A Project Report

On

# Number Plate Detection Using k-NN

Submitted in Partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY**
**IN**
**INFORMATION TECHNOLOGY**

**SUBMITTED**
By

**ASHWINI TARUN - 19671A1207**

Under the esteemed guidance of

**M. A. MUNEER**
**ASSISTANT PROFESSOR**

## Department of Information Technology
**Accredited by NBA**

## J.B. Institute of Engineering & Technology
**(UGC AUTONOMOUS)**
(Affiliated to Jawaharlal Nehru Technological University, Hyderabad)

Baskar Nagar, Yenkapally, Moinabad Mandal, R.R. District, Telangana (India)-500075

2022-2023

# J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY

**(UGC AUTONOMOUS)**

**(Accredited by NAAC, Permanently Affiliated to JNTUH)**

Bhaskar Nagar,Yenkapally, Moinabad Mandal, R.R. Dist.Telangana(India) -500 075

## DEPARTMENT OF INFORMATION TECHNOLOGY

**Accredited by NBA**

## CERTIFICATE

This is to certify that the project report entitled "**Number Plate Detection Using k-NN** " being submitted to the Department of Information Technology, J.B. Institute of Engineering and Technology, in accordance with Jawaharlal Nehru Technological University regulations as partial fulfillment required for successful completion of Bachelor of Technology is a record of bonafide work carried out during the academic year 2022-23 by,

**ASHWINI TARUN - 19671A1207**

| **Internal Guide** | **Head of the Department** |
|---|---|
| M. A. MUNEER | Dr. L. SRIDHARA RAO |
| ASSISTANT PROFESSOR | ASSOCIATE PROFESSOR |

**External Examiner**

**J.B. INSTITUTE OF ENGINEERING & TECHNOLOGY**

**(UGC AUTONOMOUS)**

**(Accredited by NAAC, Permanently Affiliated to JNTUH)**

Baskar Nagar, Yenkapally, Moinabad Mandal, R.R. Dist.Telangana(India) -500 075

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**Accredited by NBA**

# DECLARATION

We hereby certify that the Main Project report entitled **"Number Plate Detection Using k-NN"** carried out under the guidance of **M. A. MUNEER , Assistant Professor** is submitted in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Information Technology.** This is a record of bonafide work carried out by us and the results embodied in this project report have not been reproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**Date:**
**Place:**

**ASHWINI TARUN – 19671A1207**

# ACKNOWLEDGEMENT

At outset we express our gratitude to almighty lord for showering his grace and blessings upon us to complete this Main Project. Although our name appears on the cover of this book, many people had contributed in some form or the other to this project Development. We could not have done this Project without the assistance or support of each of the following.

First of all we are highly indebted to **Dr. P. C. KRISHNAMACHARY**, Principal for giving us the permission to carry out this Main Project.

We would like to thank **Dr. L. SRIDHARA RAO,** Associate Professor & Head of the Department of INFORMATION TECHNOLOGY**,** for being moral support throughout the period of the study in the Department.

We are grateful to **M. A. MUNEER,** Assistant Professor of the Department of INFORMATION TECHNOLOGY, for his valuable suggestions and guidance given by him during the execution of this Project work.

We would like to thank Teaching and Non-Teaching Staff of Department of Information Technology for sharing their knowledge with us.

**ASHWINI TARUN - 19671A1207**

# ABSTRACT

Transport surveillance and recording each vehicle's information in a journal are both exceedingly labour-intensive tasks. A " Detection and recording of number plates automatically " can be used to resolve this issue. Using the k-NN model, we have suggested a method in this research study enabling the system to be able to detect the number plate (a machine learning technique used for classification), and information about the car, such as the number plate, a picture of the vehicle, the time, etc., is stored in the database.

The k-NN model is trained using several alphabetic and numeric styles to get a high level of accuracy. Additionally, we used a separate dataset of cars in our research work to validate the findings.

The examination of real-time traffic data is then performed using the video streams. Systems for monitoring traffic on the road usually contain features for vehicle recognition, tracking, and identification. The technique of analysing a vehicle's characteristics to ascertain its owner is known as vehicle recognition. One such method is the recognition of number plates, which is possible with the use of optical character recognition algorithms. Algorithms that recognise number plates can help law enforcement locate offenders, identify the owner of the vehicle, and bring them to justice.

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# INTRODUCTION

# 1. INTRODUCTION

A metal or plastic plate used for official identifying purposes that is fastened to a car, truck, or trailer is called a number plate. Road vehicles, including cars, trucks, and motorcycles, must have number plates in all countries.. Depending on the jurisdiction, they might also be necessary for other types of vehicles like bicycles, boats, or tractors. There are frequently legal restrictions on how number plates should look. For instance, limitations on plate design, registration number positioning, and plate colour. In some jurisdictions, number plates are given to a specific vehicle and remain there for the duration of the vehicle. The number plate must be returned if the car is disassembled or exported to another nation.

The subject of study known as intelligent transport systems (ITS) is exceedingly diverse and has attracted the attention of many academics. The use of information and communication technologies to improve the efficiency of transportation systems is discussed. As a result, both commuters and drivers can navigate the roadways with increased convenience and mobility. In addition to parking management systems, electronic toll collection systems, inter-vehicle communications, and road surveillance systems, ITS can have a wide range of applications thanks to the development of fast and reliable data transmission and algorithms for the automation of manual processes. Because of the growth and improvements in the field of computer vision and image processing, road surveillance systems were one of the applications mentioned that particularly intrigued the researchers.

In recent studies, cameras have been erected and placed in strategic locations to catch the roads that need to be monitored. Then, from the video feeds, real-time traffic data analysis is done. Vehicle detection, tracking, and identification are frequently included of road surveillance systems. Vehicle recognition is the process of analyzing a vehicle's attributes in order to determine who owns it. One such technique is the recognition of number plates, which can be done using algorithms for optical character

recognition. Using algorithms to recognize number plates can assist police catch those who break traffic laws, identify the owner of the car, and prosecute them for their crimes.

Many scholars have used OCR, k-NN, algorithms to work on character recognition. A couple of papers are "Handwritten Character Recognition Using K-NN Classification Algorithm" by Siddhartha Roy and M. Saravanan and "Character Recognition Using KNN Algorithm" by Sai Jahnavi Bachu. They have demonstrated that utilizing these algorithms, we can detect characters with a high degree of accuracy.

# LITERATURE SURVEY

# 2. LITERATURE SURVEY

## 2.1 Definition of Terms:

Character recognition is not a new problem but can be traced back to systems before the inventions of computers. The earliest OCR systems were not computers but mechanical devices that are able to recognize characters, but are very slow of speed and low at accuracy [1]. In 1951, M. Sheppard invented a reading and robot GISMO that can be considered as the most earliest work on modern OCR. GISMO can read musical notations as well as words on a printed page one by one [2]. However, it can only recognize 23 characters. The machine also has the capability to copy a typewritten page. J. Rainbow, in 1954, devised a machine which can read the uppercase typewritten English characters, one per minute [3]. The early OCR systems were neglected due to errors and slow recognition rate. Hence, much research efforts were not put on the topic during 60's and 70's [4]. The developments were done on government agencies and large corporations like banks, newspapers and airlines etc. Because of the complexities associated with recognition, it was felt that there should be standardized OCR fonts for making the task easy for recognition for OCR. Hence, OCRA and OCRB were developed by ANSI and EMCA in 1970, which provided comparatively acceptable recognition rates. During the past thirty years, substantial research has been conducted on OCR [5].

Which has lead to the emergence of document image analysis (DIA), multi-lingual, handwritten and omni-font OCRs. Despite these extensive research efforts, the machine's ability to read text is still far below the human. Hence, the current OCR research is being conducted on improving accuracy and speed of OCR for diverse style documents printed/ written in unconstrained environments [6]. There has not been availability of any open source or commercial software available for complex languages like Urdu or Sindhi etc. In pattern recognition, the k-nearest neighbors algorithm (k-NN) is a non-parametric method which is used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output

depends on whether k-NN is used for classification or the regression process [7]. In k-NN classification, the output is a class membership. An object which is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to a class of that particular single nearest neighbor.

A brief description on the history of OCR is as follows. In 1929 Gustav Tauschek obtained a patent on OCR in Germany, followed by Handel who obtained a US patent on OCR in USA in 1933. In 1935 Tauschek was also granted a US patent on his OCR method. Tauschek's machine was a mechanical device which used template and a photo detect [8]. RCA engineers in 1949 worked on the first primitive computer-type OCR that helps blind people for the US Veterans Administration, but instead of converting the printed characters to machine language, their device converted it to machine language and then spoke the letters. It is too expensive and was not used after testing.

