ABSTRACT

License Plate Recognition was a computer system that recognizes any digital image automatically on the number plate. This system includes various operations such as taking pictures, localizing the number pad, truncating characters and OCR from alphanumeric characters. The main idea of this system is to design and develop effective image processing techniques and algorithms to localize the license plate in the captured image, to divide the characters from that number plate and to identify each character of the segment by using the Open Computer Vision Library. This has been implemented in K-NN algorithm and python programming language. Many applications can be implemented by using this system, such as security, highway speed detection, violation of light, identification of handwritten text, discovery of stolen cars, automatic fee collection systems.

INTRODUCTION

People from different countries interact in a multicultural environment to develop solutions to never-ending problems for men. The Open Source section is a one of the outstanding contribution in the scientific world is Python. Computer vision in the Intel's research has been producing a fruit called Open Computer Vision (Open CV), which can support the development of computer vision. At present, the use of vehicles is increasing throughout the country. All of these vehicles have a unique vehicle identification number as their main identifier. The ID is actually in the license number that refers to a legal license to participate in the public movement. Each vehicle in the world must have its own number plate that must be installed on its body (at least on the back). They need to Identify the vehicles are increasing in parallel with the number of vehicles. This identification system helps with safety, automatic switching systems, highway speed detection, light detection, stolen vehicle detection, and human and non-human loss collection systems. The auto license plate recognizing system replaces the manual license plate number writing process in the computer system.

In order to obtain an appropriate personal recognition, the license plate identification technique consists of three main topics. They are, find the location of the panel of digital images, segmentation the characters from the pictures of the panel and the visual character Recognition. The most dominant and basic step is to determined the exact location of the number plate in the captured image. The localization of a license plate has been recognized either by structural analysis and color analysis method. In the License panel area, unwanted spots are removed by parsing the connected component. ANPR is a collective control system that captures the vehicle image and identifies the license number. Some ANPR system applications

are automatic traffic control and tracking system, highway toll collection / automatic parking systems, petrol station automation, flight time monitoring. These systems automate the process of identifying vehicle license number, making it fast, cost effective.

Literature Survey:

Searching for license plate recognition is still a challenge. It involves three major steps. They specify number pad space, character segmentation, and character recognition. Each step suggested different ways to improve efficiency.

One of these methods [1] used the adaptive threshold to highlight the characters and suppress the background. In order to remove unwanted image spaces, a component algorithm is first applied to the converted binary image from the original panel. A special algorithm called Image Scissoring is used to divide the Optical Character Recognition engine called tesseract, which returns ASCII to the license number. The entire system has been implemented using open CV.

Another method [2] is to deploy the forward background feed method for character classification. The neural network is developed by using the backward-propagation algorithm. Normalization, scale and edge detection are included in the steps of the preprocessing. The horizontal and vertical graph and component survey are able to address the problem of character fragmentation.

[3] Another way in which character areas are selected is through binarization, connected component analysis. The Point Analysis method removes unwanted points and combines split points and split points. This unit achieves a 97.2% accuracy rate in character segmentation. The reliability of the recognition was 90.9%.

[4] Offers an approach that relies on effective morphological operation and the detection method of Sobel Edge. This approach is simplified to divide all letters and numbers used in the number pad using the surround box

method. After the template is fragmented, the matching policy is used to recognize numbers and characters. This whole system was implemented using MATLAB.

[5] Provides an overview of the analysis of related components and processes, such as aspect ratio analysis and pixel count analysis.

In [6] the author studies a comparison of four algorithms that are sequentially using statistical properties, the Hough Transform and Contour algorithm, the medium transformation approach and morphological processes and their results.

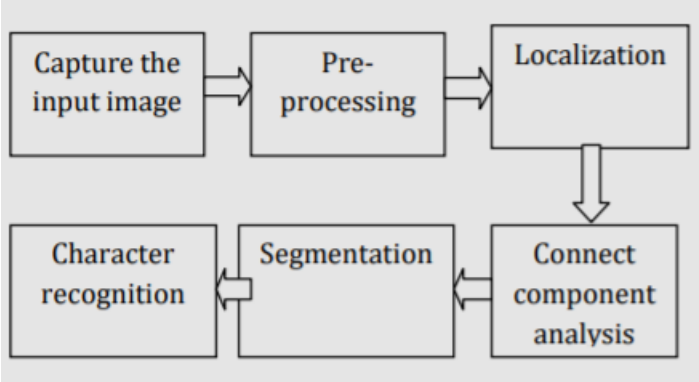
The handwritten text [7] is fragmented by the watershed algorithm. Noise removal, slope correction, budgeting and normalization were eliminated in pre-treatment. After fragmentation the process of extracting a segmented image is done by a reverse integer to convert the wavelet integer. The classification is then sorted by neuroscience

Existing System:

In many countries ANPR methods have been implemented such as Australia, Korea and a few other countries . In the development of ANPR system in many countries the number plate standards are strictly implemented. These systems use standard features for license plates such as: panel dimensions, panel borders, color and letter characters, etc., which help to easily localize the number pad and specify the car license number. In India, plate number standards are rarely followed. There are wide variations in font types, text, size, position, and colors of number plate. In a few cases, there are other undesirable decorations on the number panel. Also, different other countries, there are no special features on Indian number panel to facilitate recognition. Thus, only manual recording systems are currently being used and ANPR has not been commercially developed in India.

Proposed System:

In India, basically, there are two kinds of license-plates, black characters in white plate and black characters in yellow plate, the former for private vehicles and latter for commercial, public service vehicles. The system tries to address these two categories of plates, the high-level block diagram of the proposed system shown below.



CAPTUERE THE INPUT IMAGE:

The car's number pad is taken from a high resolution camera. The resolution of the number plate recognition system depends on the captured image. The image captured in RGB format must be converted to a gray image

PRE-PROCESSING

Pre-processing is a set of algorithms applied to the image to improve the quality by which the gray image is converted to a binary image. Before converting to a binary image, the image is smoothed to reduce noise. Pre-processing can be done by the threshold algorithm. There is a different kind of threshold like

- Global threshold
- Adaptive mean threshold
- Adaptive Gaussian threshold

Global threshold: The threshold is a nonlinear process where two levels are assigned to pixels lower or bigger than the threshold value specified. The threshold value is constant. The grayscale picture is converted to convert the binary image according to the formula Dst (x, y)= {max value if src(x,y)>T(x,y) 0 otherwise Where T (x, y) is the threshold calculated individually for each pixel. Average adaptive threshold: The value of the threshold is the average area of the neighborhood.

Gaussian Adaptive Threshold: Threshold value is the sum of the values of the values of the neighborhood where the weights are a Gaussian window. The gray picture is then converted to a binary picture by the adaptive threshold method. The threshold is the simplest way to divide objects from the background. If the background is relatively same, the global threshold can be used. For large change in background intensity the adaptive threshold is used.

NUMBER PLATE LOCALIZATION

The license plate is extracted using either a shape analysis or a color analysis method. In the General License Panel has in form of a rectangular shape. Thus, algorithms look for geometrical shapes of a rectangular proportion. In India, most license plates are white or yellow, and therefore can also use color analysis. Before you find the rectangle in an image, the image must be in a binary image or the edges of the image should be detected. Then you should find and connect to the relevant rectangular corners. Finally, the areas connected to the box are connected and all rectangular areas of interest are extracted.

4. CONNECT COMPONENT ANALYSIS

To remove the unwanted image space, the algorithm of the component connected to the binary filter is applied first. The parsing of the connected component is done to determine the characters in the image. The basic proposal is to pass through the image and find a connected pixel. Each component (dots) is distinguished and extracted.

5. SEGMENTATION

Once the license plate has been extracted, each character must be fragmented. For component division, the component label is used to see the computer in order to discover the connected areas in binary digital images. The label of connected components works by scanning a pixel-in-pixel image from top to down to find connected pixels and connected pixel cards.

6. CHARACTER RECOGNITION

To identify characters, the segmented characters in the license panel must match the templates that are already created. The recognition process returns the license number in ASCII format and saves it in a text document.

In this recognition is a two-track process. In the first pass, an attempt was made to identify each word in turn. Each satisfactory word is passed to the adaptive workbook as training data. The adaptive workbook gets an opportunity to learn the text more accurately.

Implementation:

A. Capture The image of the vehicle is captured using a high resolution photographic camera. A better choice is an Infrared (IR) camera. The camera may be rolled and pitched with respect to the license plates.

Starting from point one of capturing the image, OpenCV library has highly optimized algorithms for all image processing operations. OpenCV provides interface for different camera models. The following code snippet explains how to interface an in-built web camera and capture a frame.

```
from opencv import highgui as hg

capture = hg.cvCreateCameraCapture(0)

hg.cvNamedWindow("Snapshot")

 frame = hg.cvQueryFrame(capture)

hg.cvShowImage("Snapshot", frame)
```

B. Preprocess Preprocessing is the set algorithms applied on the image to enhance the quality. It is an important and common phase in any computer vision system. For the present system preprocessing involves two processes: Resize – The image size from the camera might be large and can drive the system slow. It is to be resized to a feasible aspect ratio. Convert Colour Space – Images captured using IR or photographic cameras will be either in raw format or encoded into some multimedia standards. Normally, these images will be in RGB mode, with three channels (viz. red, green and blue). Number of channels defines the amount colour information available on the image. The image has to be converted to grayscale.

As seen before, preprocessing involves resizing and changing colour spaces of the source image. Like any other image processing toolkits, OpenCV also provides fast and quick procedures.

Resize original = cv.LoadImageM("image.jpg")

thumbnail = cv.CreateMat(original.rows / 10, original.cols / 10, original.type)

cv.Resize(original, thumbnail)

The original image is resized to the dimensions specified in the thumbnail object. Colour space conversion

CvtColor(original,gray,CV_RGB2GRAY)

The above line of code converts the original image to gray. More image conversion codes are available at the OpenCV Wiki.

C. Localize Rear or front part of the vehicle is captured into an image. The image certainly contains other parts of the vehicle and the environment, which are of no requirement to the system. The area in the image that interests us is the license plate and needs to be localized from the noise.

the image is converted to black and white. There are two motivations for this operation –

1. Highlighting characters and

2. Suppressing background. Localization is done by an image processing technique called Thresholding. The pixels of the image are truncated to two values depending upon the value of threshold. Threshold requires pre-image analysis for identifying the suitable threshold value. Adaptive thresholding technique determines a local optimal threshold value for each image pixel so as to avoid the problem originating from non-uniform illumination.

Threshold operation is performed in this phase.

To retain the image quality, adaptive threshold algorithms are to be used. Previous researches have concluded that Otsu's thresholding algorithm is the efficient way of binarizing the image. OpenCV provides complex and efficient adaptive thresholding algorithms including Otsu method.

cvThreshold(image, binary_image,128,255, CV_THRESH_OTSU)

The above line of code returns a binary image which is adaptively thresholded. The arguments follow the order:

1. Source image,

2. Destination image,

3. Threshold value,

4. Resultant value, and 5. Type of threshold. The type CV_THRESH_OTSU performs Otsu algorithm on the source image.

D. Connected Component Analysis

In order to eliminate undesired image areas, a connected component algorithm is first applied to the binarized plate candidate. Connected component analysis is performed to identify the characters in the image. Basic idea is to traverse

through the image and find the connected pixels. Each of the connected components (blobs) are labelled and extracted. bellow Fig. shows the filtered blobs.

cvBlobsLib is a library to perform binary images connected component labelling. It also provides functions to manipulate, filter and extract results from the extracted blobs. The library provides two basic functionalities:

Extract 8-connected components in binary or grayscale images.

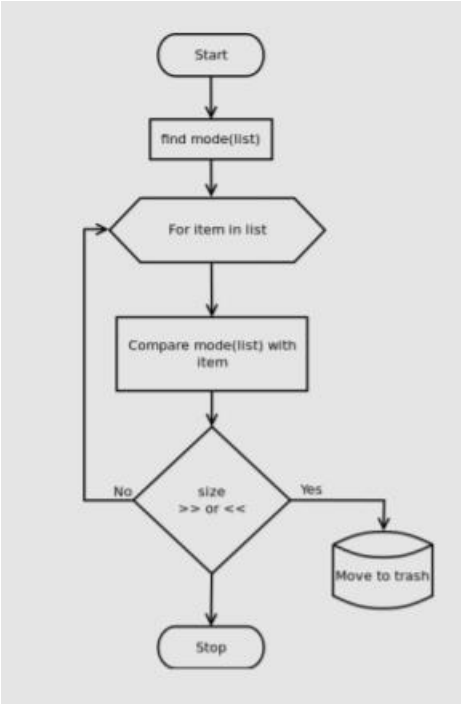
- Filter the obtained blobs to get the interest objects in the image. This is performed using the Filter method from CBlobResult.
- The library is thread-safe if different objects per thread are used.

myblobs = CBlobResult(binary_image, mask, 0, True)

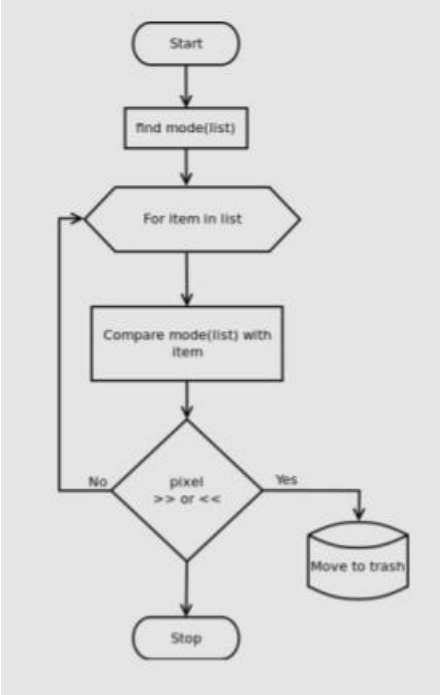
myblobs.filter_blobs(325,2000)

blob_count = myblobs.GetNumBlobs()

- The connected components are labelled using the above code snippet. filter_blobs method is used to filter out the blobs of required dimensions.



Aspect ratio based elimination



Pixel coordinate based selection

E. Segmentation Segmentation is the process of cropping out the labelled blobs. These blobs are expected to be the required portion of the license number. A special algorithm called Image Scissoring is introduced here. In this algorithm, the license plate is vertically scanned and scissored at the row on which there is no white pixel and the scissored area is copied into a new matrix.

Image Scissoring is hard-coded in Python by scanning the image vertically and cropping out white portions. The algorithm, is fast and efficient than compared to other predefined image cropping techniques. Segmentation phase also involves classification of the collected blobs and recording only the essential ones. Undesirable blobs occur even after segmentation.

These are removed by two methods:

1) Aspect ratio based elimination.

2) Pixel coordinate based selection. Aspect ratio based elimination: The aspect ratio (row/column) of each blob is calculated and recorded.

A binarized candidate is sure of containing more characters than unwanted blobs. The mean of the aspect ratios are calculated and compared to all the blobs in turn. .

