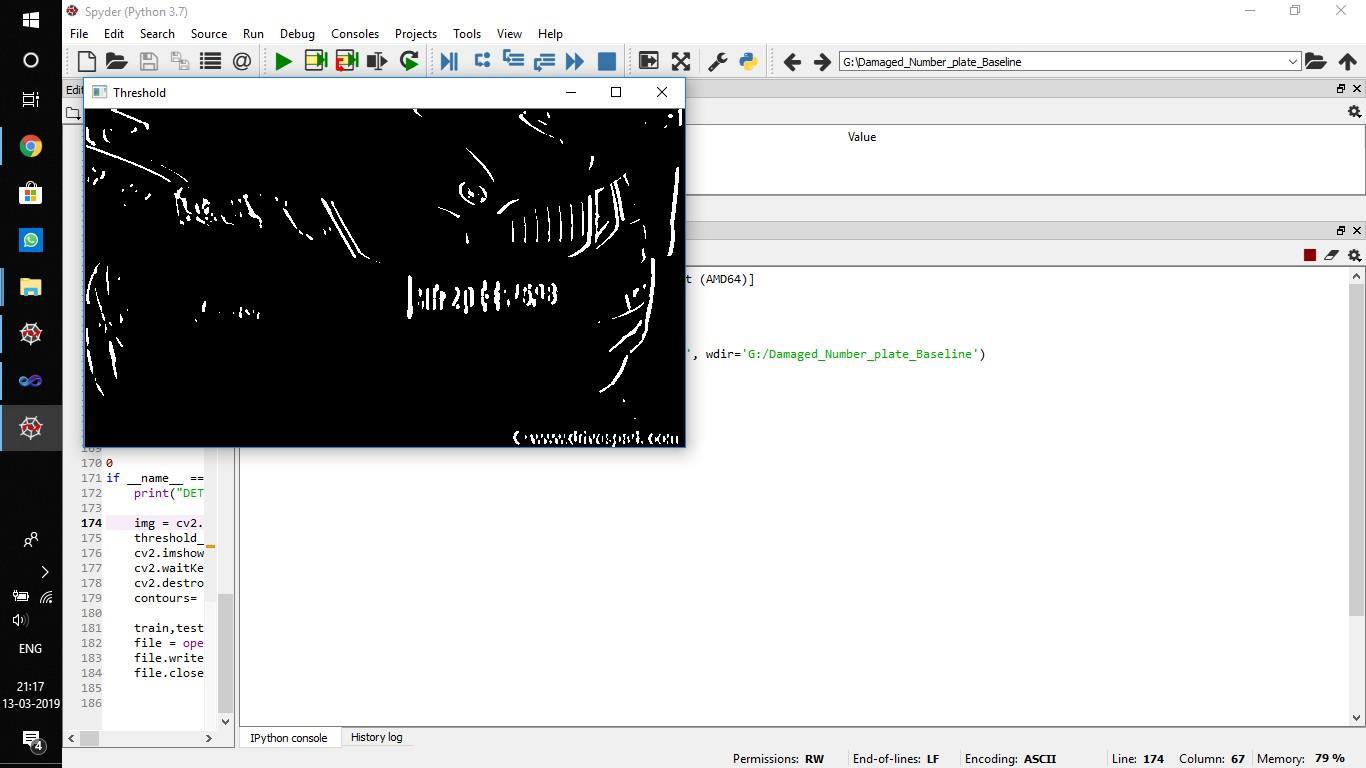
**GRAY SCALE CONVERSION OF THE INPUT**