In1978,Kurzweil Computer Products began selling a commercial version of the optical character recognition computer program. LexisNexis was one of the first customers, and bought the program to upload paper legal and news documents onto its nascent online database. The methods are tedious and are highly involved in computational power. In Di gesu the idea of using both intensities and spatial information has been considered to take into account local information used in human perception [9]. The number of new methodologies and strategies were proposed over the past years to find global as well as local solutions in nonlinear multimodal function optimization. In addition attempts were also been made to use Fuzzy Logic, Artificial Neural Network for optical character recognition. With the help of crowding multiple peaks can be maintained in multimodal optimization problem. Crowding method is extremely reliable in detecting the peaks on bimodal histogram. It makes use of an OCR system which uses the histogram equalization to extract images. The histogram used by the mentioned algorithm is bimodal in nature so it can be divided into two classes.

Genetic algorithm is further used to select the threshold from the histogram for

extracting the object from the background. Claudiu was investigated using simple training data pre-processing which gave experts with less errors correlated than that of different nets trained on the same or bootstrapped data [10]. Hence committees that simply average the expert outputs considerably improve recognition rates. Our committee-based classifiers of isolated handwritten characters were the first on par with human performance and can be used as the basic building blocks of any OCR system (all our results were achieved by software running on powerful yet cheap gaming cards). Georgios has represented a methodology for offline handwritten character recognition [11]. The proposed methodology relies on a new feature extraction technique based on recursive subdivisions of the character image so that the resulting sub-images at every iteration were balanced (approximately equal) numbers of foreground pixels, as far as this is possible. Feature extraction is allowed by a two- stage classification scheme based on the level of granularity of the feature extraction method. Classes with high values in the confusion matrix are merged at a certain level and for each group of merged classes, granularity features from the level that best distinguishes them were employed[12]. Two handwritten character databases (CEDAR and CIL) as well as two handwritten digit databases (MNIST and CEDAR) which were used in order to demonstrate the effectiveness of the proposed technique.

Sankaran presented a novel recognition approach which results in a 15% decrease in word error rate on heavily degraded Indian language document images. OCRs have considerably good performance on good quality documents, but easily fail in presence of degradations [7]. Also, classical OCR approaches perform poorly over complex scripts such as those for Indian languages. Sankaran addressed these issues by proposing to recognize character n-gram images, which are basically groupings of consecutive character/ component segments. Their approach was unique, since they use the character n-grams as a primitive for recognition rather than for post-processing [8]. By exploring the additional content present in the character n-gram images, we enable better disambiguation between confusing characters in the recognition phase. Labels obtained from recognizing the constituent n- grams are then fused to obtain a label for the word that emitted them.

Their methods are inherently robust to degradations such as cuts and merges which are common in digital libraries of scanned documents. We also present a reliable and scalable scheme for recognizing character n-gram images. Zhang discussed the misty, foggy, and hazy weather conditions lead to image color distortion and reduce the resolution and the contrast of the observed object in the outdoor scene acquisition. In order to detect and remove haze, this article proposes a novel effective algorithm for visibility enhancement from a single gray or color image [4]. It can be considered that the haze mainly concentrates in one component of the multilayer image, the haze-free image is constructed through haze layer estimation based on the image filtering approach using both low-rank technique and the overlap averaging scheme. By using parallel analysis with Monte Carlo simulation from the coarse atmospheric veil by the median filter, the refined smooth haze layer is acquired with both less texture and retaining depth changes [9]. With the use of dark channel prior, the normalized transmission coefficient is calculated to restore fogless image. Experimental results show that the proposed algorithm is a simpler and efficient method for clarity improvement and contrast enhancement from a single foggy image. Moreover, this can be comparable with the state-of-the-art methods, and even has better results than them.

# SYSTEM ANALYSIS

# 3. SYSTEM ANALYSIS

## 3.1. EXISTING SYSTEM

Number Plate Recognition using OCR is an existing system. In this existing system an image is sent, the number plate recognition system will attempt to locate the number plate before sending it to the OCR for character recognition. The number plate area is once again searched from the image if OCR cannot read the characters from it.

## 3.2 Drawbacks of Existing System

- It's not always of excellent quality.
- Both expensive and time-consuming.
- Occasionally incorrect.
- Being unable to recognize some languages.
- Inaccuracy with damaged texts.

## 3.3 PROPOSED SYSTEM

In this paper we propose a method through which the number plate recognition can be achieved more precisely. Here we use K-Nearest Neighbor technique for character recognition. The first stage is capturing the license plate images for dataset collection. These images will be

used as the training and testing data. Second, the image pre-processing stage includes cropping, resizing and Gray scaling. The third stage is the segmentation process includes edge detection and thresholding. The fourth stage is feature extraction which applies contour detection. In the next stage, all the images from the previous stage are segmented to obtain letters and numbers images. These images will be used as the input for training and testing data. The training and testing data are processed by using K-Nearest Neighbor (KNN) algorithm.

## 3.4 Advantages of Proposed System

- More efficient than existing approaches.
- It may help classifying the things very fast.
- It may produce high accuracy.
- It is much more feasible and effective in classifying the elements in the segmentation phases.

## 3.5 Software Development Life Cycle Model

**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 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:

  - Evaluating the first prototype in terms of its strengths, weakness, and risks.
  - Defining the requirements of the second prototype.
  - Planning an designing the second prototype.
  - 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

The following diagram shows how a spiral model acts like:

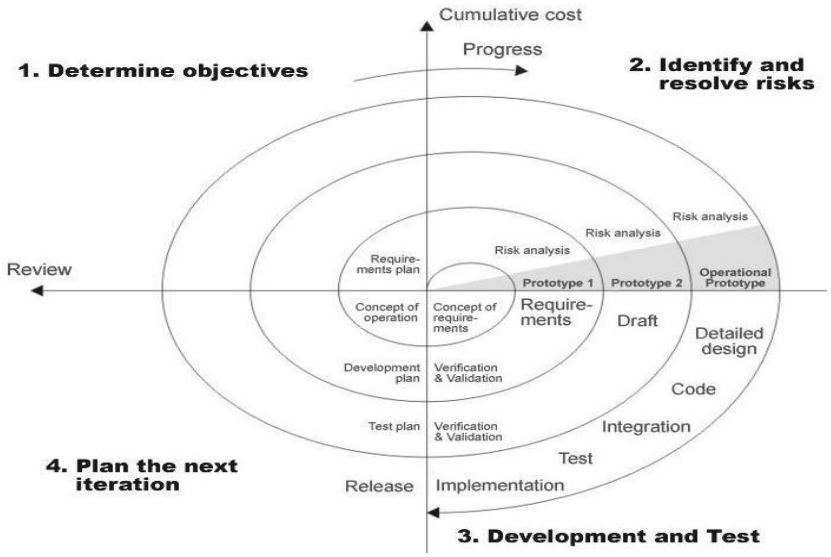


Figure: 3.1 Spiral Model

## Advantages

- Estimates(i.e. budget, schedule etc .) become more relistic as work progresses, because important issues discoved earlier.
- It is more able to cope with the changes that are software development generally entails.
- Software engineers can get their hands in and start woring on the core of a project earlier.

## 3.6 Project Implementation Plan

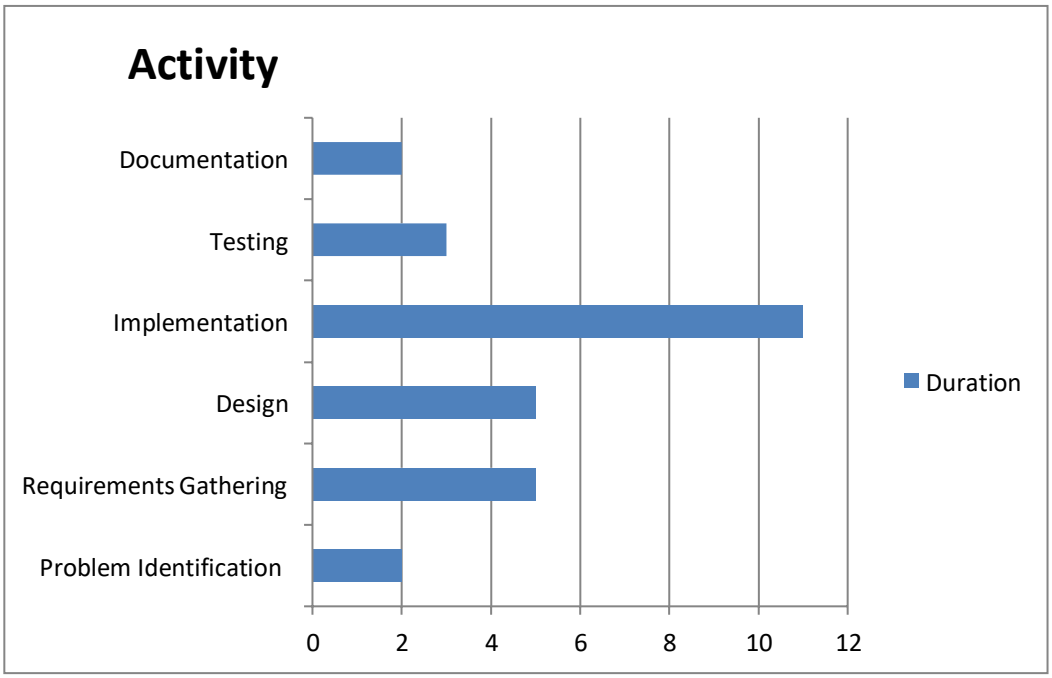


Figure: 3.2 Project Implementation Plan

# SOFTWARE
# REQUIREMENT SPECIFICATION

# 4. SOFTWARE REQUIREMENT SPECIFICATIONS

## 4.1 Functional Requirements

- ➔ An Image with proper resolution should be sent as an input the system.
- ➔ Model need to be trained on the character detection.
- ➔ We need to train an algorithm with different characters of different styles.
- ➔ The software should be able to detect the Characters on the Number Plate.
- ➔ We need to check the accuracy of the detection