If anyone of them has a larger deviation, that blob is removed from the candidate. This algorithm was deviced based on research and experiment through out the process. The dynamic nature of Python is exploited in every step of this algorithm. Pixel coordinate based selection: This algorithm thrives on the fact that license numbers are occuring in the plate in a single set of rows. Effectively, we can detect the edge of the license plate, and select the blobs coming between the minimum and maximum row coordinates. This can reduce the amount of unwanted blobs and make the system more accurate.

Character Recognition Finally, the selected blobs are send to a Optical Character Recognition (OCR) Engine, which returns the ASCII of the license number.

Tesseract OCR engine has a Python wrapper, which make character recognition quick and easy.

```
from tesseract import image_to_string

image = open("blob.jpg")

 text = image_to_string(image)
```

ALGORITHM DESCRIPTON

1: Begin

2: Input: Original Image

3: Output: Characters

4: Method: K-Nearest Neighbors

5: LP: License Plate

6: Convert RGB image to Grayscale

7: Filter Morphological Transformation

8: Transforms Grayscale image to binary image

9: Filter Gaussian for Blurs image

10: Finding all contours in image

11: Search & recognize all possible character in image

12: Crop part of image with highest candidate LP

13: Crop the LP from original image

14: Apply steps from 6 to 11 again on crop image

15: Print the characters in LP

16: End

K-Nearest Neighbors algorithm (or KNN):

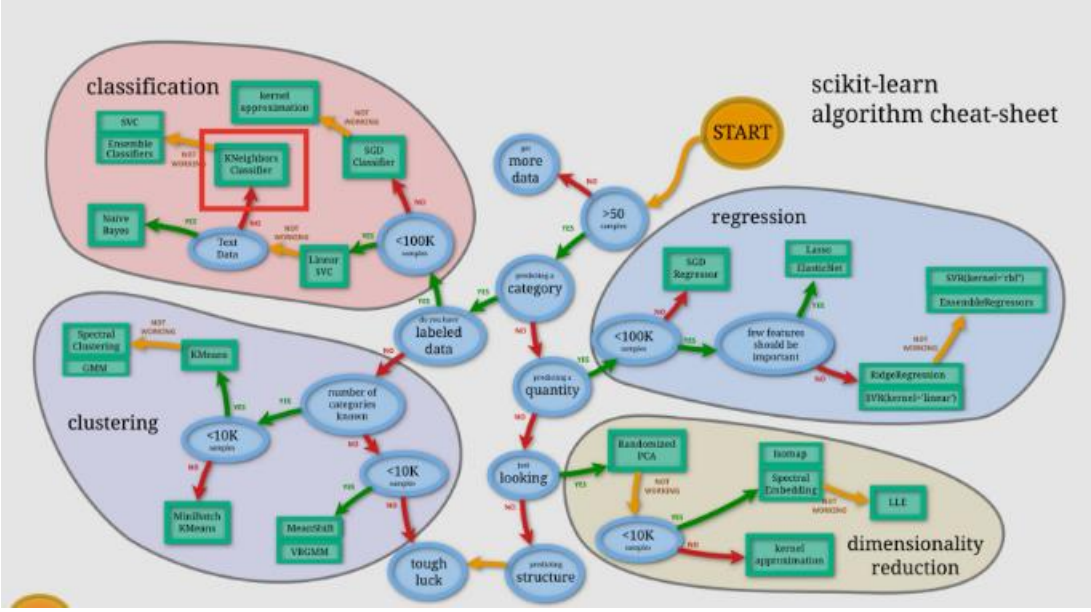
KNN algorithm is one of the simplest classification algorithm and it is one of the most used learning algorithms. So what is the KNN algorithm? I'm glad you asked! KNN is a non-parametric, lazy learning algorithm. Its purpose is to use a database in which the data points are separated into several classes to predict the classification of a new sample point.

Just for reference, this is "where" KNN is positioned in the algorithm list of scikit learn. BTW, scikit-learn documentation (clickable link) is amazing! Worth a read if you're interested in machine learning.

When we say a technique is non-parametric , it means that it does not make any assumptions on the underlying data distribution. In other words, the model structure is determined from the data. If you think about it, it's pretty useful, because in the "real world", most of the data does not obey the typical theoretical assumptions made (as in linear regression models, for example). Therefore, KNN could and probably should be one of the first choices for a classification study when there is little or no prior knowledge about the distribution data.

KNN is also a lazy algorithm (as opposed to an eager algorithm). Does that mean that KNN does nothing, like these polar bears imply??? Not quite. What this means is that it does not use the training data points to do any generalization. In other words, there is no explicit training phase or it is very minimal. This also means that the training phase is pretty fast . Lack of generalization means that KNN keeps all the training data. To be more exact, all (or most) the training data is needed during the testing phase.

KNN Algorithm is based on feature similarity: How closely out-of-sample features resemble our training set determines how we classify a given data point:



Example of k-NN classification. The test sample (inside circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3 (outside circle) it is assigned to the second class because there are 2 triangles and only 1 square inside the inner circle. If, for example k = 5 it is assigned to the first class (3 squares vs. 2 triangles outside the outer circle).

KNN can be used for classification — the output is a class membership (predicts a class — a discrete value). An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors. It can also be used for regression —

output is the value for the object (predicts continuous values). This value is the average (or median) of the values of its k nearest neighbors.

A few Applications and Examples of KNN

Credit ratings — collecting financial characteristics vs. comparing people with similar financial features to a database. By the very nature of a credit rating, people who have similar financial details would be given similar credit ratings. Therefore, they would like to be able to use this existing database to predict a new customer's credit rating, without having to perform all the calculations.

Should the bank give a loan to an individual? Would an individual default on his or her loan? Is that person closer in characteristics to people who defaulted or did not default on their loans?

In political science — classing a potential voter to a "will vote" or "will not vote", or to "vote Democrat" or "vote Republican".

More advance examples could include handwriting detection (like OCR), image recognition and even video recognition.

Some pros and cons of KNN

Pros:

No assumptions about data — useful, for example, for nonlinear data

Simple algorithm — to explain and understand/interpret

High accuracy (relatively) — it is pretty high but not competitive in comparison to better supervised learning models

Versatile — useful for classification or regression

Cons:

- Computationally expensive — because the algorithm stores all of the training data
- High memory requirement
- Stores all (or almost all) of the training data
- Prediction stage might be slow (with big N)
- Sensitive to irrelevant features and the scale of the data

Quick summary of KNN

- The algorithm can be summarized as:
- A positive integer k is specified, along with a new sample
- We select the k entries in our database which are closest to the new sample
- We find the most common classification of these entries
- This is the classification we give to the new sample
- A few other features of KNN:
- KNN stores the entire training dataset which it uses as its representation.
- KNN does not learn any model.

KNN makes predictions just-in-time by calculating the similarity between an input sample and each training instance.

Software & Hardware requirements

HARDWARE REQUIREMENTS:

- Processor : Intel i3 and above
- RAM : 4GB and Higher
- Hard Disk : 500GB: Minimum

SOFTWARE REQUIREMENTS:

- Programming Language / Platform : Python
- IDE : pycharm
- Web Framework : DJango
- UI Technologies : html, css, js
- Database : sqlite

LIBRARIES

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. It contains various features including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and FORTRAN code
- Useful linear algebra, Fourier transforms, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas:

Pandas are an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using

Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load,

prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. Standard Python distribution doesn't come bundled with Pandas module. A lightweight alternative is to install NumPy using popular Python package installer, pip. pip install pandas

Open CV:

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.

OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

Computer Vision

Computer Vision can be defined as a discipline that explains how to reconstruct, interrupt, and understand a 3D scene from its 2D images, in terms of the properties of the structure present in the scene. It deals with modeling and replicating human vision using computer software and hardware.

Computer Vision overlaps significantly with the following fields – Image Processing – It focuses on image manipulation.

Pattern Recognition – It explains various techniques to classify patterns.

Photogrammetry – It is concerned with obtaining accurate measurements from images.

Computer Vision Vs Image Processing

Image processing deals with image-to-image transformation. The input and output of image processing are both images.

Computer vision is the construction of explicit, meaningful descriptions of physical objects from their image. The output of computer vision is a description or an interpretation of structures in 3D scene.

Features of OpenCV Library

Using OpenCV library, you can – Read and write images Capture and save videos Process images (filter, transform)

Perform feature detection

Detect specific objects such as faces, eyes, cars, in the videos or images.

Analyze the video, i.e., estimate the motion in it, subtract the background, and track objects in it.

OpenCV was originally developed in C++. In addition to it, Python and Java bindings were provided. OpenCV runs on various Operating

Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

This tutorial explains the concepts of OpenCV with examples using Java bindings.

STUDY OF THE SYSTEM

To provide flexibility to the users, the interfaces have been developed that are accessible through a browser. The GUI'S at the top level have been categorized as

Analysis

Although the scale of this project is relatively small, to produce a professional solution is it imperative that the current problem is understood accurately. However, this task has been made doubly difficult by the lack of support from the company. Thankfully, the Application manager has been kind enough to spare me some of his own time to discuss the problem with me further. Therefore, this chapter is concerning with analyzing the current situation and expectations of the user for this system.

Requirements:

The minimum requirements of the project are listed below:

- Examine the tools and methodologies required to gain an overview of the system requirements for the proposed database.
- Examine suitable database management systems that can be used to implement the proposed database.
- Evaluate appropriate website authoring and web graphic creation tools that can be used to develop web based forms for the proposed database
- Produce and apply suitable criteria for evaluating the solution

Requirement Analysis:

Taking into account the comparative analysis stated in the previous section we could start specifying the requirements that our website should achieve. As a basis, an article on all the different requirements for software development was taken into account during this process. We divide the requirements in 2 types: functional and nonfunctional requirements.

Functional requirements

Functional requirement should include function performed by a specific screen outline work-flows performed by the system and other business or compliance requirement the system must meet.

Functional requirements specify which output file should be produced from the given file they describe the relationship between the input and output of the system, for each functional requirement a detailed description of all data inputs and their source and the range of valid inputs must be specified.

The functional specification describes what the system must do, how the system does it is described in the design specification.

If a user requirement specification was written, all requirements outlined in the user requirements specifications should be addressed in the functional requirements.

- The user should be able to register and manage his appointments online at any time.
- Database has to store all the information efficiently without any information loss.
- The user shall be able to search for the doctors by specialty, name, working time and/or gender.
- The user can change his profile info at any time
- hospital can manage all appointments made with him on his account

Functional Requirements:

1) Loading Dataset
2) Data cleaning
3) Data Transformation
4) Data Visualization
5) Choosing Algorithm
6) Train Dataset
7) Validate Dataset
8) Record Accuracy
9) Create Model
10) Predict Outcome

Nonfunctional requirement

Describe user-visible aspects of the system that are not directly related with the functional behavior of the system. Non-Functional requirements include quantitative constraints, such as response time (i.e. how fast the system reacts to user commands.) or accuracy (.e. how precise are the systems numerical answers.).

- Portability
- Reliability
- Usability
- Time Constraints
- Error messages
- Actions which cannot be undone should ask for confirmation
- Responsive design should be implemented
- Space Constraints
- Performance
- Standards
- Ethics
- Interoperability
- Security
- Privacy
- Scalabilit

INPUT DESIGN AND OUTPUT DESIGN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

SYSTEM STUDY

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SOFTWARE REQUIREMENT SPECIFICATION

What is SRS?

Software Requirements Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase).

The SRS phase consists of two basic activities:

Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, Specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic of this phase.

Role of SRS:

The purpose of the SRS is to reduce the communication gap between the clients and the developers. SRS is the medium though which the client and user needs are accurately specified.

It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Purpose:

The purpose of this document is to describe all external requirements for the E-learning System. It also describes the interfaces for the system.

- Scope**:**

  This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also by the basis for validating the final deliver system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alternations without the permission of the client.

- Overview**:**

  The SRS begins the translation process that converts the software Requirements into the language the developers will use. The SRS draws on the Use Cases from the user Requirement Document and analyses the situations from a number of perspectives

to discover and eliminate inconsistencies, ambiguities and omissions before development progresses significantly under mistaken assumptions.

Proposed System Architecture:

The proposed system is built around conventional three-tier architecture. The three-tier architecture for web development allows programmers to separate various aspects of the solution design into modules and work on them separately. That is, a developer who is best at one part of development, say UI development need not worry about the implementation levels so much. It also allows for easy maintenance and future enhancements. The three-tiers of the solution include:

- The Layout**:**

This tier is at the uppermost layer and is closely bound to the user, i.e., the users of the system interact with it through this tier.

- The business-tier**:**

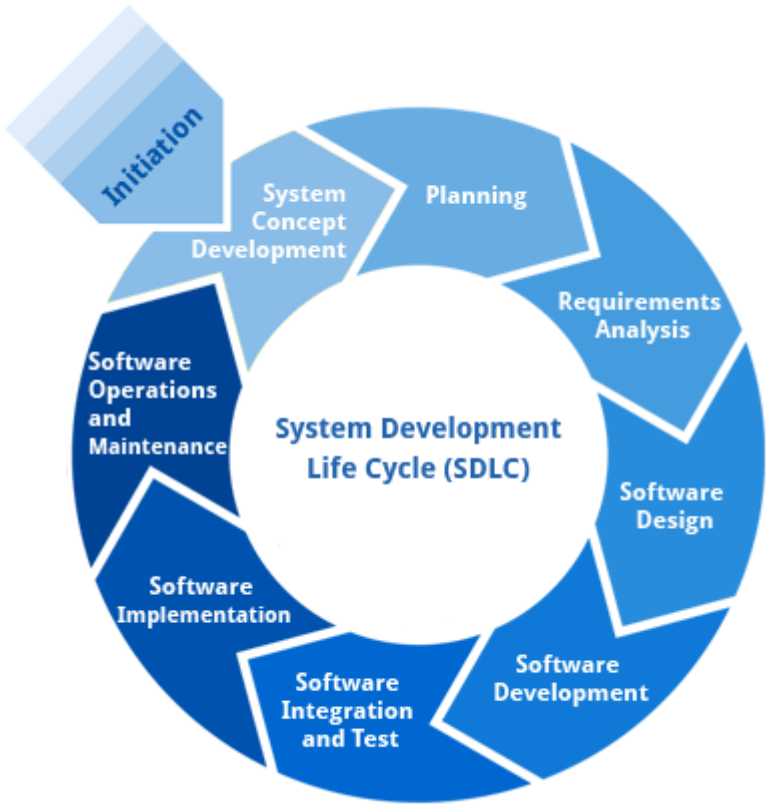
This tier is responsible for implementing all the business rules of the organization. It operates on the data provided by the users through the web-tier and the data stored in the underlying data-tier. So in a way this tier works on data from the web-tier and the data-tier in order to perform task for the users in agreement with the business rules of the organization.

- The data-tier**:**

This tier contains the persist able data that is required by the business tier to operate on. Data plays a very important role in the functioning of any organization. Thus, persisting of such data is very important. The data tier performs the job of persisting the data.

Software Development Life Cycle:

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies use to develop these systems.



Requirement Analysis and Design

Analysis gathers the requirements for the system. This stage includes a detailed study of the business needs of the organization. Options for changing the business process may be considered. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken

during this phase. The logical system of the product is developed in this phase.