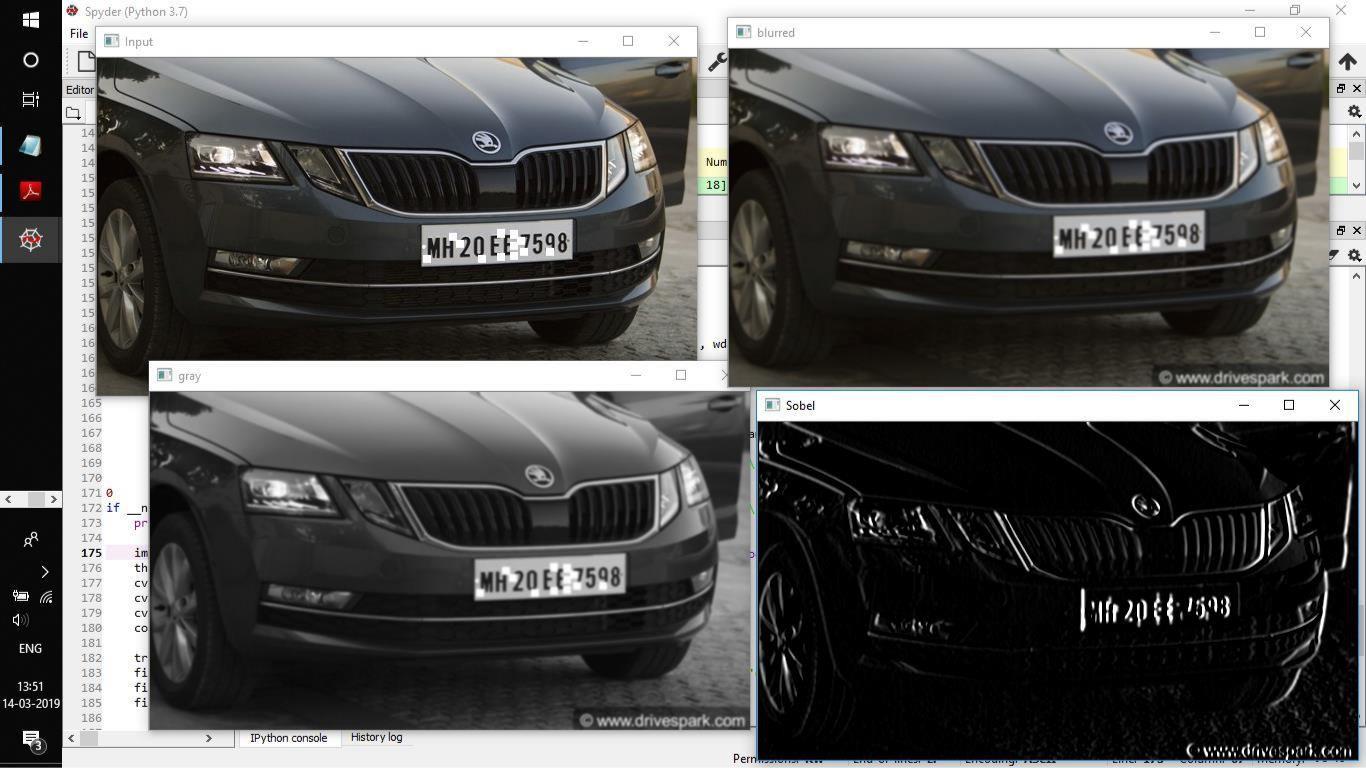
**SOBEL ALGORITHM FOR EDGE** **DETECTION IN**