## 4.2 Non-Functional Requirements

The major non-functional Requirements of the system are as follows

1. **Usability:** The system is designed in a way where the user needs to put the image information.
2. **Reliability:** The system is more reliable because of the qualities that are inherited from the chosen python platform. The code built by using python is more reliable.
3. **Accessibility:** It can be easily accessible, that is click and run.
4. **Efficiency:** Resources consumption for given load is quite low.
5. **Fault tolerance:** Our system is fault tolerant due to insufficient hardware and Training.
6. **Robustness:** Our system is not capable to cope with errors during execution.

7. **Scalability:** Our project is scalable, that is we can add more resources to our project without disturbing the current scenario.

## 4.3 Software Requirement Specifications

- Operating System : Windows11
- Language : Python
- Tool : VSCode

**Modules Required:**

- Open Computer Vision
- Sci-kit Learn
- Numpy

## 4.4 Hardware Requirement Specifications

- System Type: 64-bit Operating System, I3 Processor.
- Hard Disk: 5 GB.
- RAM : 2 GB and more.
- 3 Mega pixel Camera
- Monitor : 15 inch VGA Colour
- Mouse : Hp Mouse
- Keyboard : Standard Keyboard

# SYSTEM DESIGN

# 5. SYSTEM DESIGN

## 5.1 SYSTEM ARCHITECTURE

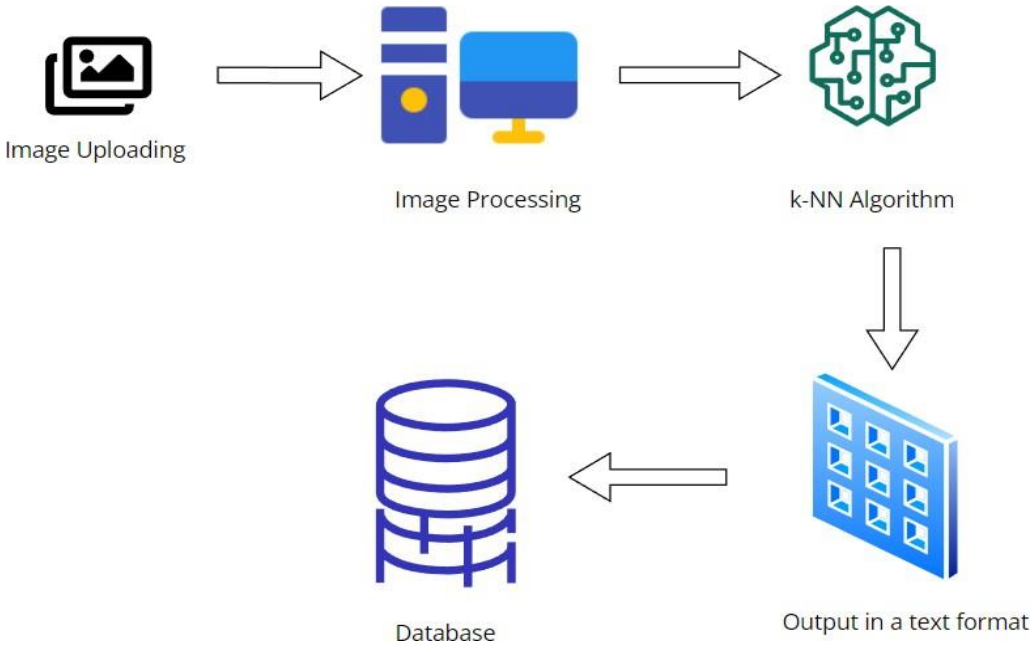


Figure: 5.1 System Architecture

Once the user uploads the image to the system, the system will process the image and send the image to the Character recognition phase where the k-NN algorithm has been already trained with different styles. Now, with all the learnings and experiences the k-NN model with try to classify the text on the number plate. Later the text can be saved in database.

## 5.2 DESIGN TOOL USED – VISUAL PARADIGM TOOL

### Introduction

Visual Paradigm (VP-UML) is a UML CASE Tool supporting UML2, SysML and Business Process Modeling Notation (BPMN)from the Object Management Group (OMG). In addition, modeling support, it provides report generation and code engineering capabilities including code generation. It can reverse engineer diagrams from code, and provide round-trip engineering for various programming languages.

### Contents

- Product Editions
- UML Modeling
- Requirements Management
- Business Process Modeling
- Data Modeling

### Product Editions

Higher-priced editions provide more features.

The following editions were available:

- Community Edition

- Modeler Edition
- Standard Edition
- Professional Edition
- Enterprise Edition

## UML Modeling

Visual Paradigm supports 13 types of diagrams:

- Class Diagram
- Use Case Diagram
- Sequence Diagram
- Communication Diagram
- State Machine Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram
- Package Diagram

- Object Diagram
- Composite Structure Diagram
- Profile Diagram
- Timing Diagram
- Interaction Overview Diagram

## Requirements Management

- Visual Paradigm supports requirements management including user stories, use cases, SysML requirement diagrams and textual analysis.
- A SysML requirement diagram specifies the capability or condition that must be delivered in the target system. Capability refers to the functions that the system must support. Condition means that the system should be able to run or produce the result given a specific constraint. Visual Paradigm provides a SysML requirement diagram for specifying and analyzing requirements

## Business Modeling

Supports BPMN 2.0 for modeling of business processes. The latest version (Aug 2016) also supports Case Management with CMMN

## Data Modeling

Visual Paradigm supports both Entity Relationship Diagrams (ERD) and Object Relational Mapping Diagrams (ORMD). ERD is used to model the relational database. ORMD is one of the tools to show the mapping between class from object-oriented world and entity in relational database world.

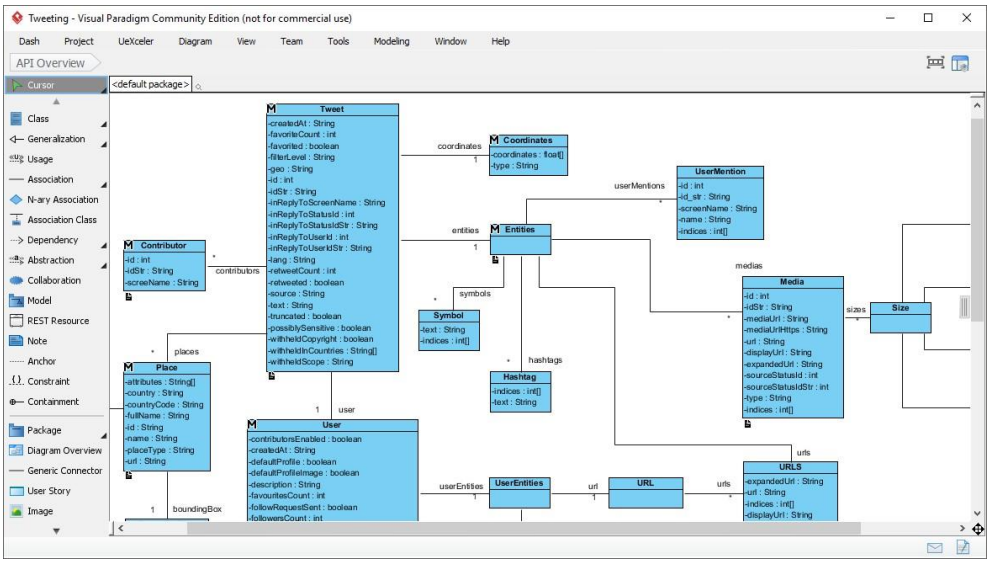


Figure: 5.2 Interface of Visual Paradigm

Visual Paradigm 17.0 running on Windows 11

| | |
|---|---|
| Developer(s) | Visual Paradigm International Ltd. |
| Initial Release | 20 June 2002; 15 years ago |
| Stable Release | 17.0 / August 01, 2022 |
| License | Proprietary with Free Community Edition |

## 5.3 UML DIAGRAMS

### 5.3.1 USECASE DIAGRAM :

It shows the set of use cases, actors & their relationships. In our project we have 1 actor and 1 system where the user needs to input the image to the system and the system processes the image and outputs the text on the number plate.