Implementation

In this phase the designs are translated into code. Computer programs are written using a conventional programming language or an application generator. Programming tools like Compilers, Interpreters, and Debuggers are used to generate the code. Different high level programming languages like PYTHON 3.6, Anaconda Cloud are used for coding. With respect to the type of application, the right programming language is chosen.

Testing

In this phase the system is tested. Normally programs are written as a series of individual modules, this subject to separate and detailed test. The system is then tested as a whole. The separate modules are brought together and tested as a complete system. The system is tested to ensure that interfaces between modules work (integration testing), the system works on the intended platform and with the expected volume of data (volume testing) and that the system does what the user requires (acceptance/beta testing).

Maintenance

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

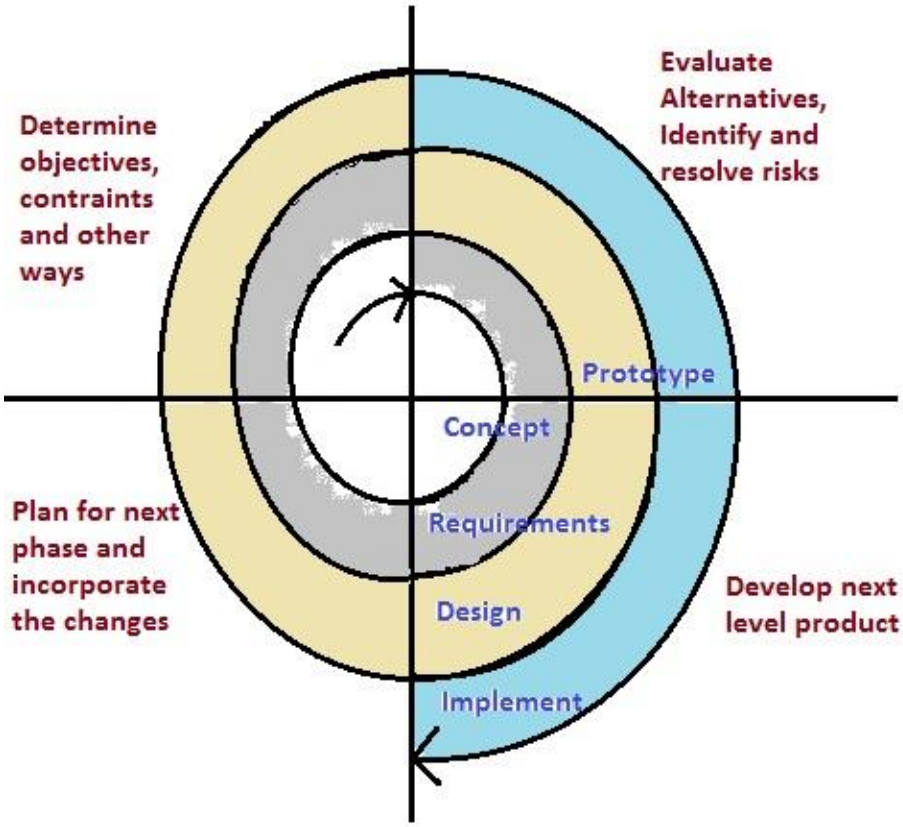
SDLC METHDOLOGIES

This document play a vital role in the development of life cycle (SDLC) as it describes the complete requirement of the system. It means for use by developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

SPIRAL MODEL was defined by Barry Boehm in his 1988 article, "A spiral Model of Software Development and Enhancement. This model was not the first model to discuss iterative development, but it was the first model to explain why the iteration models.

As originally envisioned, the iterations were typically 6 months to 2 years long. Each phase starts with a design goal and ends with a client reviewing the progress thus far. Analysis and engineering efforts are applied at each phase of the project, with an eye toward the end goal of the project.

The following diagram shows how a spiral model acts like:



The steps for Spiral Model can be generalized as follows:

- The new system requirements are defined in as much details as possible.
- This usually involves interviewing a number of users representing all the external or internal users and other aspects of the existing system.
- A preliminary design is created for the new system.
- A first prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.
- A second prototype is evolved by a fourfold procedure:

    1. Evaluating the first prototype in terms of its strengths, weakness, and risks.

    2. Defining the requirements of the second prototype.

3. Planning a designing the second prototype.

4. Constructing and testing the second prototype.

- At the customer option, the entire project can be aborted if the risk is deemed too great. Risk factors might involve development cost overruns, operating-cost miscalculation, or any other factor that could, in the customer's judgment, result in a less-than-satisfactory final product.
- The existing prototype is evaluated in the same manner as was the previous prototype, and if necessary, another prototype is developed from it according to the fourfold procedure outlined above.
- The preceding steps are iterated until the customer is satisfied that the refined prototype represents the final product desired.
- The final system is constructed, based on the refined prototype.
- The final system is thoroughly evaluated and tested. Routine maintenance is carried on a continuing basis to prevent large scale failures and to minimize down time.

Software Environment

What is Python? Executive Summary

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. Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

IMPLEMENTATION TECHNICS:

About python:

The Python programming language is an Open Source, cross-platform, high level, dynamic, interpreted language.

The Python 'philosophy' emphasizes readability, clarity and simplicity, whilst maximizing the power and expressiveness available to the programmer. The ultimate compliment to a Python programmer is not that his code is clever, but that it is elegant. For these reasons Python is an excellent 'first language', while still being a powerful tool in the hands of the seasoned and cynical programmer.

Python is a very flexible language. It is widely used for many different purposes. Typical uses include :

- Web application programming with frameworks like Zope, Django and Turbogears
- System administration tasks via simple scripts
- Desktop applications using GUI toolkits like Tkinter or wxPython (and recently Windows Forms and IronPython)
- Creating windows applications, using the Pywin32 extension for full windows integration and possibly Py2exe to create standalone programs
- Scientific research using packages like Scipy and Matplotlib

Good to know

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being

updated with anything other than security updates, is still quite popular.

- In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages

- Python was designed for readability, and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

jupyter notebook:

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more**.**

upyter Notebooks are a powerful way to write and iterate on your Python code for data analysis. Rather than writing and re-writing an entire program, you can write lines of code and run them one at a time. Then, if you need to make a change, you can go back and make your edit and rerun the program again, all in the same window.

Jupyter Notebook is built off of *IPython*, an interactive way of running Python code in the terminal using the *REPL model* (Read-Eval-Print-Loop). The IPython Kernel runs the computations and communicates with the Jupyter Notebook front-end interface. It also allows Jupyter Notebook to support multiple languages. Jupyter Notebooks extend IPython through additional features, like storing your code and output and allowing you to keep markdown notes.

If you'd rather watch a video instead of read an article, please watch the following instructions on how to use a Jupyter Notebook. They cover the same information.

LAUNCH A NOTEBOOK

To launch a Jupyter notebook, open your terminal and navigate to the directory where you would like to save your notebook. Then type the command jupyter notebook and the program will instantiate a local server at localhost:8888 (or another specified port).

```
[9] -> jupyter notebook
[I 18:31:51.264 NotebookApp] Serving notebooks from, local directory: /Users/janedoe
[I 18:31:51.264 NotebookApp] 0 active kernels
[I 18:31:51.264 NotebookApp] The Jupyter Notebook is running at:
[I 18:31:51.264 NotebookApp] http://localhost:8888/?token=f9b639294933fa68c64c78effb9b65
19f9d8c45c903baa43
```

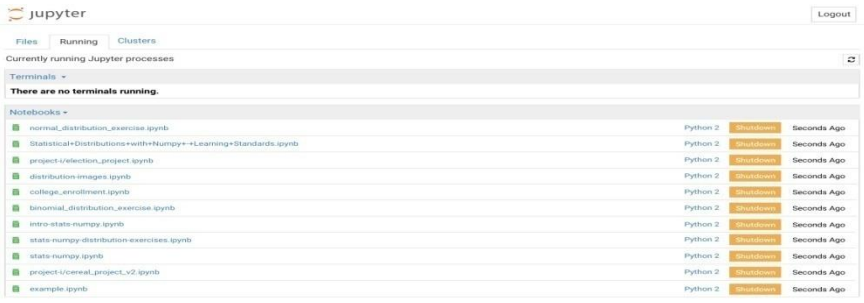
A browser window should immediately pop up with the Jupyter Notebook interface, otherwise, you can use the address it gives you. The notebooks have a unique token since the software uses pre-built Docker containers to put notebooks on their own unique path. To stop the server and shutdown the kernel from the terminal, hit control-C twice.

JUPYTER INTERFACE

Now you're in the Jupyter Notebook interface, and you can see all of the files in your current directory. All Jupyter Notebooks are identifiable by the **notebook icon** next to their name. If you already have a Jupyter Notebook in your current directory that you want to view, find it in your files list and click it to open.

To create a new notebook, go to **New** and select **Notebook - Python 2**. If you have other Jupyter Notebooks on your system that you want to use, you can click **Upload** and navigate to that particular file.

Notebooks currently running will have a green icon, while non-running ones will be grey. To find all currently running notebooks, click on the **Running** tab to see a list.



INSIDE THE NOTEBOOK

When you open a new Jupyter notebook, you'll notice that it contains a *cell*.

Cells are how notebooks are structured and are the areas where you write your code. To run a piece of code, click on the cell to select it, then press SHIFT+ENTER or press the play button in the toolbar above. Additionally, the **Cell** dropdown menu has several options to run cells, including running one cell at a time or to run all cells at once.

SYSTEM DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

SOFTWARE DESIGN

In designing the software following principles are followed:

1. **Modularity and partitioning**: software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.

2. **Coupling:** modules should have little dependence on other modules of a system.

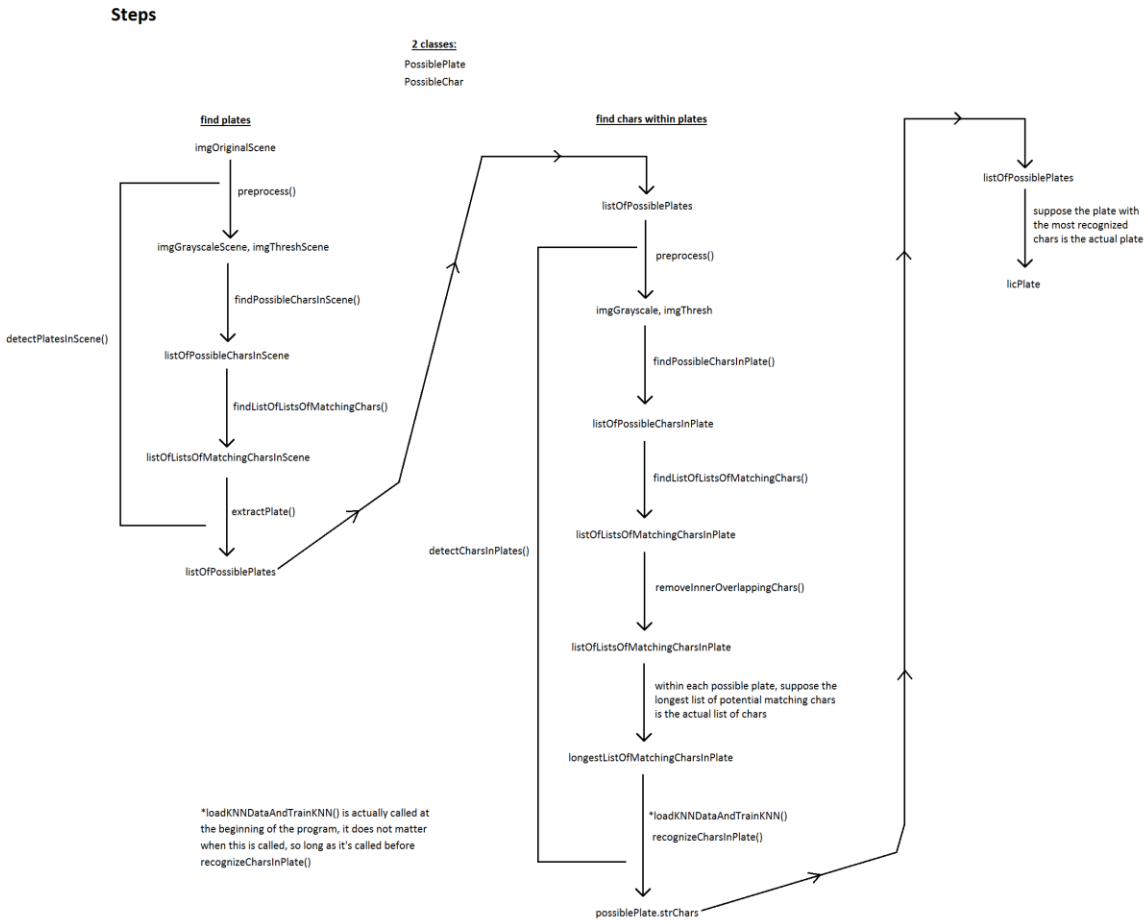
3. **Cohesion:** modules should carry out in a single processing function.

4. **Shared use:** avoid duplication by allowing a single module be called by other that need the function it provide

DATA FLOW DIAGRAM:

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional details

Flow Chart:


Steps
2 classes:
PossiblePlate
PossibleChar
find plates
imgOriginalScene
preprocess()
imgGrayscaleScene, imgThreshScene
findPossibleCharsInScene()
detectPlatesInScene()
listOfPossibleCharsInScene
findListOfListsOfMatchingChars()
listOfListsOfMatchingCharsInScene
extractPlate()
listOfPossiblePlates
find chars within plates
listOfPossiblePlates
preprocess()
imgGrayscale, imgThresh
findPossibleCharsInPlate()
listOfPossibleCharsInPlate
findListOfListsOfMatchingChars()
detectCharsInPlates()
listOfListsOfMatchingCharsInPlate
removeInnerOverlappingChars()
listOfListsOfMatchingCharsInPlate
within each possible plate, suppose the longest list of potential matching chars is the actual list of chars
longestListOfMatchingCharsInPlate
*loadKNNDataAndTrainKNN()
recognizeCharsInPlate()
possiblePlate.strChars
*loadKNNDataAndTrainKNN() is actually called at the beginning of the program, it does not matter when this is called, so long as it's called before recognizeCharsInPlate()
listOfPossiblePlates
suppose the plate with the most recognized chars is the actual plate
licPlate

Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

Data flow diagrams can be divided into logical and physical. The logical data flow diagram describes flow of data through a system to perform certain functionality of a business. The physical data flow diagram describes the implementation of the logical data flow.

UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

> The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS:

The Primary goals in the design of the UML are as follows:

Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.

Provide extendibility and specialization mechanisms to extend the core concepts.

Be independent of particular programming languages and development process.

Provide a formal basis for understanding the modeling language.