**THE INPUT IMAGE**

51

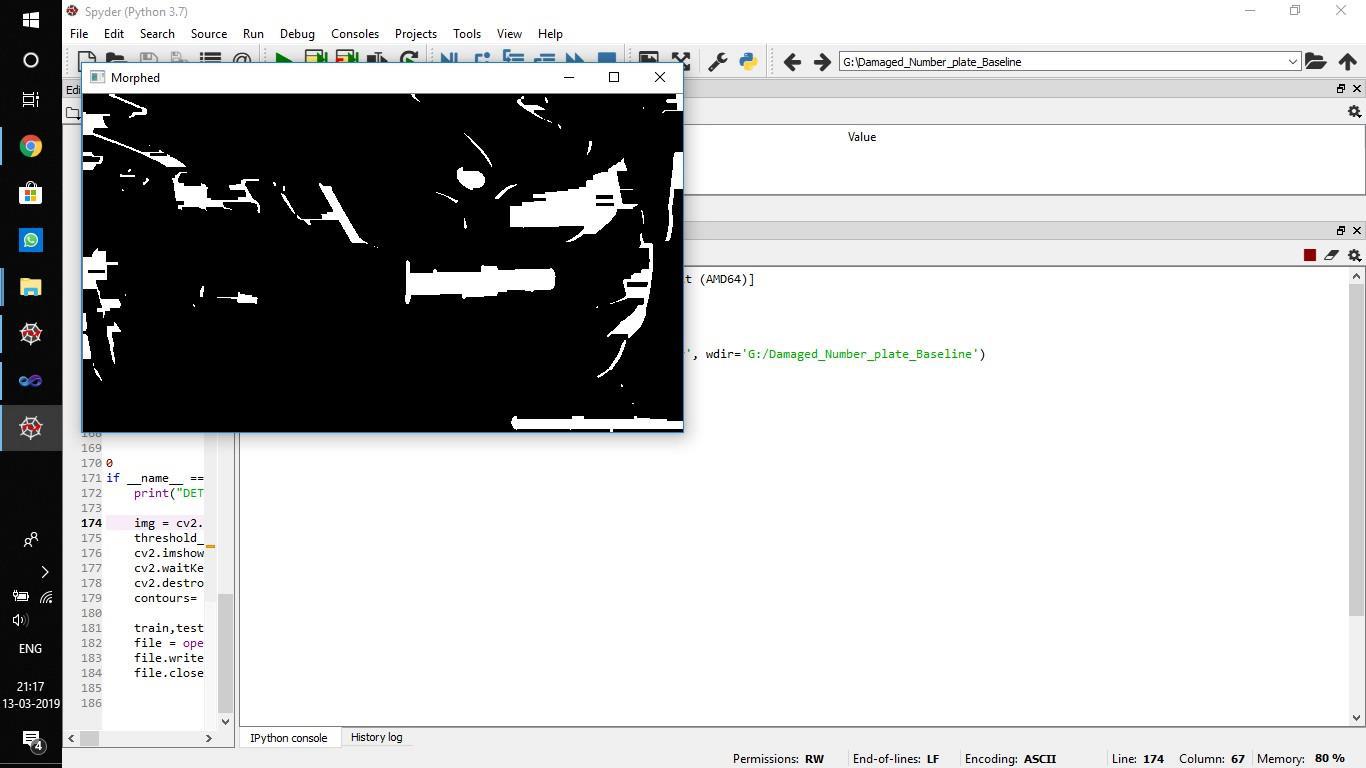


**THRESHOLDING OF INPUT IMAGE**

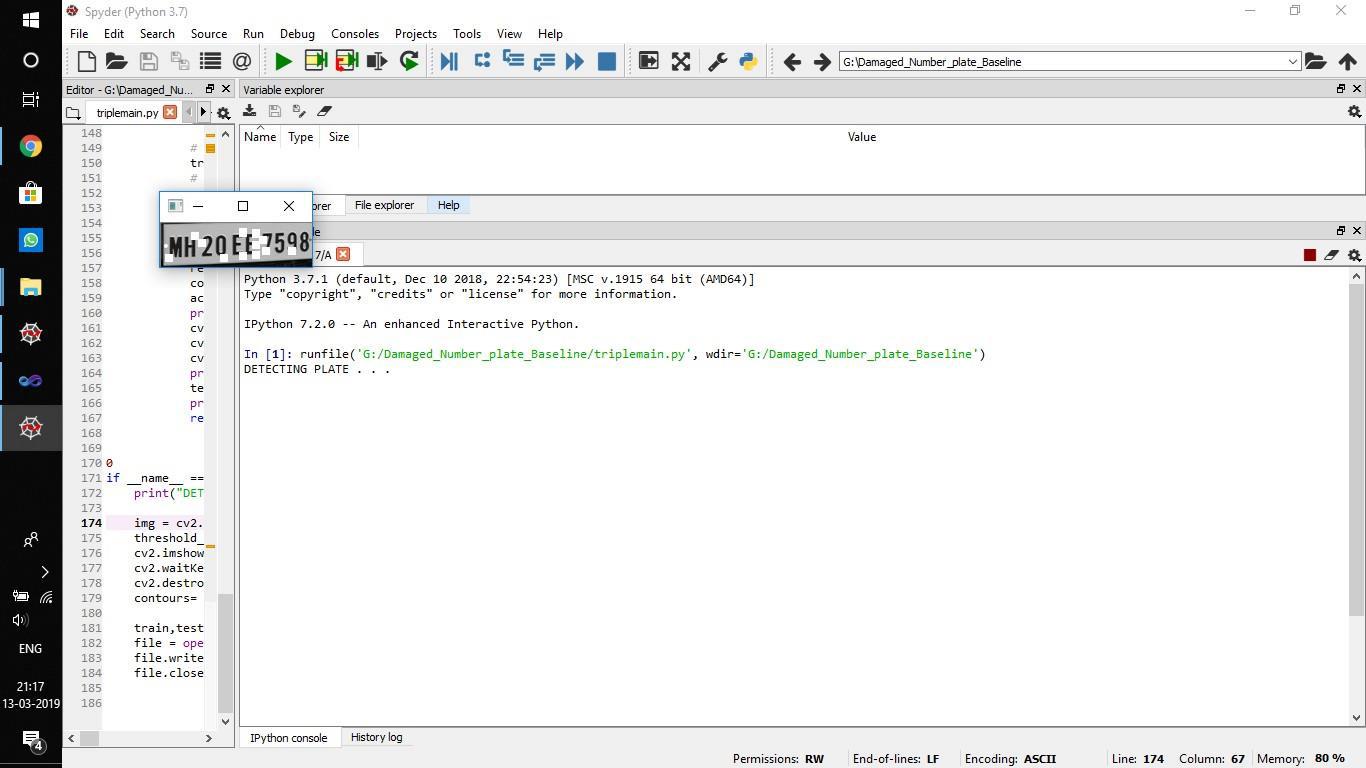


**PREPROCESSING OF THE INPUT IMAGE**

52

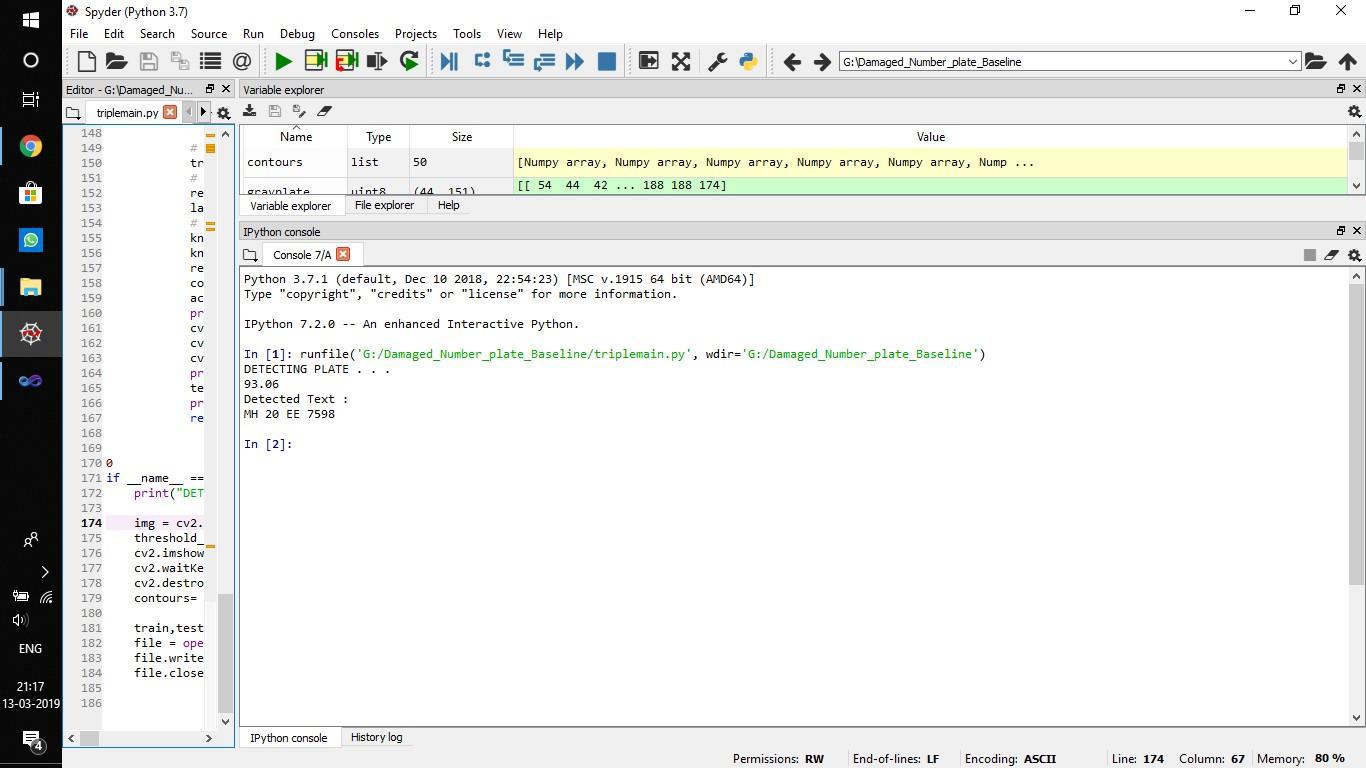


**EXTRACTING CONTOURS IN THE GIVEN INPUT IMAGE USING MORPHOLOGY FUNCTION**

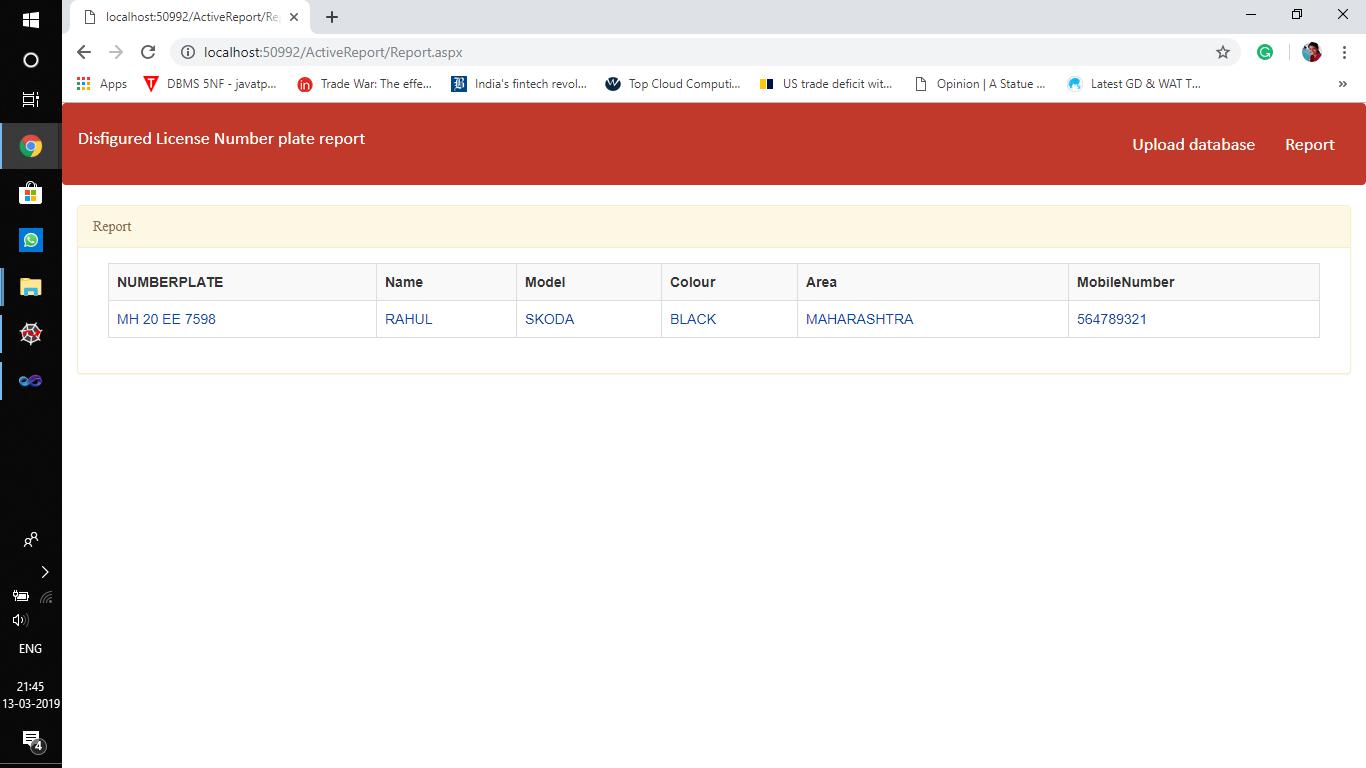


**CROPPED IMAGE OF THE INPUT IMAGE**

53



**DETECTED TEXT**



**DETAILS OF THE CORRESPONDING VEHICLE**

54

**REFERENCES**

1. S.Jayapradha, Dr.S.P.Manikandan ,2017,” Vehicle License Plate Tracking and Owner Details Identification”, Computer Engineering, International Journal of