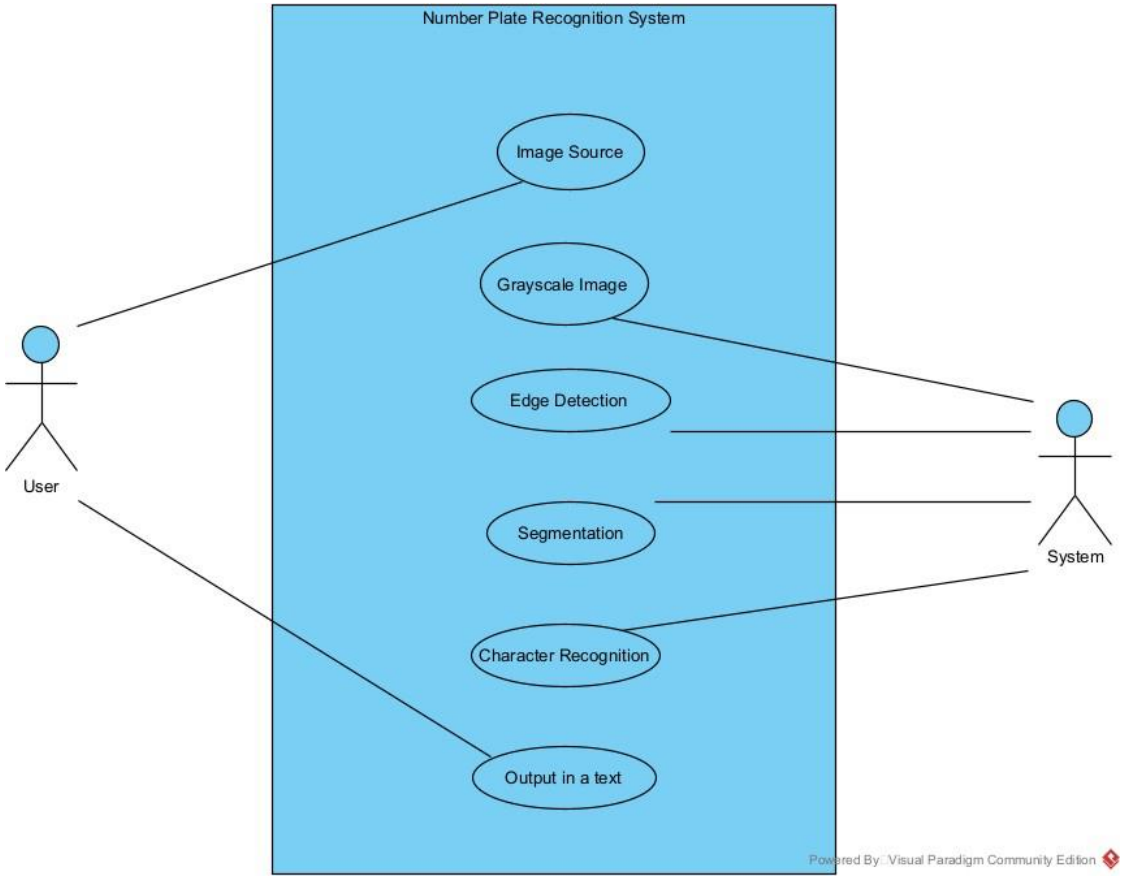


Figure: 5.3 Use Case Diagram

### 5.3.2 CLASS DIAGRAM:

*A class diagram* shows a set of classes, interfaces, and collaborations and their relationships. These diagrams are the most common diagram found in modeling object-oriented systems. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system.

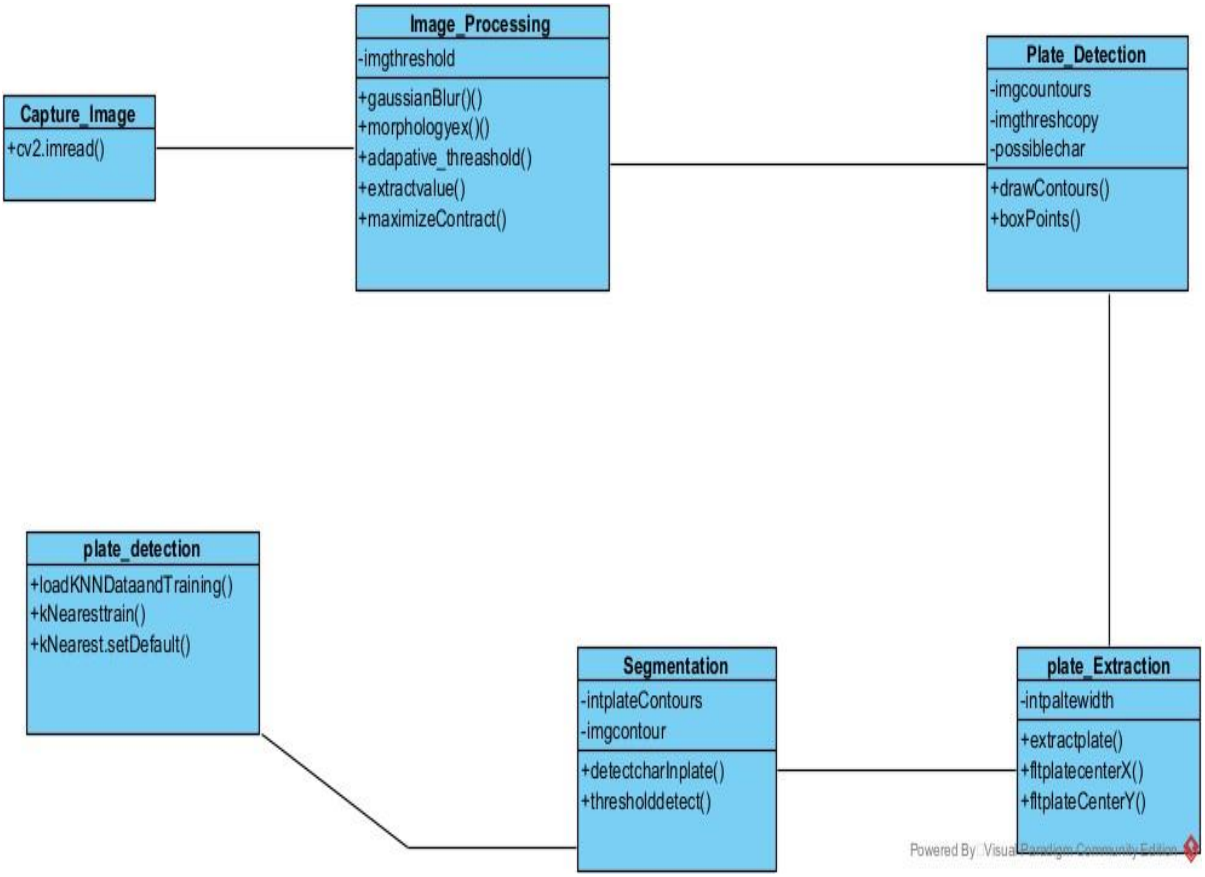


Figure: 5.4 Class Diagram

### 5.3.3 SEQUENCE DIAGRAM:

A *sequence diagram* is an interaction diagram that emphasizes the time-ordering of messages; a *collaboration diagram* is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Sequence diagrams and collaboration diagrams are isomorphic, meaning that you can take one and transform it into the other.