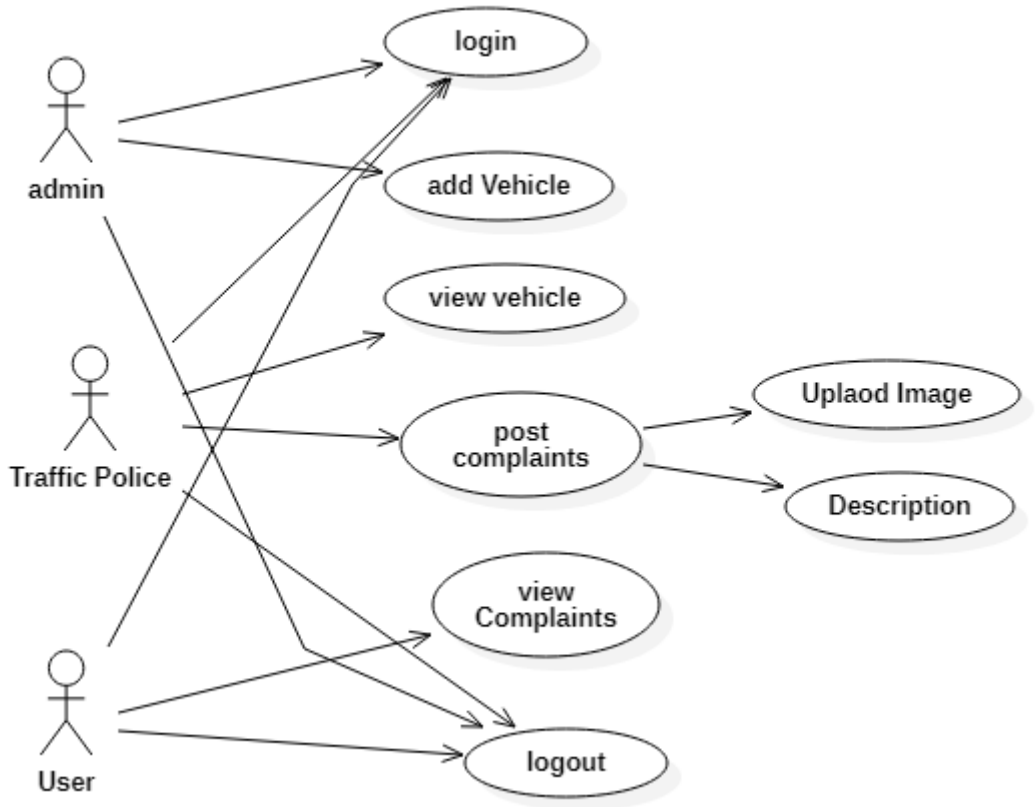
Encourage the growth of OO tools market.

Support higher level development concepts such as collaborations, frameworks, patterns and components.

Integrate best practices.

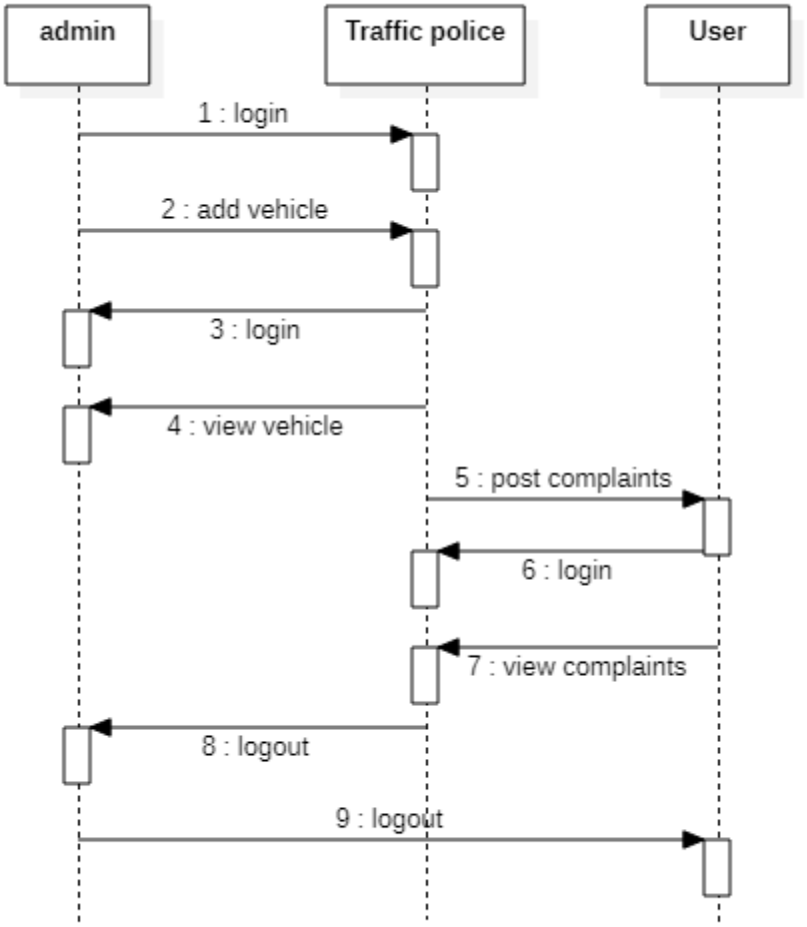
Use Case Diagram:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



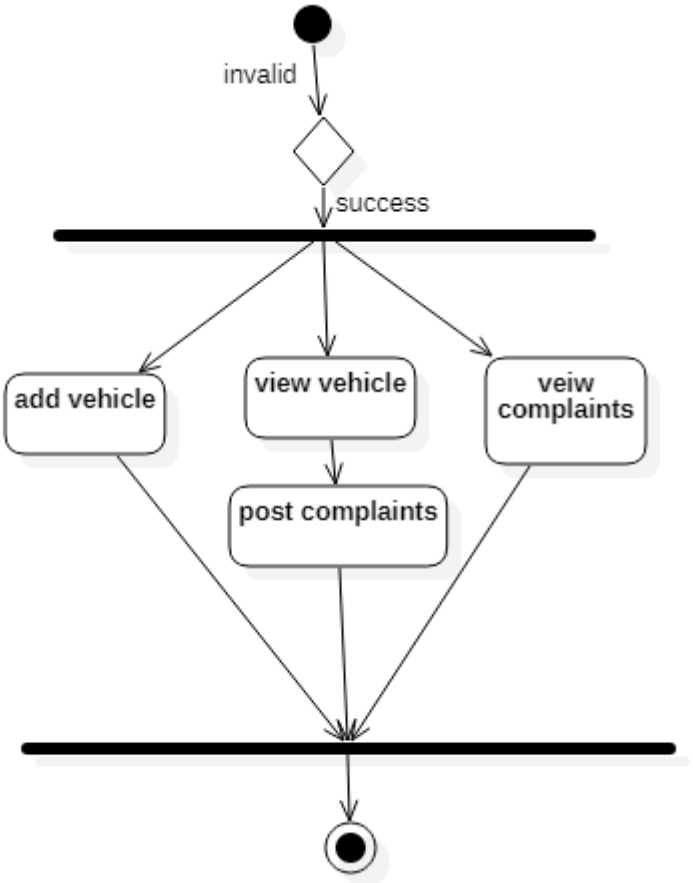
Sequence diagram:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment.


admin
Traffic police
User
1 : login
2 : add vehicle
3 : login
4 : view vehicle
5 : post complaints
6 : login
7 : view complaints
8 : logout
9 : logout

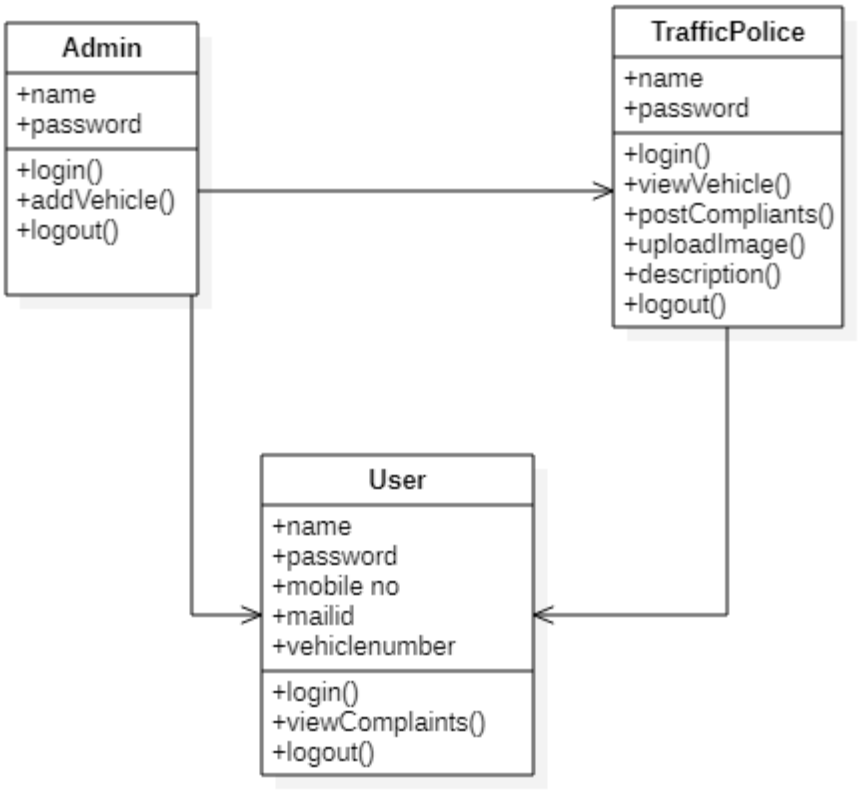
Activity diagram:

•Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



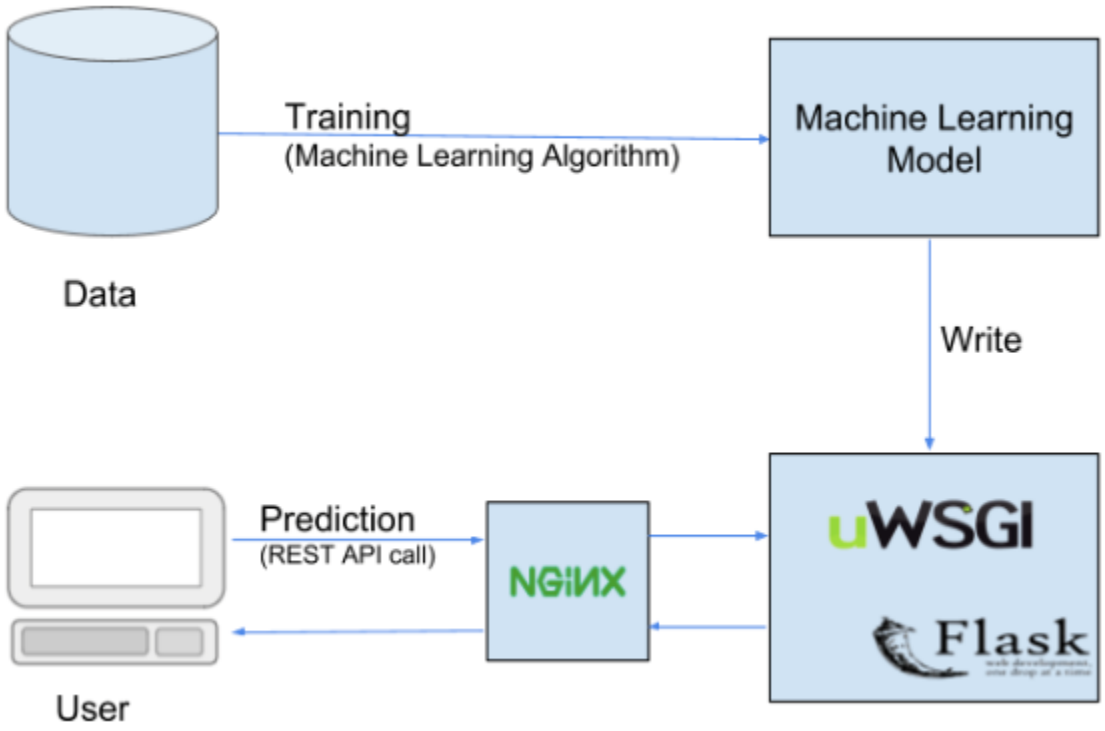
CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



Deployment diagram

There may be more steps involved, depending on what specific requirements you have, but below are some of the main steps:



Packaging your ML model

Securing your packaged ML model

Planning of how to serve/expose your ML model to the consumers (as a REST service, etc.)

If you have planned for a REST service, then, creating a REST API for your ML model

Securing your REST API

Deploying your REST service in production (using Docker and Kubernetes)

Packaging your ML model

Instead of saving the ML model as it is, you can package your ML model (say mnist), and create a .pkl file for it, using Python's joblib library.

For creating and restoring the .pkl file, you can either use joblib library or pickle library of Python. You normally use joblib to save an object having large data, else, you use pickle library. Here, in this case, we have used joblib library.

.pkl file is nothing but a serialized pickle file, which if you want, you can compress it further, to save storage space, using say Python's gzip library. After you apply compression, your ML model file name will look like – mnist.pkl.gz

Securing your packaged ML model

The content of your pickle file (in this case your ML model) could be intercepted and modified by anyone over the network if your network is not secured. Hence, it is advisable to use secured (encrypted) network connection while exchanging the pickle file.

In addition to this, the pickle file could be signed (using 'cryptographic signature') before storing or transmitting, and this signature can be verified before it is restored at the receiver's end (say your REST API). Cryptographic signature helps in detecting any alterations to your pickle file data.

Cryptographic signature uses a cryptographic algorithm which generates a cryptographic hash of your pickle file data along with shared secret key. SHA-1 cryptographic algorithm is considered to be the best algorithm to create a stronger hash, hence, it is highly advisable to use it.