Informative & Futuristic Research, vol. 4,no. 8,pp : 7071-7078

[2] Rahim Panahi, Iman Gholampour, 2016,” Accurate Detection and Recognition of Dirty Vehicle Plate Number for High-Speed Applications”, IEEE transactions on intelligent transportation systems.

1. Prof.Amit Kukreja, Swati Bhandari, Sayali Bhatkar, Jyoti Chavda, Smita Lad, 2017,“Indian Vehicle Number Plate Detection Using Image Processing”,

International Research Journal of Engineering and Technology,vol.4,no.4.

[4] Mahmood Ashoori-Lalimi, Sedigheh Ghofrani, 2011, “An Efficient Method for Vehicle License Plate Detection in Complex Scenes”, Circuits and Systems, IEEE Journal, vol. 2, pp : 320-325.

1. Bhonsale Tejas, Dhamal Omkar, Dhumal Rutuj, Khedekar Prajakta, Patil Bhakti, 2016,“Number Plate Recognition and Document Verification using Feature

Extraction OCR Algorithm “, International Journal of Engineering And Computer science, vol. 5,no. 11, pp : 19177-19179.

1. Sarbjit Kaur, Sukhvir Kaur, 2014,“An Efficient Approach for Automatic Number Plate Recognition System under Image Processing”, International Journal

of Computer science and Information Technology, vol. 5,no. 3, pp :2954-2959.

1. Cosmo H.Munuo, Dr. Michael Kisangiri, 2014,” Vehicle Number Plates Detection and Recognition using improved Algorithms: A Review with Tanzanian

Case study”, International Journal of Engineering and Computer Science, vol.3, no. 5, pp :5828- 5832.

[8] Divya Gilly, Kumudha Raimond, PhD,2013, “A Survey on License Plate

55

Recognition Systems”, International Journal of Computer Application,vol. 61, no. 6.

1. Sourav Roy, Amitava Choudhury, Joydeep Mukherjee, 2013, “An Approach towards Detection of Indian Number Plate from Vehicle”, International Journal

of Innovative Technology and Exploring Engineering, vol. 2, no. 4.

[10] Mohanad Alata, Mohammad Al-Shabi, 2006, “Text detection and character recognition through fuzzy logic”, Journal of Electrical Engineering, vol. 57, no. 5,pp : 258-267.

56

57