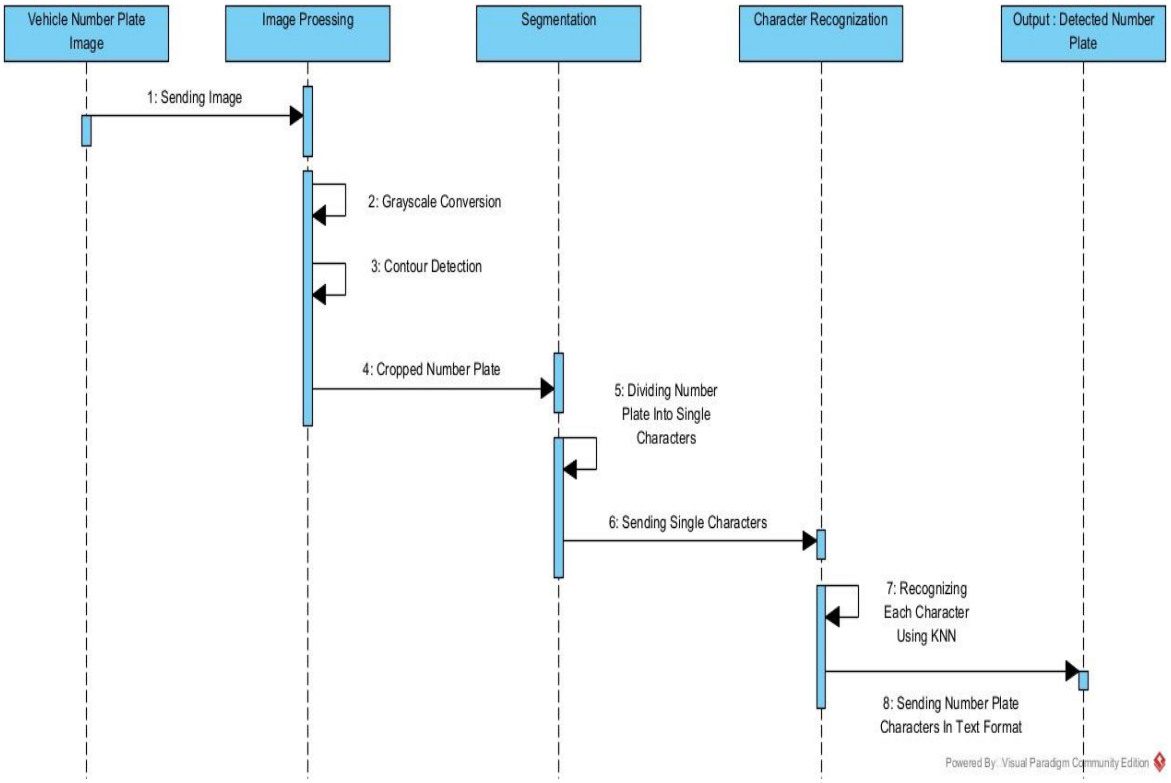


Figure: 5.5 Sequence Diagram

### 5.3.4 ACTIVITY DIAGRAM:

An *activity diagram* is a special kind of a state chart diagram that shows the flow from activity to activity within a system. Activity diagrams address the dynamic view of a system. They are especially important in modeling the function of a system and emphasize the flow of control among objects.

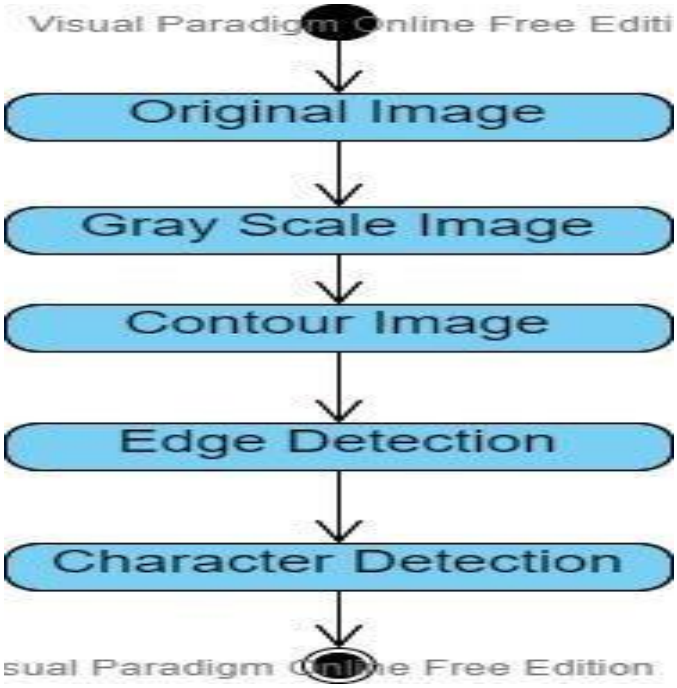


Figure: 5.6 Activity Diagram

# IMPLEMENTATION

# 6. IMPLEMENTATION

## 6.1 INTRODUCTION

### Software Environment:

Python is a programming language. Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other language use punctions, and it has fewer syntactical constructions than other programming languages.

### Python Programming Language

- **Python is Interpreted –** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive –** You can actually sit at a Python prompt and interact with the interact with the interpreter directly to write your programs.
- **Python is Object-Oriented –** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language –** Python is a great language for the beginner-level python programmer and supports the development of wide range of applications from simple text processing to WWW browsers to games.

### History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Module-3, C, C++, Algol-68, Smalltalk, and UNIX shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

## Python Virtual Machine (PVM)

We know that computers understand only machine code that comprises 1s and 0s. Since computer understands only machine code, it is imperative that we should convert any program into machine code before it is submitted to the computer for execution. For this purpose, we should take the help of a compiler. A compiler normally converts the program source code into machine code.

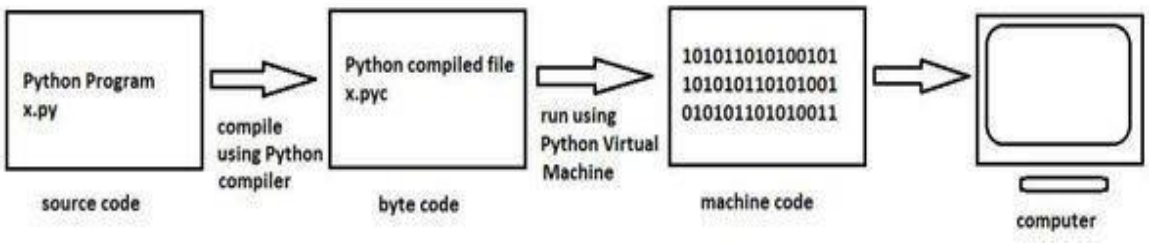


Figure: 6.1 Flow from Source code to Machine code in PVM

Python compiler does the same task but in a slightly different manner. It converts the program source code into another code, called byte code. Each Python program statement is converted into a group of byte code instructions. Then what is byte code? Byte code represents the fixed set of instructions created by Python developers representing all types of operations. The size of each byte code instruction is 1 byte (or 8bits) and hence these are called byte code instructions. Python organization says that there may be newer instructions added to the existing byte code instructions from time to time. We can find byte code instructions in the .pyc file. Following Figure shows the role of virtual machine in converting byte code instructions into machine code. The role of Python Virtual Machine (PVM) is to convert the byte code instructions into machine code so that the computer can execute those machine code instructions and display the final output. To carry out this conversion, PVM is equipped with an interpreter. The

interpreter converts the byte code into machine code and sends that machine code to the computer processor for execution. Since interpreter is playing the main role, often the Python Virtual Machine is also called an interpreter.

**Python Features**

Python's features include

- **Easy-to-learn** -- Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read --** Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain --** Python's source code is fairly easy-to-maintain.
- **A broad standard library --** Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode --** Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable --** Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable --** you can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases --** Python provides interfaces to all major commercial databases.
- **GUI Programming --** Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable -** Python provides a better structure and support for large programs than shell scripting.

## 6.2 TECHNOLOGIES USED

We have used few technologies such as Open-cv, sk-Learn, Tkinter. They have explained in detail below:

### 6.2.1 Open-cv

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

The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When OpenCV was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi-core processing.

### Image-Processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image and or to extract some useful information from it.

If we talk about the basic definition of image processing then "Image processing is the analysis and manipulation of a digitized image, especially in order to improve its quality".

An image may be defined as a two-dimensional function f(x, y), where x and y are spatial(plane) coordinates, and the amplitude of fat any pair of coordinates (x, y) is called the intensity or grey level of the image at that point.

In another word An image is nothing more than a two-dimensional matrix (3-D in case of coloured images) which is defined by the mathematical function f(x, y) at any point is giving the pixel value at that point of an image, the pixel value describes how bright that pixel is, and what colour it should be.

Image processing is basically signal processing in which input is an image and output is image or characteristics according to requirement associated with that image.

## Sk-Learn

Scikit-learn is mainly coded in Python and heavily utilizes the NumPy library for highly efficient array and linear algebra computations. Some fundamental algorithms are also built in Cython to enhance the efficiency of this library. Support vector machines, logistic regression, and linear SVMs are performed using wrappers coded in Cython for LIBSVM and LIBLINEAR, respectively. Expanding these routines with Python might not be viable in such circumstances.

Scikit-learn works nicely with numerous other Python packages, including SciPy, Pandas data frames, NumPy for array vectorization, Matplotlib, seaborn and plotly for plotting graphs, and many more.

## k-Nearest Neighbor

K-Nearest Neighbour is a good supervised learning technique in machine learning. Based on the idea that the new data point and old instance point can be comparable, the K-NN algorithm places a new data point in the categories that are similar to the already trained and loaded categories. When a new data point is introduced, the K-NN algorithm classifies it and saves it all depending on how similar it is to the current data. This indicates that we can quickly and accurately classify new data points in the provided categories using K-NN.

Even though the K-NN approach may be used to solve both classification and regression problems, it is often applied to classification tasks. Since K-NN is a non-parametric method, the underlying data is not taken into account..