Sample Code

```
# DetectPlates.py


import cv2

import numpy as np

import math

import Main

import randomimport Preprocess

import DetectChars

import PossiblePlate

import PossibleChar


# module level variables ##########################################################################

PLATE_WIDTH_PADDING_FACTOR = 1.3

PLATE_HEIGHT_PADDING_FACTOR = 1.5


##################################################################################################

def detectPlatesInScene(imgOriginalScene):
```

```
    listOfPossiblePlates = []                # this will be the return value

    height, width, numChannels = imgOriginalScene.shape

    imgGrayscaleScene = np.zeros((height, width, 1), np.uint8)
    imgThreshScene = np.zeros((height, width, 1), np.uint8)
    imgContours = np.zeros((height, width, 3), np.uint8)

    cv2.destroyAllWindows()

    if Main.showSteps == True: # show steps ###################################################################

        cv2.imshow("0", imgOriginalScene)

    # end if # show steps #####################################################################

    imgGrayscaleScene, imgThreshScene = Preprocess.preprocess(imgOriginalScene)          # preprocess to get grayscale and threshold images
```

```
    if Main.showSteps == True: # show steps
##################################################
###########

        cv2.imshow("1a", imgGrayscaleScene)

        cv2.imshow("1b", imgThreshScene)

    # end if # show steps
#####################################################
#############################

            # find all possible chars in the scene,

            # this function first finds all contours, then only includes
contours that could be chars (without comparison to other chars yet)

    listOfPossibleCharsInScene =
findPossibleCharsInScene(imgThreshScene)

    if Main.showSteps == True: # show steps
##################################################
###########

        print("step 2 - len(listOfPossibleCharsInScene) = " + str(

            len(listOfPossibleCharsInScene)))  # 131 with MCLRNF1 image

        imgContours = np.zeros((height, width, 3), np.uint8)
```

```
        contours = []


        for possibleChar in listOfPossibleCharsInScene:
            contours.append(possibleChar.contour)
        # end for


        cv2.drawContours(imgContours, contours, -1, Main.SCALAR_WHITE)
        cv2.imshow("2b", imgContours)
    # end if # show steps #####################################################################


        # given a list of all possible chars, find groups of matching chars
        # in the next steps each group of matching chars will attempt to be recognized as a plate
    listOfListsOfMatchingCharsInScene = DetectChars.findListOfListsOfMatchingChars(listOfPossibleCharsInScene)
```

```
    if Main.showSteps == True: # show steps
##########################################################
###########
        print("step 3 - listOfListsOfMatchingCharsInScene.Count = " +
str(
            len(listOfListsOfMatchingCharsInScene)))  # 13 with MCLRNF1
image


        imgContours = np.zeros((height, width, 3), np.uint8)


        for listOfMatchingChars in listOfListsOfMatchingCharsInScene:
            intRandomBlue = random.randint(0, 255)
            intRandomGreen = random.randint(0, 255)
            intRandomRed = random.randint(0, 255)


            contours = []


            for matchingChar in listOfMatchingChars:
                contours.append(matchingChar.contour)
            # end for
```

```
            cv2.drawContours(imgContours, contours, -1, (intRandomBlue, intRandomGreen, intRandomRed))

        # end for

        cv2.imshow("3", imgContours)

    # end if # show steps #########################################################################

    for listOfMatchingChars in listOfListsOfMatchingCharsInScene: # for each group of matching chars

        possiblePlate = extractPlate(imgOriginalScene, listOfMatchingChars)         # attempt to extract plate

        if possiblePlate.imgPlate is not None:                          # if plate was found

            listOfPossiblePlates.append(possiblePlate)                  # add to list of possible plates

        # end if

    # end for

    print("\n" + str(len(listOfPossiblePlates)) + " possible plates found") # 13 with MCLRNF1 image
```

```
    if Main.showSteps == True: # show steps
##########################################
###########

        print("\n")

        cv2.imshow("4a", imgContours)



        for i in range(0, len(listOfPossiblePlates)):

            p2fRectPoints =
cv2.boxPoints(listOfPossiblePlates[i].rrLocationOfPlateInScene)



            cv2.line(imgContours, tuple(p2fRectPoints[0]),
tuple(p2fRectPoints[1]), Main.SCALAR_RED, 2)

            cv2.line(imgContours, tuple(p2fRectPoints[1]),
tuple(p2fRectPoints[2]), Main.SCALAR_RED, 2)

            cv2.line(imgContours, tuple(p2fRectPoints[2]),
tuple(p2fRectPoints[3]), Main.SCALAR_RED, 2)

            cv2.line(imgContours, tuple(p2fRectPoints[3]),
tuple(p2fRectPoints[0]), Main.SCALAR_RED, 2)



            cv2.imshow("4a", imgContours)
```

```
            print("possible plate " + str(i) + ", click on any image and press a key to continue . . .")

            cv2.imshow("4b", listOfPossiblePlates[i].imgPlate)

            cv2.waitKey(0)

        # end for

        print("\nplate detection complete, click on any image and press a key to begin char recognition . . .\n")

        cv2.waitKey(0)

    # end if # show steps #########################################################################

    return listOfPossiblePlates

# end function

#########################################################################################

def findPossibleCharsInScene(imgThresh):

    listOfPossibleChars = []                # this will be the return value
```

```
    intCountOfPossibleChars = 0

    imgThreshCopy = imgThresh.copy()

    contours, npaHierarchy = cv2.findContours(imgThreshCopy, cv2.RETR_LIST, cv2.CHAIN_APPROX_SIMPLE)  # find all contours

    height, width = imgThresh.shape
    imgContours = np.zeros((height, width, 3), np.uint8)

    for i in range(0, len(contours)):                   # for each contour

        if Main.showSteps == True: # show steps ###################################################
            cv2.drawContours(imgContours, contours, i, Main.SCALAR_WHITE)
        # end if # show steps #####################################################################
```

```
        possibleChar = PossibleChar.PossibleChar(contours[i])

        if DetectChars.checkIfPossibleChar(possibleChar):                # if contour is a possible char, note this does not compare to other chars (yet) . . .
            intCountOfPossibleChars = intCountOfPossibleChars + 1 # increment count of possible chars
            listOfPossibleChars.append(possibleChar)                # and add to list of possible chars
        # end if
    # end for

    if Main.showSteps == True: # show steps #########################################################
        print("\nstep 2 - len(contours) = " + str(len(contours)))  # 2362 with MCLRNF1 image
        print("step 2 - intCountOfPossibleChars = " + str(intCountOfPossibleChars))  # 131 with MCLRNF1 image
        cv2.imshow("2a", imgContours)
    # end if # show steps #########################################################
```

```
    return listOfPossibleChars

# end function

##################################################################################################################

def extractPlate(imgOriginal, listOfMatchingChars):

    possiblePlate = PossiblePlate.PossiblePlate()        # this will be the return value


    listOfMatchingChars.sort(key = lambda matchingChar: matchingChar.intCenterX)        # sort chars from left to right based on x position


            # calculate the center point of the plate

    fltPlateCenterX = (listOfMatchingChars[0].intCenterX + listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterX) / 2.0

    fltPlateCenterY = (listOfMatchingChars[0].intCenterY + listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterY) / 2.0


    ptPlateCenter = fltPlateCenterX, fltPlateCenterY
```

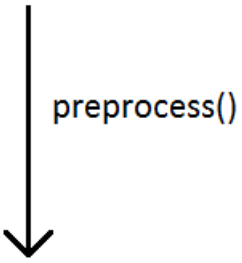
```
        # calculate plate width and height

    intPlateWidth = int((listOfMatchingChars[len(listOfMatchingChars) - 1].intBoundingRectX + listOfMatchingChars[len(listOfMatchingChars) - 1].intBoundingRectWidth - listOfMatchingChars[0].intBoundingRectX) * PLATE_WIDTH_PADDING_FACTOR)

    intTotalOfCharHeights = 0

    for matchingChar in listOfMatchingChars:
        intTotalOfCharHeights = intTotalOfCharHeights + matchingChar.intBoundingRectHeight
    # end for

    fltAverageCharHeight = intTotalOfCharHeights / len(listOfMatchingChars)

    intPlateHeight = int(fltAverageCharHeight * PLATE_HEIGHT_PADDING_FACTOR)

        # calculate correction angle of plate region

    fltOpposite = listOfMatchingChars[len(listOfMatchingChars) - 1].intCenterY - listOfMatchingChars[0].intCenterY
```

```
    fltHypotenuse = DetectChars.distanceBetweenChars(listOfMatchingChars[0], listOfMatchingChars[len(listOfMatchingChars) - 1])

    fltCorrectionAngleInRad = math.asin(fltOpposite / fltHypotenuse)

    fltCorrectionAngleInDeg = fltCorrectionAngleInRad * (180.0 / math.pi)

            # pack plate region center point, width and height, and correction angle into rotated rect member variable of plate

    possiblePlate.rrLocationOfPlateInScene = ( tuple(ptPlateCenter), (intPlateWidth, intPlateHeight), fltCorrectionAngleInDeg )

            # final steps are to perform the actual rotation

            # get the rotation matrix for our calculated correction angle

    rotationMatrix = cv2.getRotationMatrix2D(tuple(ptPlateCenter), fltCorrectionAngleInDeg, 1.0)

    height, width, numChannels = imgOriginal.shape          # unpack original image width and height

    imgRotated = cv2.warpAffine(imgOriginal, rotationMatrix, (width, height))       # rotate the entire image
```

```
    imgCropped = cv2.getRectSubPix(imgRotated, (intPlateWidth, intPlateHeight), tuple(ptPlateCenter))

    possiblePlate.imgPlate = imgCropped         # copy the cropped plate image into the applicable member variable of the possible plate

    return possiblePlate

# end function
```

Screen Shots

imgOriginalScene



imgGrayscaleScene, imgThreshScene

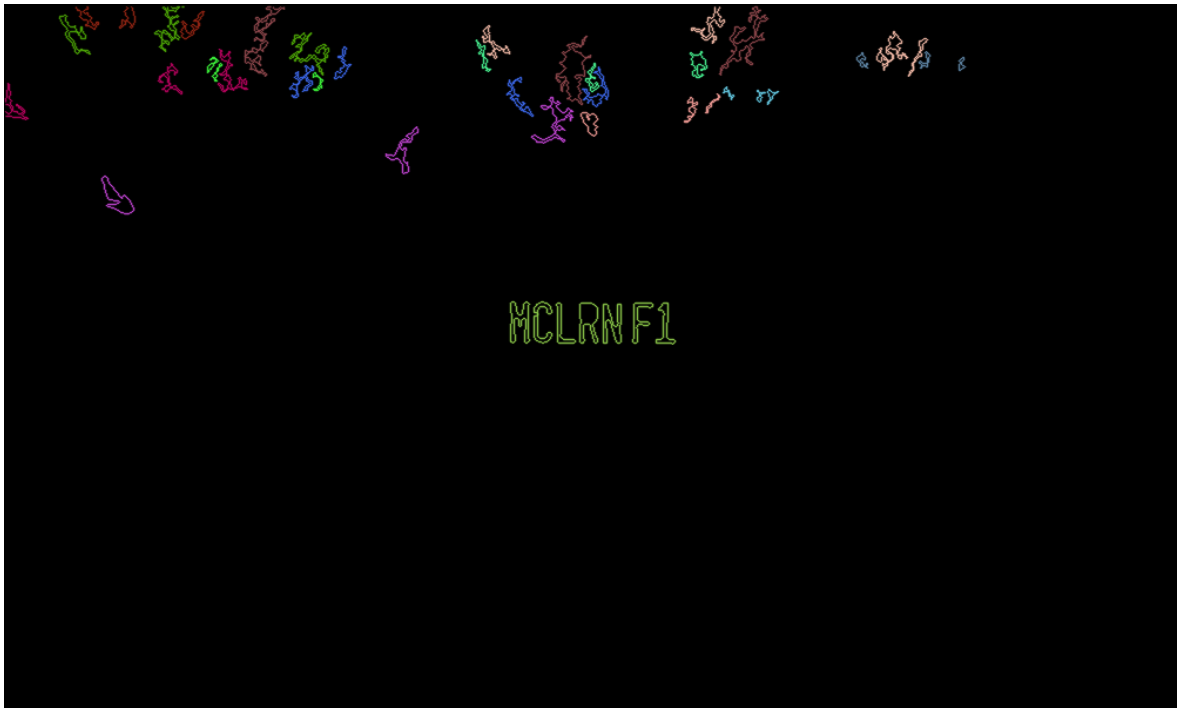
findPossibleCharsInScene()

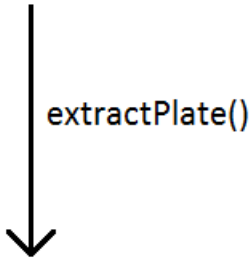
all contours (2362 w/MCLRN F1 image)

vectorOfPossibleCharsInScene (131 w/MCLRN F1 image)

findVectorOfVectorsOfMatchingChars()

vectorOfVectorsOfMatchingCharsInScene (13 w/MCLRN F1 image)





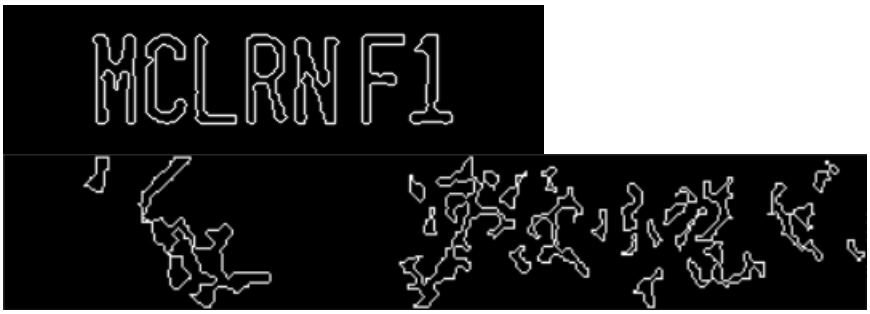
vectorOfPossiblePlates (13 w/MCLRN F1 image)

preprocess()

imgGrayscale, imgThresh

findPossibleCharsInPlate()

vectorOfPossibleCharsInPlate



findVectorOfVectorsOfMatchingChars()

vectorOfVectorsOfMatchingCharsInPlate

removeInnerOverlappingChars()

vectorOfVectorsOfMatchingCharsInPlate

within each possible plate, suppose the longest list of potential matching chars is the actual list of chars

longestVectorOfMatchingCharsInPlate

recognizeCharsInPlate()

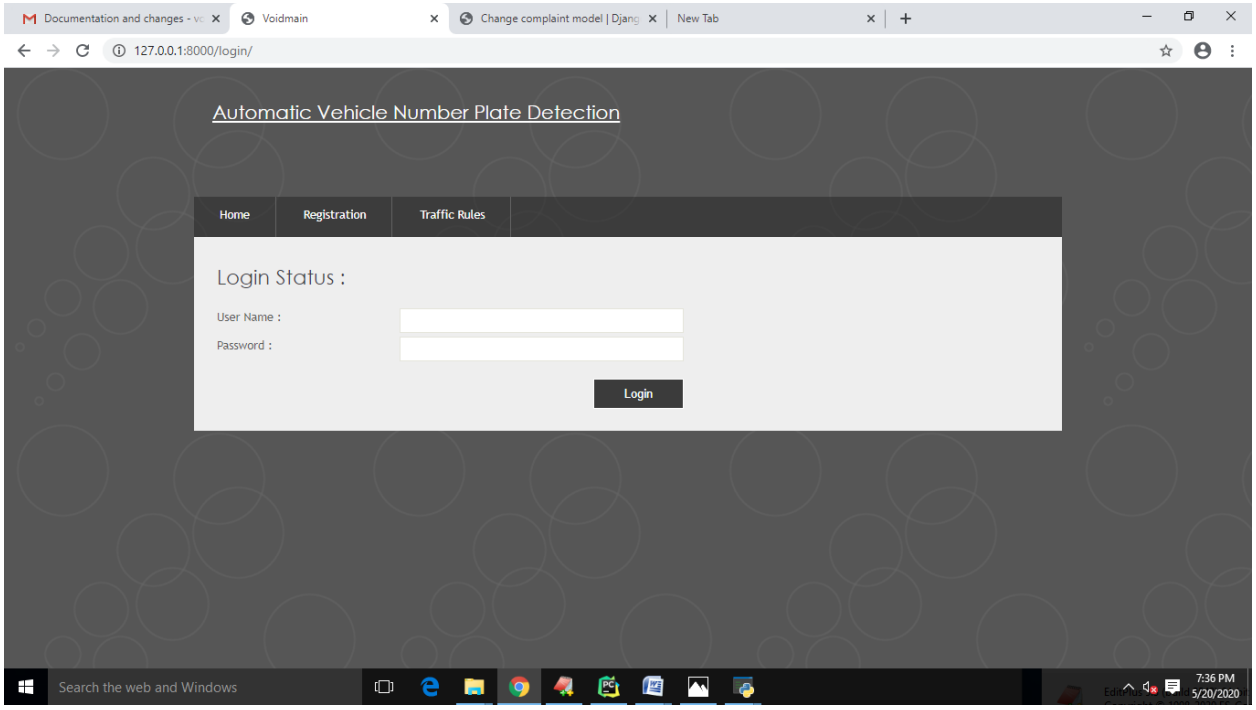

```
chars found in plate number 0 = MCLRNF1,
chars found in plate number 5 = I1I,
```