The K-NN algorithm is also known as the "lazy learner" algorithm because it stores the training dataset before performing an action on it at the time of classification

instead of immediately learning from it. The K-NN algorithm simply stores the dataset during the training phase and uses it to classify new data points into a category that is very similar to the training dataset..

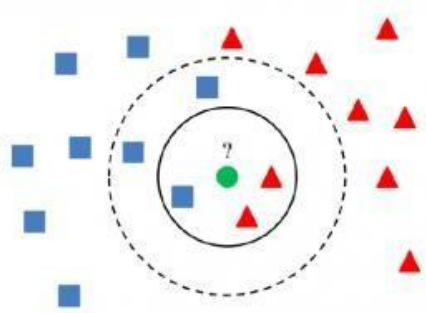


Figure: 6.2 k-NN Algorithm

Choose the neighbour with the number K. Determine the Euclidean distance between K neighbours. Pick the K closest neighbours based on the Euclidean distance estimate.

$$d(x,\ y)\ =\ \sqrt{\sum_{i=1}^{n}(y_i - x_i)^2}$$

Figure: 6.3 Manhattan distance

Among these k neighbours, total the number of data points in each category. Assign the new data points to the category where the neighbour count is at its highest. Our model is complete.

### 6.2.2 Tkinter

We have made the user interface based on tkinter module which supports by python. Here we made two clickable buttons called as Browser and Classify. Using Browser button the user can able to browser the image and upload the image. Next, we have Classify button. By using this the user can able to view the output which is classified image.

- Background : We have used internal method for setting the background.
- Button : We have used internal method for setting the buttons.

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

There are number of options which are used to change the format of the widget. Number of options can be passed as parameters separated by commas. Some of them are listed below.

- bd: to set the border width in pixels.
- bg: to set the normal background color.
- cursor: to set the cursor used in the canvas.
- highlight color: to set the color shown in the focus highlight.
- width: to set the width of the widget.
- height: to set the height of the widget.

## 6.3 Coding Standards

DMAIC is a data driven quality strategy used to improve processes It is an integral part of six sigma intiative ,but in general can be implemented as a standalone quality improvement procedure or as part of other process improvement initiatives such as lean.

DMAIC is an acronym for the five phases that make up the process:

- **Define** the problem, improvement activity for improvement, the project goals, and customer requirements.
- **Measure** process performance
- **Analyze** the process to determine root causes of variation ,poor performance
- **Improve** process performance by addressing and eliminating the root causes
- **Control** the improve process and future process performance

Define, measure, analyze, improve and control (DMAIC) is a structured problem-solving method. Each phase builds on the previous one, with the goal of implementing long-term solutions to problems. Sometimes, project leaders or sponsors don't feel a formal approach is necessary, but most problem-solving efforts benefit from a disciplined method.

The tools used in the define phase lay the foundation for the project. The team accurately and succinctly defines the problem, identifies customers and their requirements, and determines skills and areas that need representation on the project team. Individuals who must be part of the core team or be ad-hoc members are identified, and project measures, financials and a communication plan are established.

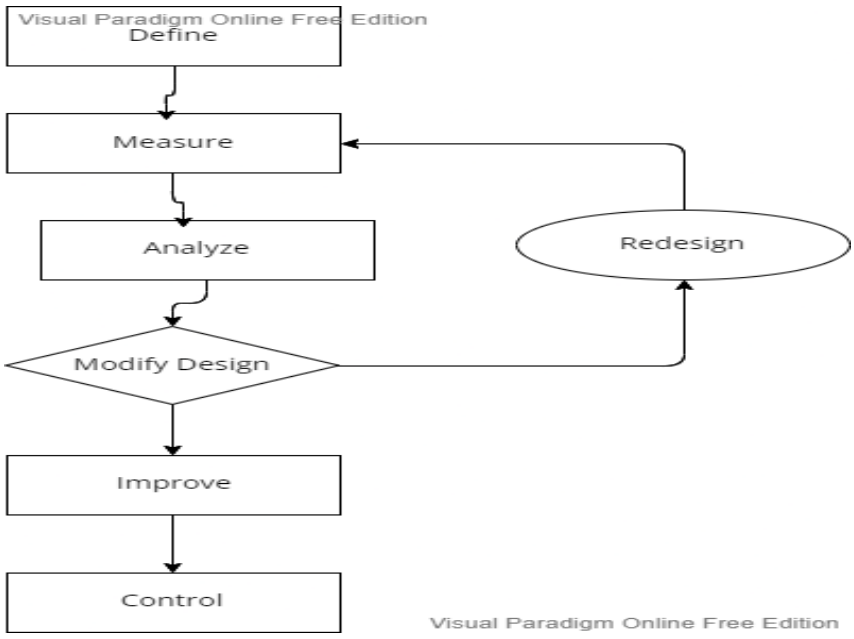


Figure: 6.4 DMAIC

The measure phase is when the true process is identified and documented. Process steps, and corresponding inputs and outputs are identified. Measurement systems are identified or developed, and validated and improved as required. Baseline performance is established with trustworthy data.

In the analyze phase, the critical inputs are identified. Inputs that have a strong relationship with the outputs and root causes are determined. These critical inputs are the drivers of performance.

In the improve phase, potential solutions are identified and evaluated, and the process is optimized. The critical inputs that must be controlled to maintain performance that reliably satisfies the customer are determined. Process capability and project financials are estimated.

The control phase establishes mistake-proof, long-term measurement and reaction plans. The team develops standard operating procedures and establishes process capability. Project financials are updated, verified and reported. Control plans are established with reaction plans, ownership and control is transitioned to the process owner, and lessons are documented. The team documents opportunities to spread the outcomes to other areas in the organization.

**When to use DMAIC**

When improving a current process, if the problem is complex or the risks are high, DMAIC should be the go-to method. Its discipline discourages a team from skipping crucial steps and increases the chances of a successful project, making DMAIC a process most projects should follow.

There are two approaches to implementing DMAIC. The first is the team approach in which individuals who are skilled in the tools and method, such as quality or process improvement experts, lead a team. The team members work on the project part-time while caring for their everyday responsibilities. The quality or process improvement expert might be assigned to several projects. These are long-duration projects taking months to complete.

## 6.4 MODULES

1. Image Processing
2. Segmentation
3. Character Classification

### 6.4.1 IMAGE PROCESSING :

#### i. Gray Scale Image

First the image has to be converted into a Grayscale image. A Grayscale images are created by storing the intensity of light in each pixel. It appears to be a black-and-white picture. It ranges in value from dark black to dazzling white..

#### ii. Noise Reduction

We can use a Gaussian filter for the noise reduction. Just consider that if we capture an image in less light, the image may have some noise in it. Here, we can use a Gaussian filter or blur to reduce these noises. In the case of texts, this will make the characters appear more clearly..

#### iii. Edge Detection

The way in which we can find the areas in a digital image that have sharp changes in brightness is called edge detection. The edges of an image are places where the brightness of the picture may vary sharply. The Roberts detection of edges and Sobel detection of edges are the two major ways to perform edge detection in the image processing phase..

**iv. Cropping of Image**

Here, the image is cropped to the number plate detection based on edge detection. The remaining sections are eliminated because they are superfluous.

**v. Resizing of the image**

Because the image was taken at a variety of distances, each cropped image has a different size. The images should, however, be input in the same size. So, all of the cropped images should be resized to the same size.

## 6.4.2 Segmentation:

The number plate is separated into individual characters in this stage, which will be useful in subsequent steps. This is important because image segmentation, which divides an image into smaller sets called "image segments" and reduces the complexity of the image while enabling further processing of each segment, is a technique..

The segmentation stage of the image recognition system is crucial because it will extract the features for additional processing, like recognition, the segmentation stage of the image recognition system is essential. Images can be segmented for the purpose of classifying image pixels.

## 6.4.3 Character Classification:

The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another. We have used k-NN Algorithm which has been already trained that will helpful for the classifying the characters.

## 6.5 UNIT TESTCASES

| Test Case Id | Test Case Description | Expected Results | Actual Result | Pass/Fail |
|---|---|---|---|---|
| 01 | User Interface | User should able to upload and view the result images on the system | Users are able to interact with UI and also able to upload and view the output images | Pass |
| 02 | Image Browsing | The User should able to browser and upload the image | The User is able to browse and upload the image | Pass |
| 03 | Image Classify | Once if the user clicks the classify button he should be able to view the classified output | The user is able to hit the classify button and view the classified button. | Pass |

Table: 6.1 Unit Test Cases

# SYSTEM TESTING

# 7. SYSTEM TESTING

## 7.1 TESTING PLAN

The software engineering process can be viewed as a spiral. Initially system engineering defines the role of software and leads to software requirement analysis where the information domain, functions, behavior, performance, constraints and validation criteria for software are established. Moving inward along the spiral, we come to design and finally to coding. To develop computer software we spiral in along streamlines that decrease the level of abstraction on each turn.

A strategy for software testing may also be viewed in the context of the spiral. Unit testing begins at the vertex of the spiral and concentrates on each unit of the software as implemented in source code. Testing progress by moving outward along the spiral to integration testing, where the focus is on the design and the construction of the software architecture. Talking another turn on outward on the spiral we encounter validation testing where requirements established as part of software requirements analysis are validated against the software that has been constructed. Finally we arrive at system testing, where the software and other system elements are tested as a whole.

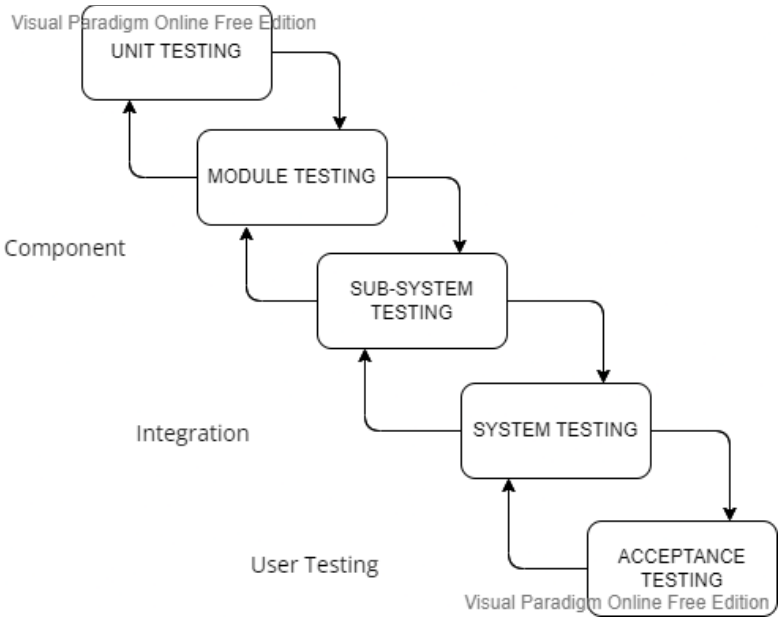


Figure: 7.1 Testing Plan

## 7.2 SOFTWARE TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 7.2.1 TESTING METHODOLOGIES

The following are the Testing Methodologies:

- **Unit Testing.**
- **Integration Testing.**
- **User Acceptance Testing.**
- **Output Testing.**
- **Validation Testing.**
- **Black box Testing.**

- **White box Testing.**

### 7.2.2 Unit Testing

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing. During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

### 7.2.3 Integration Testing

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

### 7.2.4 User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

### 7.2.5 Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

### 7.2.6 Validation Testing

Validation checks are performed on the following fields.

### Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

### Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modules are integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

### Preparation of Test Data

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

### 7.1.6 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.

### 7.1.7 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.

## TESTING STRATEGY:

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

## 7.3 TEST CASES

| Test Case Id | Test Case Description | Expected Results | Actual Result | Pass/Fail |
|---|---|---|---|---|
| 01 | User Interface | User should able to upload and view the result images on the system | Users are able to interact with UI and also able to upload and view the output images | Pass |
| 02 | Image Browsing | The User should able to | The User is able to browse and | Pass |

| | | | | |
|---|---|---|---|---|
| | | browser and upload the image | upload the image | |
| 03 | Image Classify | Once if the user clicks the classify button he should be able to view the classified output | The user is able to hit the classify button and view the classified button. | Pass |

Table 7.1 : Test Cases for User Interface

| Test ID | Description | Expected Output | Actual Output | Result |
|---|---|---|---|---|
| 01 | User Interface Testing | User should able to upload and view the result images on the system | Users are able to interact with UI and also able to upload and view the output images | Pass |
| 02 | Images with proper lights and visibility | MCLRNF1 | MCLRNF1 | Pass |
| 03 | An Image in which the text is displayed brightly. | RIPLS1 | RIPLS1 | Pass |
| 04 | An image with improper edges and lights | NVSBLE | NVSBE | Fail |

Table: 7.2 Test Cases for Algorithm

## 7.4 BUG REPORT

| S.no | Steps | Description |
|---|---|---|
| 1. | **Bug_id** | Bug_1 |
| 2. | **Bug Description** | Invalid Image |
| 3. | **Expected output** | Proper Detected Text |
| 4. | **Actual output** | Invalid Image |
| 5. | **Priority** | High |
| 6. | **Severity** | High |

Table: 7.3 Bug Report - 1

| S.no | Steps | Description |
|---|---|---|
| 1. | **Bug_id** | Bug_2 |
| 2. | **Bug Description** | Could not able to Upload Image |
| 3. | **Expected output** | Visible of Image on the Screen |
| 4. | **Actual output** | Image is not visible on the screen |
| 5. | **Priority** | High |
| 6. | **Severity** | High |

Table: 7.4 Bug Report - 2

| S.no | Steps | Description |
|---|---|---|
| 1. | **Bug_id** | Bug_3 |
| 2. | **Bug Description** | Could not able to Load the k-NN the model |
| 3. | **Expected output** | Proper Image Detection |
| 4. | **Actual output** | k-NN model could not able to detect the image. |
| 5. | **Priority** | High |
| 6. | **Severity** | High |

Table: 7.5 Bug Report - 3

# RESULT SCREENS

# 8. RESULT SCREENS

## User Interface Page - 1


Figure: 8.1 User Interface Page

## Navigation:

- To run the project first we need to activate the python environment by executing ".\env\Scripts\activate" in the command prompt.
- Next, we need to execute "python apptorun.py" command to execute the python file.
- Then the UI gets opened with two buttons one is "Browse" and another is "Classify".
- We can use Browse button to browse for the image.

## User Interface Page - 2

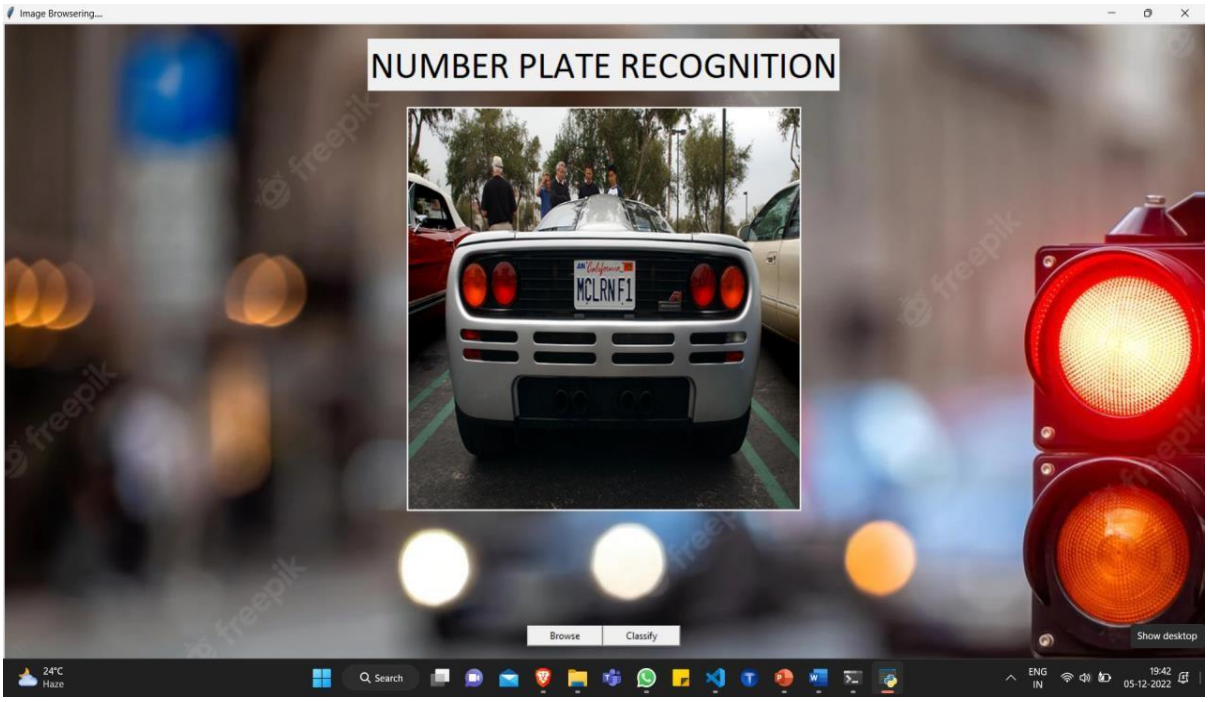


Figure: 8.2 User Interface Page

## Navigation:

- Once we could browse the image, the selected image gets displayed upon the screen.
- Next, We need to click upon classify button to view the output.