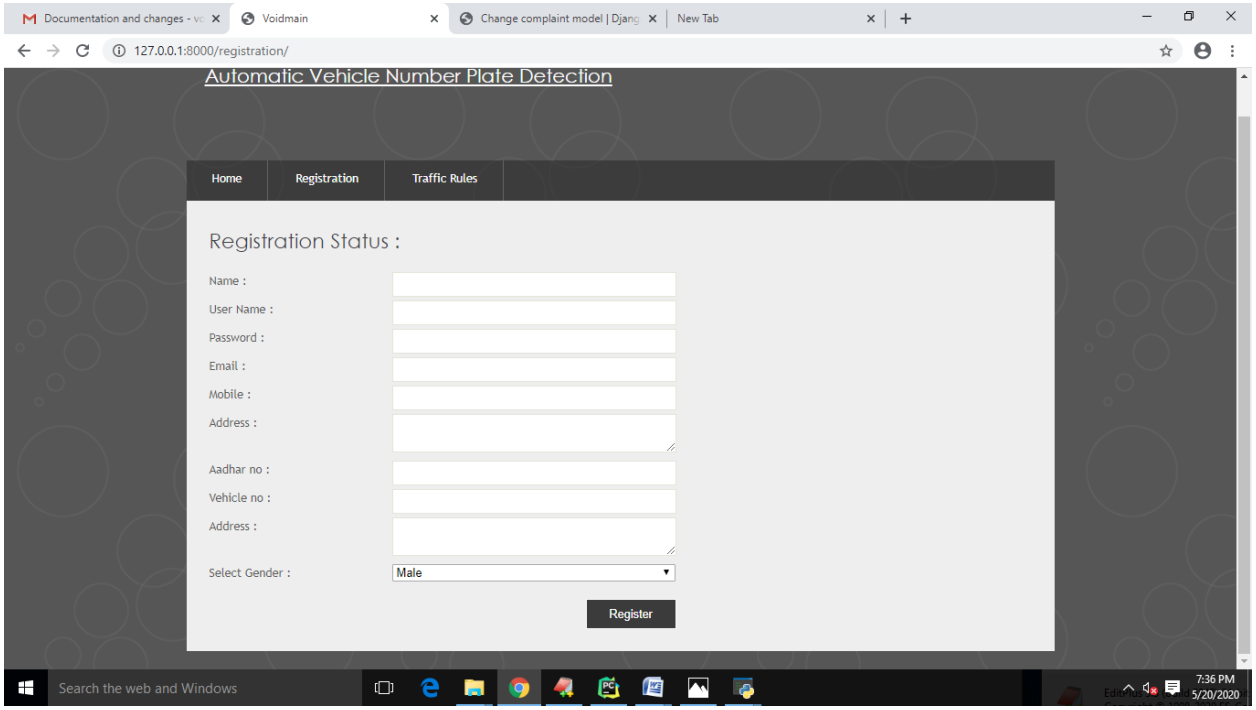
possiblePlate.strChars

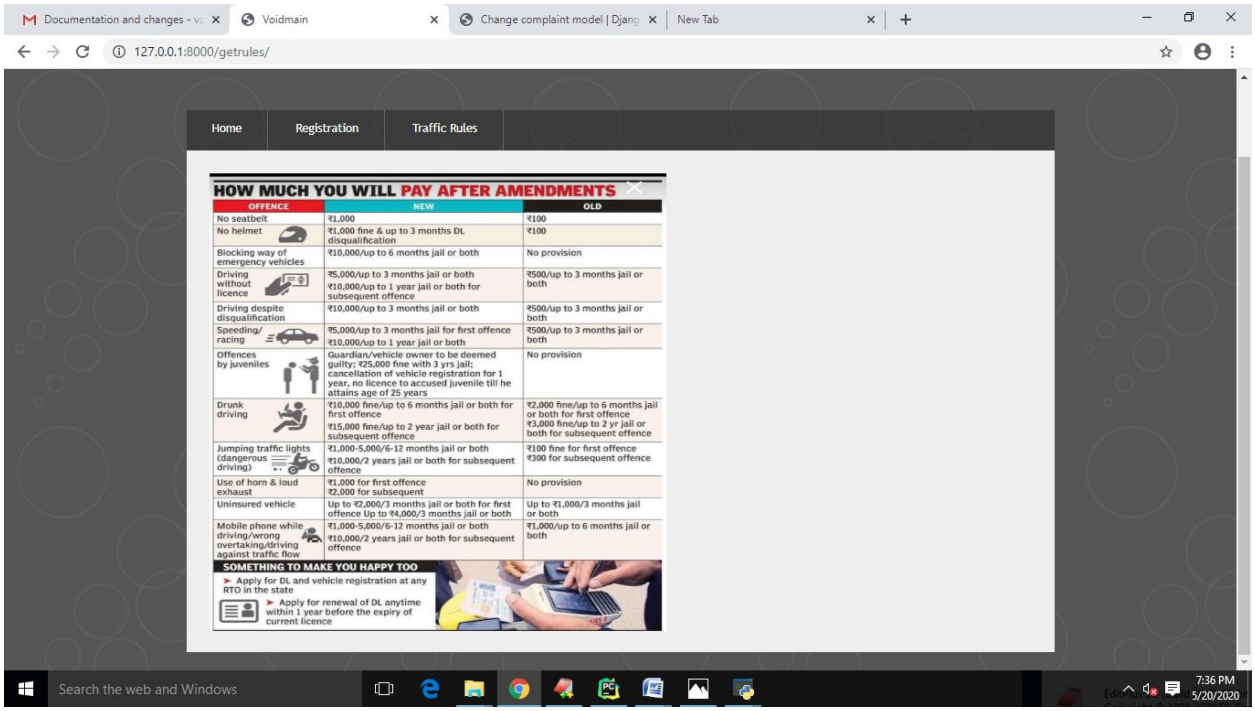
suppose the plate with
the most recognized
chars is the actual plate

Run Main

```
C:\Python27\python.exe C:/Users/cdahms/Doc

13 possible plates found

license plate read from image = MCLRNF1

-----------------------------------------
```

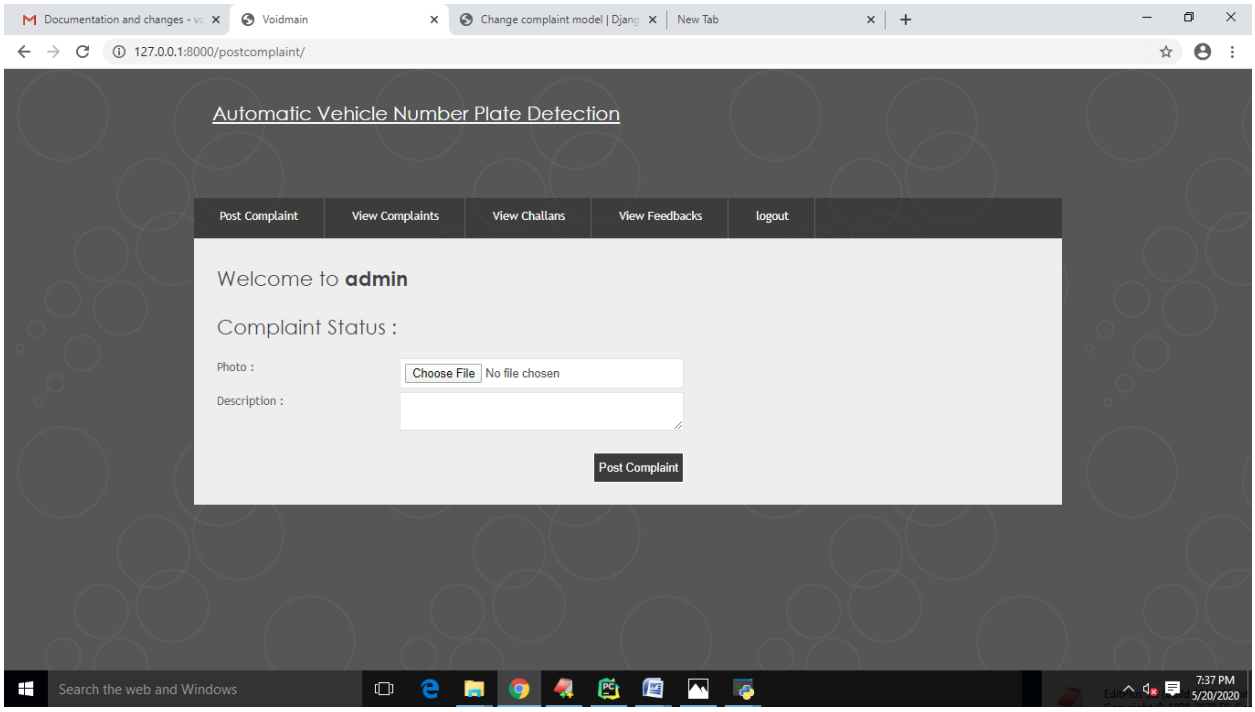

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/login/
Automatic Vehicle Number Plate Detection
Home
Registration
Traffic Rules
Login Status :
User Name :
Password :
Login
Search the web and Windows
7:36 PM
5/20/2020

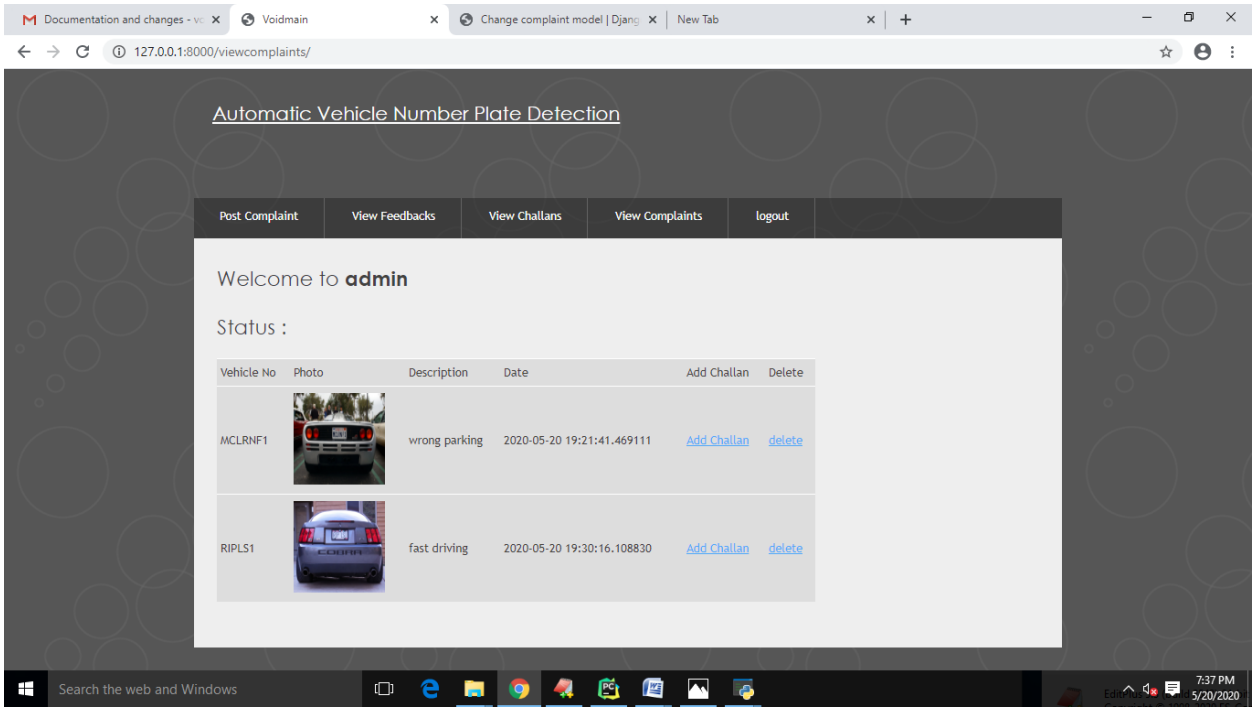

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/registration/
Automatic Vehicle Number Plate Detection
Home
Registration
Traffic Rules
Registration Status :
Name :
User Name :
Password :
Email :
Mobile :
Address :
Aadhar no :
Vehicle no :
Address :
Select Gender :
Male
Register
Search the web and Windows
7:36 PM
5/20/2020

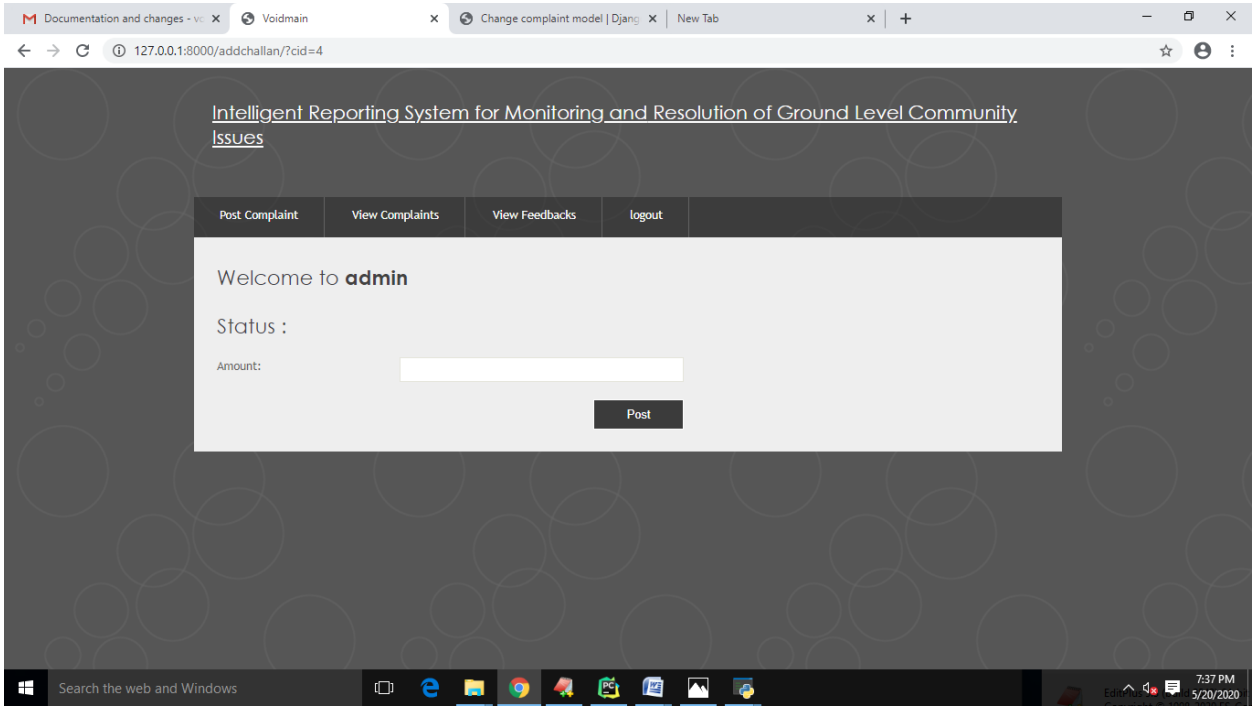

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/getrules/
Home
Registration
Traffic Rules
HOW MUCH YOU WILL PAY AFTER AMENDMENTS
OFFENCE
NEW
OLD
No seatbelt
₹1,000
₹100
No helmet
₹1,000 fine & up to 3 months DL disqualification
₹100
Blocking way of emergency vehicles
₹10,000/up to 6 months jail or both
No provision
Driving without licence
₹5,000/up to 3 months jail or both
₹10,000/up to 1 year jail or both for subsequent offence
₹500/up to 3 months jail or both
Driving despite disqualification
₹10,000/up to 3 months jail or both
₹500/up to 3 months jail or both
Speeding/ racing
₹5,000/up to 3 months jail for first offence
₹10,000/up to 1 year jail or both
₹500/up to 3 months jail or both
Offences by juveniles
Guardian/vehicle owner to be deemed guilty; ₹25,000 fine with 3 yrs jail; cancellation of vehicle registration for 1 year, no licence to accused juvenile till he attains age of 25 years
No provision
Drunk driving
₹10,000 fine/up to 6 months jail or both for first offence
₹15,000 fine/up to 2 year jail or both for subsequent offence
₹2,000 fine/up to 6 months jail or both for first offence
₹3,000 fine/up to 2 yr jail or both for subsequent offence
Jumping traffic lights (dangerous driving)
₹1,000-5,000/6-12 months jail or both
₹10,000/2 years jail or both for subsequent offence
₹100 fine for first offence
₹300 for subsequent offence
Use of horn & loud exhaust
₹1,000 for first offence
₹2,000 for subsequent
No provision
Uninsured vehicle
Up to ₹2,000/3 months jail or both for first offence Up to ₹4,000/3 months jail or both
Up to ₹1,000/3 months jail or both
Mobile phone while driving/wrong overtaking/driving against traffic flow
₹1,000-5,000/6-12 months jail or both
₹10,000/2 years jail or both for subsequent offence
₹1,000/up to 6 months jail or both
SOMETHING TO MAKE YOU HAPPY TOO
Apply for DL and vehicle registration at any RTO in the state
Apply for renewal of DL anytime within 1 year before the expiry of current licence
Search the web and Windows
7:36 PM
5/20/2020

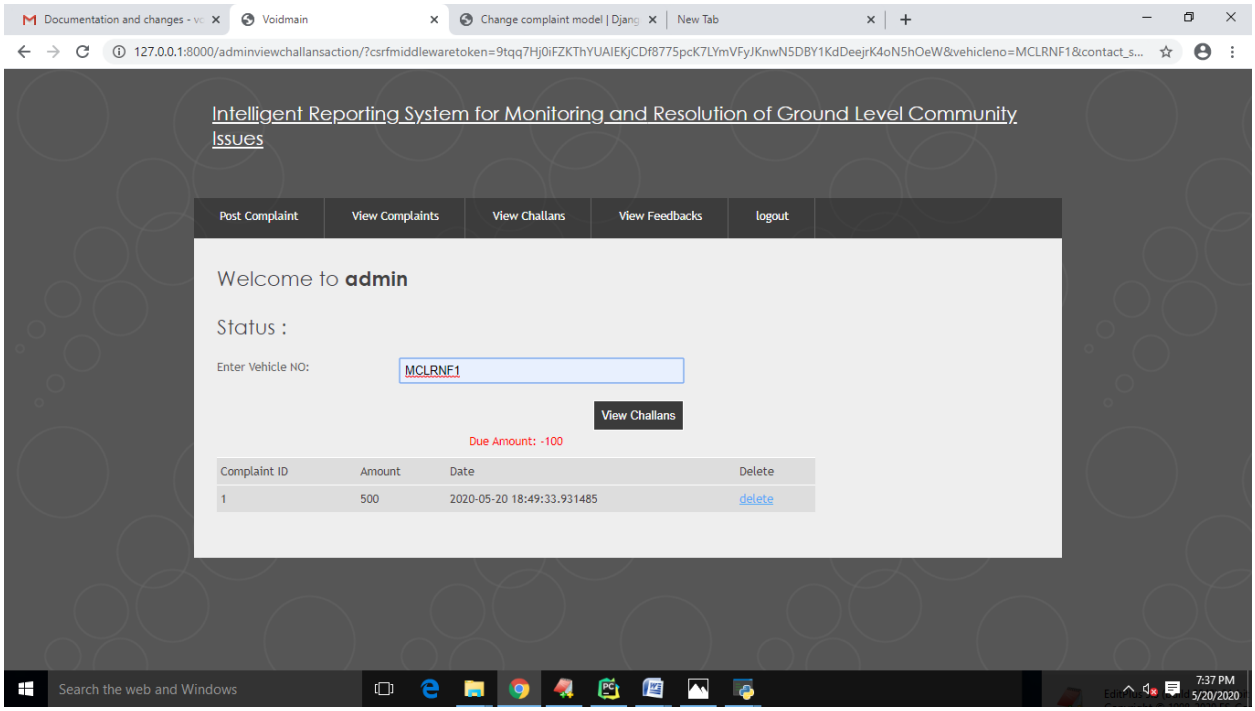
//==========

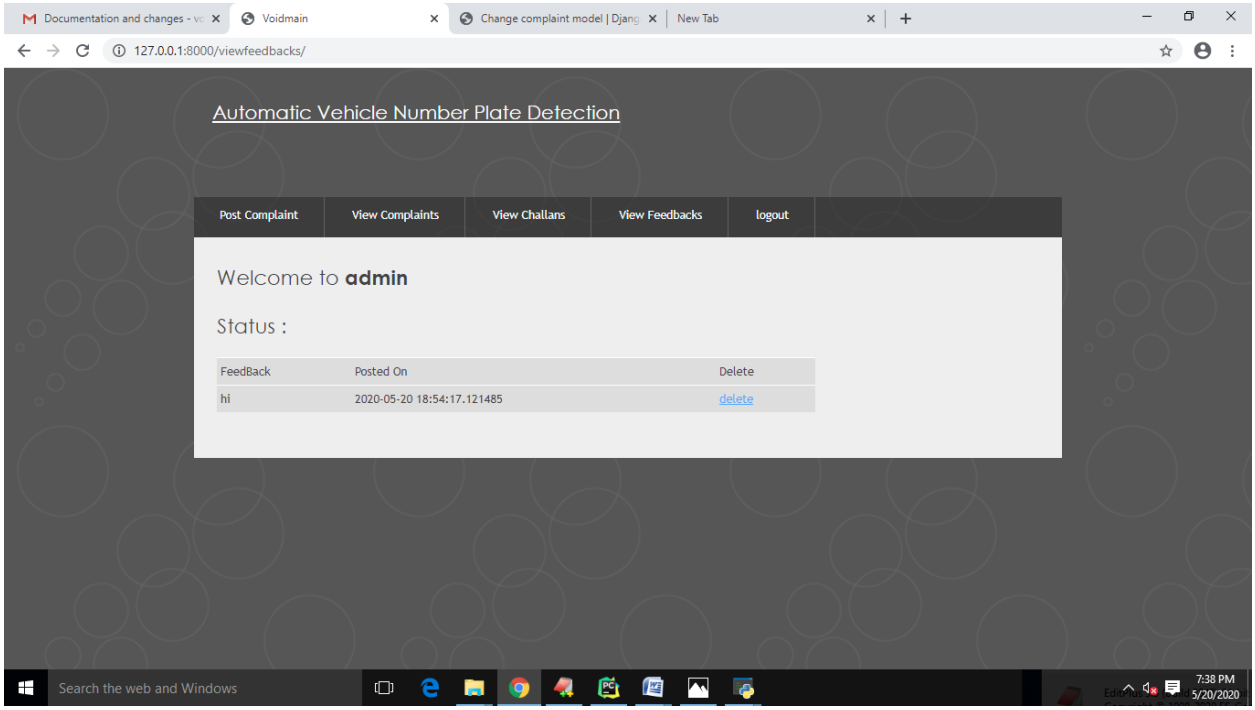
==


Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/postcomplaint/
Automatic Vehicle Number Plate Detection
Post Complaint
View Complaints
View Challans
View Feedbacks
logout
Welcome to admin
Complaint Status :
Photo :
Choose File
No file chosen
Description :
Post Complaint
Search the web and Windows
7:37 PM
5/20/2020

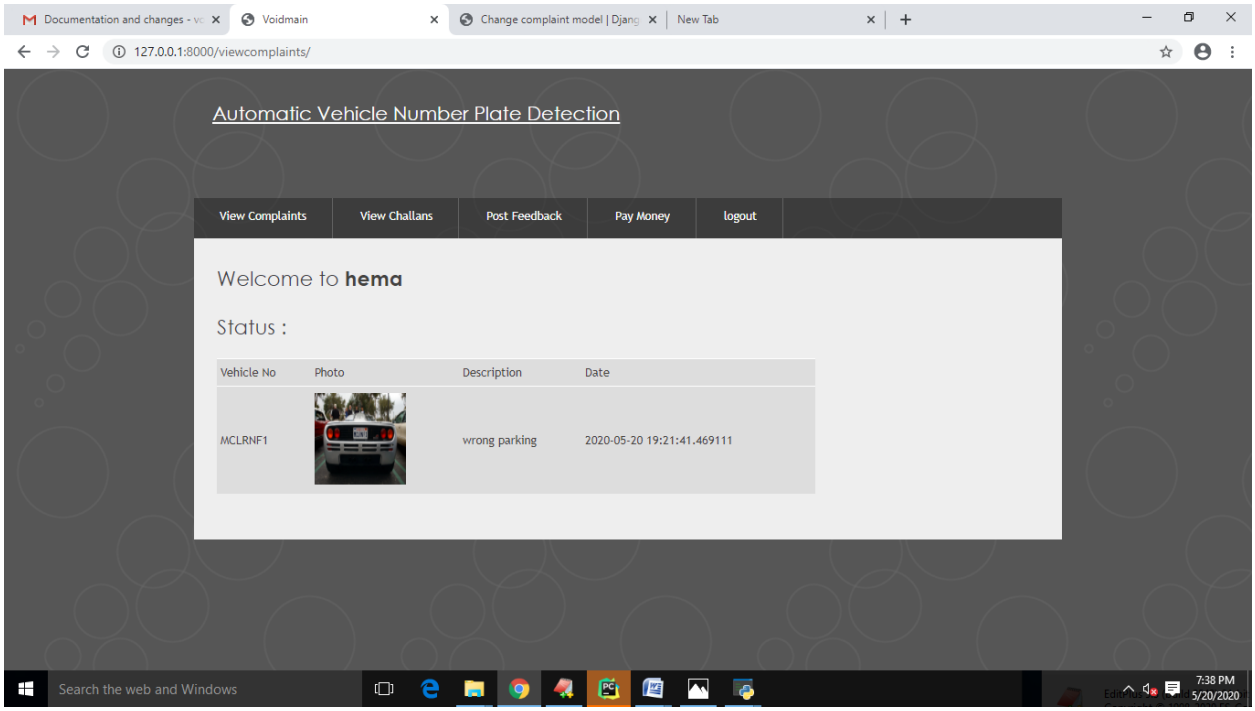

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/viewcomplaints/
Automatic Vehicle Number Plate Detection
Post Complaint
View Feedbacks
View Challans
View Complaints
logout
Welcome to admin
Status :
Vehicle No
Photo
Description
Date
Add Challan
Delete
MCLRNF1
wrong parking
2020-05-20 19:21:41.469111
Add Challan
delete
RIPLS1
fast driving
2020-05-20 19:30:16.108830
Add Challan
delete
Search the web and Windows
7:37 PM
5/20/2020

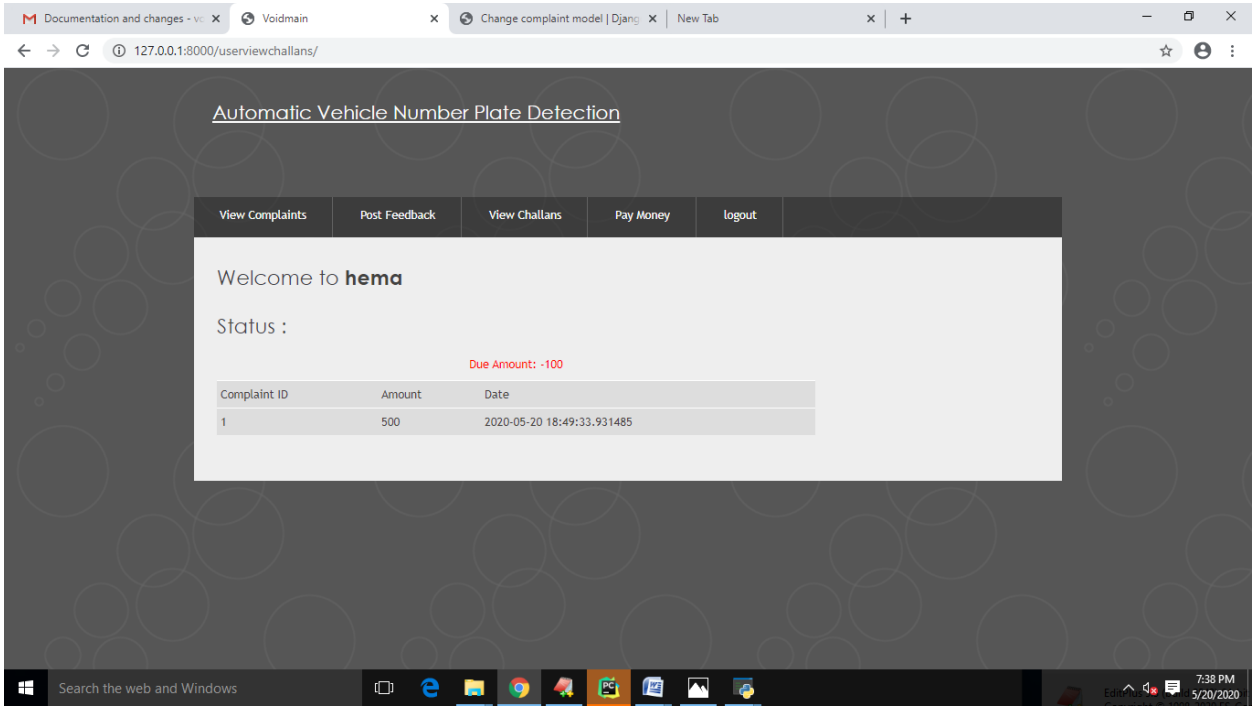

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/addchallan/?cid=4
Intelligent Reporting System for Monitoring and Resolution of Ground Level Community Issues
Post Complaint
View Complaints
View Feedbacks
logout
Welcome to admin
Status :
Amount:
Post
Search the web and Windows
7:37 PM
5/20/2020

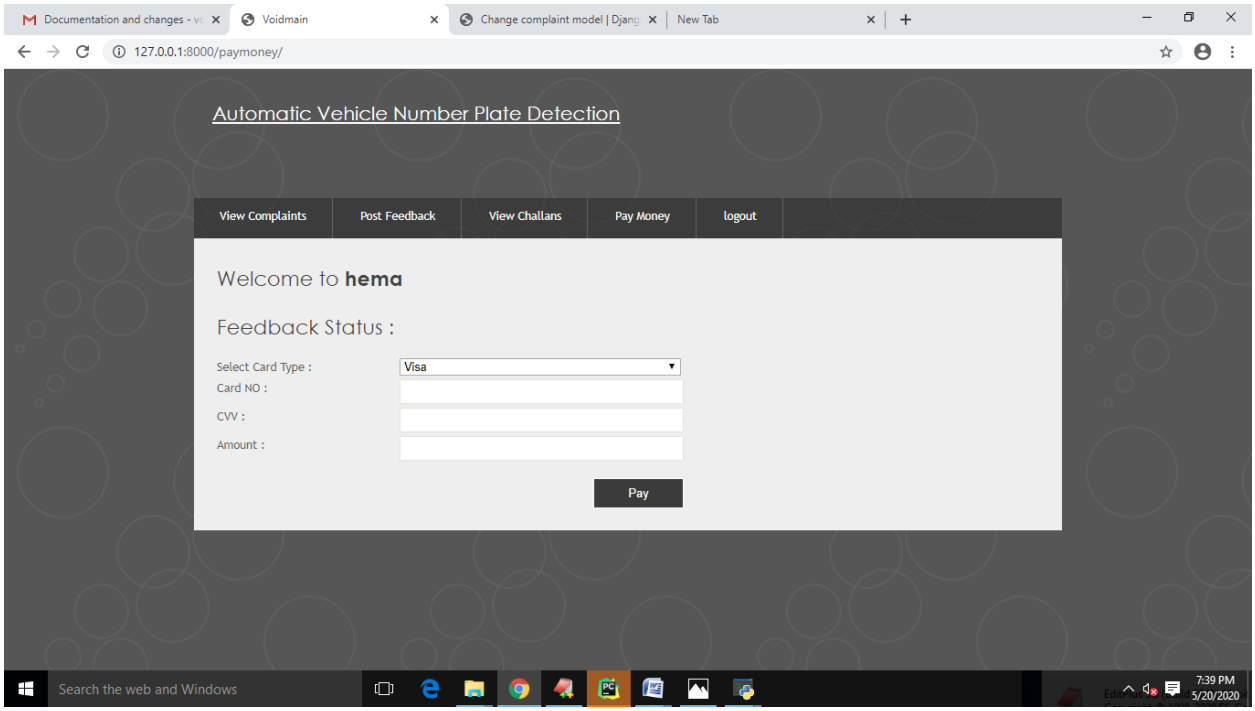

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/adminviewchallansaction/?csrfmiddlewaretoken=9tqq7Hj0iFZKThYUAIEKjCDf8775pcK7LYmVFyJKnwN5DBY1KdDeejrK4oN5hOeW&vehicleno=MCLRNF1&contact_s...
Intelligent Reporting System for Monitoring and Resolution of Ground Level Community Issues
Post Complaint
View Complaints
View Challans
View Feedbacks
logout
Welcome to admin
Status :
Enter Vehicle NO:
MCLRNF1
View Challans
Due Amount: -100
Complaint ID
Amount
Date
Delete
1
500
2020-05-20 18:49:33.931485
delete
Search the web and Windows
7:37 PM
5/20/2020

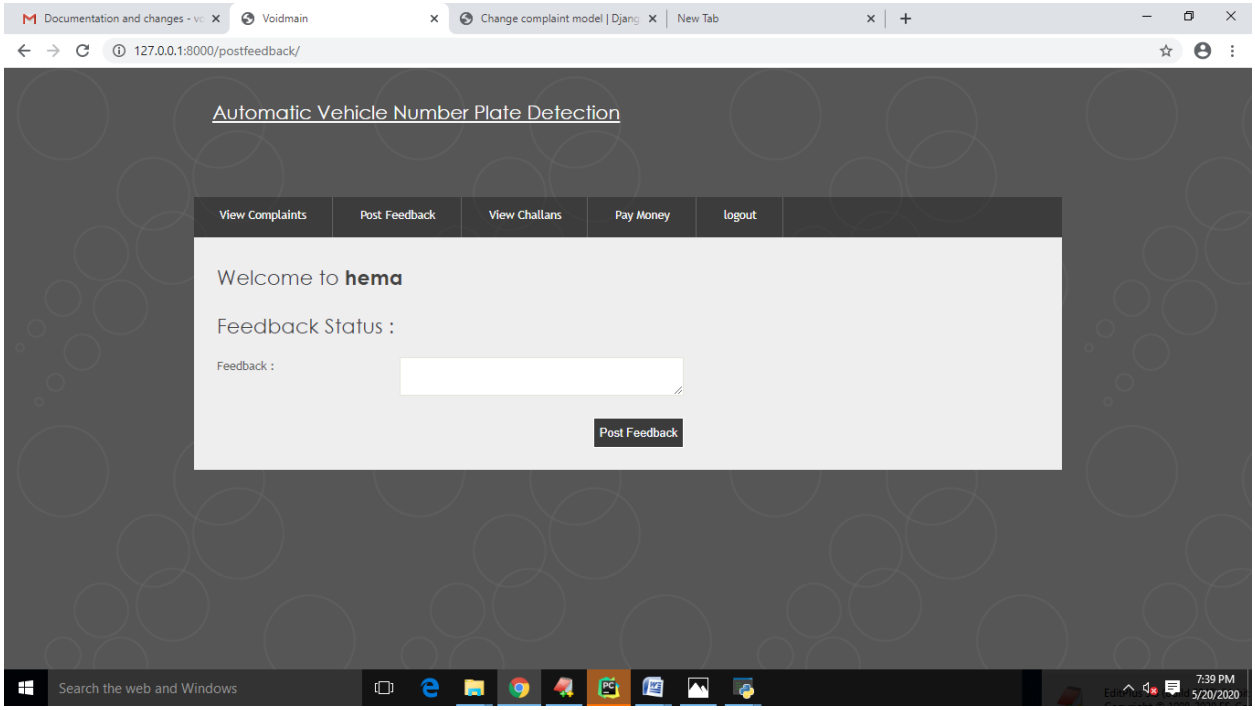

Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/viewfeedbacks/
Automatic Vehicle Number Plate Detection
Post Complaint
View Complaints
View Challans
View Feedbacks
logout
Welcome to admin
Status :
FeedBack
Posted On
Delete
hi
2020-05-20 18:54:17.121485
delete
Search the web and Windows
7:38 PM
5/20/2020

//==========
====


Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/viewcomplaints/
Automatic Vehicle Number Plate Detection
View Complaints
View Challans
Post Feedback
Pay Money
logout
Welcome to hema
Status :
Vehicle No
Photo
Description
Date
MCLRNF1
wrong parking
2020-05-20 19:21:41.469111
Search the web and Windows
7:38 PM
5/20/2020


Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/userviewchallans/
Automatic Vehicle Number Plate Detection
View Complaints
Post Feedback
View Challans
Pay Money
logout
Welcome to hema
Status :
Due Amount: -100
Complaint ID
Amount
Date
1
500
2020-05-20 18:49:33.931485
Search the web and Windows
7:38 PM
5/20/2020


Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/paymoney/
Automatic Vehicle Number Plate Detection
View Complaints
Post Feedback
View Challans
Pay Money
logout
Welcome to hema
Feedback Status :
Select Card Type :
Visa
Card NO :
CVV :
Amount :
Pay
Search the web and Windows
7:39 PM
5/20/2020


Documentation and changes - v
Voidmain
Change complaint model | Djang
New Tab
127.0.0.1:8000/postfeedback/
Automatic Vehicle Number Plate Detection
View Complaints
Post Feedback
View Challans
Pay Money
logout
Welcome to hema
Feedback Status :
Feedback :
Post Feedback
Search the web and Windows
7:39 PM
5/20/2020

TYPES OF TESTS

Test strategy and approach
Field testing will be performed manually and functional tests will be written in detail.

Test objectives
All field entries must work properly.
Pages must be activated from the identified link.
The entry screen, messages and responses must not be delayed.

Features to be tested
Verify that the entries are of the correct format
No duplicate entries should be allowedAll links should take the user to the correct page.

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Test cases

Test case Id	Test case Description	Test Data	Expected Result	Actual Result	Pass /Fail
1	Load Image	Testing Dataset	Dataset is loaded	Successfully loaded	Pass
2	Detect Possible Plates	Need to Find the Possible Number plates in Image	Detect Possible Number plates	Number plates detected	Pass
3	Detect Characters in Plates	Need to Find the Characters in Image	Detect possible Characters in plates	Possible Characters detected	Pass

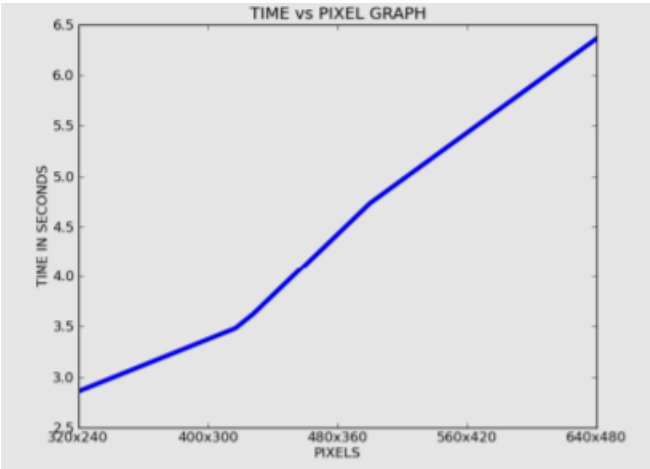
Results Analysis:

The system has been put to test for various measurements of performance and accuracy

Operation	Sample	Success	Fail	Success Ratio
License plate localization	100	92	8	92%
Character Separation	92	88	4	95.7%
Character Recognition	88	83	5	94.3%

Performance Analysis

During the initial days, the system suffered severe performance faults. It took more than eighteen seconds for recognizing the license plate and extracting the number. Subsequent research on algorithm and code optimization drastically brought down the operation time to two seconds. One of the key factor that determined performance was the size of input image. The flowing figure indicates this relationship.



Conclusion

Scanning number plate sometimes goes unsuccessful by using the shape analysis method to detect exact area of the plate. Future extension of this work is to develop character recognition using template matching algorithm. Detecting number plate characters during night times work efficient but it gets inefficient in case of sunny time

The message of this research is to show that free and open source technologies are matured enough for scientific computing domains. Python and OpenCV are good points of start for researchers and students of computer vision

The system works satisfactorily for wide variations in illumination conditions and different types of number plates commonly found in India. It is definitely a better alternative to the existing proprietary systems, even though there are known restrictions.

REFERENCES

[1] S. Uma1, M .Sharmila2 Implementation of License Plate Recognition System in ARM Cortex A8 Board IJCEM International Journal of Computational Engineering & Management, Vol. 19 Issue 3, May 2016 ISSN (Online): 2230- 7893 www.IJCEM.org

[2] N.Abirami1, Dr. J.S.Leena, Jasmine2 Accurate vehicle number plate recognition and Real time Identification using Raspberry Pi International Research Journal of Engineering and Technology (IRJET) Automatic Number Plate Recognition (ANPR) System for Indian conditions

[3] Prathamesh Kulkarni (Student Member, IEEE), Ashish Khatri, Prateek Banga, Kushal Shah* *University of Pune, Dept. of Electronics and Telecommunication, India.

[4] Pratiksha Jain , Neha Chopra, and Vaishali Gupta, "Automatic License Plate Recognition using OpenCV", International Journal of Computer Applications Technology and Research Volume 3– Issue 12, 756 - 761, 2014.

[5] Ankit Sharma, Dipti R Chaudhary, "Character Recognition Using Neural Network", IJETT, ISSN: 2231-5381,Vol.4 Issue 4, pp 662-667, April 2013

[6] Youngwoo Yoon, Kyu-Dae Ban, Hosub Yoon, and Jaehong Kim," Blob Extraction based Character Segmentation Method for Automatic License Plate Recognition System" Robot/Cognition System Research Department,IEEE.

[7] Ragini Bhat, Bijender Mehandia," Recognition of vehicle number plate using matlab", International journal of innovative research in electrical, electronics, instrumentation and control engineering vol. 2, issue 8, august 2014

[8] N.Vishwanath,S.Somasundaram, M.R. Rupesh Ravi, N. Krishnan Nallaperumal," Connected Component Analysis for Indian License Plate Infra-Red and Color Image Character Segmentation", IEEE International Conference on Computational Intelligence and Computing Research, 2012.

[9] Nima Asadi, " A Study of Automatic License Plate Recognition Algorithms and Techniques", Intelligent Embedded Systems

[10] P. Mathivanan, B. Ganesamoorthy and P. Maran," Watershed Algorithm Based Segmentation For Handwritten Text Identification", ICTACT Journal on Image And Video Processing, Volume: 04, Issue: 03,pp 767- 772, February 2014

[11] Attila József Kun, Zoltán Vámossy," Traffic Monitoring with Computer Vision",IEEE, Applied Machine Intelligence and Informatics, 2009

[12] Ray Smith," An Overview of the Tesseract OCR Engine", Document Analysis and Recognition, ICDAR 2007. 9th International Conference, Pages: 629 – 633

[13] Chirag Patel, Atul Patel, Dharmendra Patel," Optical Character Recognition by Open Source OCR Tool Tesseract: A Case Study", International Journal of Computer Applications (0975 – 8887) Volume 55– No.10, October 2012

[14] A. Katartzis and M. Petrou, "Current trends in super- resolution image reconstruction," Image Fusion: Algorithms and Applications, 2008.

[15] Jameson, H. S. Abdullah, S. Norul, A. N. Ghazali, N. Nur, and N. A.Zamani, "Multiple Frames Combination Versus Single Frame Super Resolution Methods for CCTV Forensic Interpretation," Journal of Information Assurance & Security, vol. 8, 2013.

[16]B. Zitova and 1. Fiusser, "Image registration methods: a survey," Image and Vision Computing, vol. 21, no. II, pp. 977- 1000, 2003.

[17] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," Signal Processing Magazine, IEEE, vol. 20, pp. 21-36, 2003.

[18] P. Vandewalle, S. SU, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," EURASIP Journal on Advances in Signal Processing, vol. 2006,2006