## User Interface Page - 3

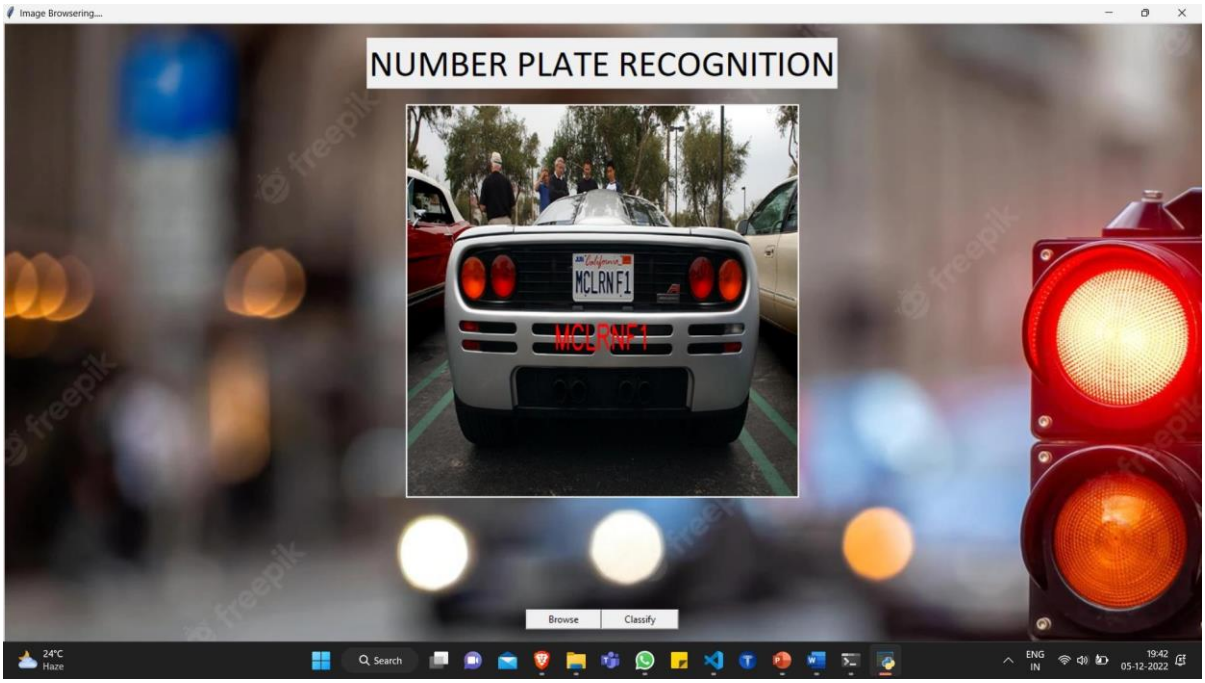


Figure: 8.3 User Interface Page

## Navigation:

- Once, the user clicks the classify button.
- An image gets replaced with the image that has text on it.

# CONCLUSION & FUTURE SCOPE

# 9. CONCLUSION AND FUTURE SCOPE

The automatic vehicle and number plate detection system plays an important role in detecting security threat. In this project, I aimed to develop a Number Plate Recognition System using KNN. We got images that has no noise from internet for testing purpose. And We also used different styles of text to train the k-NN model for classification.

The image first goes into the few processing techniques such has Gray-scaling and edge detection. After that the images goes into segmentation phase and those segmented characters are sent for text classification. As the k-NN algorithm has been trained it will try to perform text classification on them.

The k-nearest neighbor algorithm was employed to identify number plates. The number plate detection accuracy of the system is 82.75%. It can be said that the real-time k-NN plate recognition system worked well.

A hybrid classifier that can combine different classifiers and train it on a wider variety of dataset to increase the accuracy. A system where it can identify the no-helmet or Triple riders and send the challan direct to there phone.

By positioning the camera to capture the ideal body and using neural networks in two layers, this accuracy can be significantly increased. As many scholars like to research in this field, they can build much advanced techniques for detection and recognition which may increase the accuracy.

# BIBILOGRAPHY

# 10. REFERENCES


[1] M. S. Uddin, A. K. Das and M. A. Taleb, "Real-time Area Based Traffic Density Estimation by Image Processing for Traffic Signal Control System: Bangladesh Perspective," in 2nd Int'l Conf. on Electrical Engineering and Information & Communication Technology, Dhaka, 2015.

[2] B. F. Momin and T. M. Mujawar, "Vehicle detection and Attribute based search of vehicles in video surveillance system," in International Conference on Circuit, Power and Computing Technologies, Nagercoil, 2015.

[3] A. R. F. Quiros, A. Abad, A. C. P. Uy, A. Bandala, R. A. Bedruz, E. Sybingco and E. P. Dadios, "Automated Traffic Violation Apprehension System Using Genetic Algorithm and Artificial Neural Network," in TENCON 2016 - 2016 IEEE Region 10 Conference, Singapore, 2016.

[4] Sai Jahnavi Bachu, B. Tech in K L University, "Character Recognition using KNN Algorithm" International Journal of Science and Research (IJSR),ISSN: 2319-7064,SJIF (2019): 7.583

[5] M. Kumar, M. K. Jindal and R. K. Sharma, "k-nearest neighbour based offline handwritten Gurmukhi character recognition," in ernational Conference on Image Information Processing,Himachal Pradesh, 2011.

[6] Qiu D, Lu J, Sun X and Jiang J 2010 Application of graphic minimal covering algorithm in the distribution of surveillance cameras in small and medium-sized city road networks 2010 International Conference On Computer Design and Applications vol 1 pp V1-200-V1-203

[7] Islam R, Sharif K F and Biswas S 2015 Automatic vehicle number plate recognition using structured elements 2015 IEEE Conference on Systems, Process and Control (ICSPC) pp 44-48

[8] Pradeepa, J., E. Srinivasana & S. Himavathib. Neural Network based Recognition System Integrating Feature Extraction and Classification for English Handwritten. International Journal of Engineering 25: 99–106(2012).

[9] Y. LeCun, et al., Comparison of learning algorithms for handwritten digit recognition, in: F. Fogelman-Souli e, P. Gallinari (Eds.), Proceedings of the

International Conference on Artificial Neural Networks, Nanterre, France, 1995, pp. 53–60.

[10] Object Recognition Using K-Nearest Neighbor Supported By Eigen Value Generated From the Features of an Image, Dr. R.Muralidharan

[11] "Handwritten Digit Recognition Using K-Nearest Neighbour Classifier": U Ravi Babu, Dr. Y Venkateswarlu, Aneel Kumar Chintha.

[12] Kulkarni P, Khatri A, Banga P and Shah K 2009 Automatic number plate recognition (anpr) system for indian conditions 2009 19th International Conference Radioelektronika pp 111-114

# APPENDIX

# 11. APPENDIXES

## 11.1 SAMPLE CODE

### //apptorun.py:

```
import glob
import Main
import os
import cv2
from tkinter import *
from tkinter import filedialog
import os
import tkinter as tk
from PIL import Image, ImageTk
def showimage():
   fln = filedialog.askopenfilename(initialdir=os.getcwd(), title="Select Image File",
filetypes=(("All Files", "*.*"), ("PNG File", "*.png"),("JPG File", "*.jpg")))
   global txt
   txt = Main.main(fln)
   img = Image.open(fln)
   img = img.resize((500,500))
   img = ImageTk.PhotoImage(img)
   lbl.image = img
   lbl.configure(image=img)

def classify():
   img =
Image.open("C:/Users/Tarun/OneDrive/Desktop/Numberplate/SavedImg/R1.png").resize
((500,500))
```

```
    img = ImageTk.PhotoImage(img)
    lbl.configure(image=img)
    lbl.image = img
    print("--",txt)
    lbl.configure(text=txt)
root = Tk()
root.geometry("500x300")
image_0 =
Image.open("C:/Users/Tarun/OneDrive/Desktop/Numberplate/LicPlateImages/wallpaper.
webp")
bck_end = ImageTk.PhotoImage(image_0)
lbl = Label(root,image=bck_end)
lbl.place(x=0,y=0)
lbl=Label(root, text="NUMBER PLATE RECOGNITION", font=('Calibri 36'))
lbl.pack(pady=20)
frm = Frame(root,bg="#88cffa")
frm.pack(side=BOTTOM, padx=15, pady = 15)
lbl = Label(root)
lbl.pack()
btn = Button(frm,text = "       Browse       ", command=showimage)
btn.pack(side=tk.LEFT)
btn = Button(frm, text = "       Classify       ", command=classify)
btn.pack(side=tk.RIGHT)
root.title("Image Browsering.... ")
root.mainloop()
```

## // Main.py

```
import cv2
import numpy as np
import os
import DetectChars
```

```
import DetectPlates
import PossiblePlate

SCALAR_BLACK = (0.0, 0.0, 0.0)
SCALAR_WHITE = (255.0, 255.0, 255.0)
SCALAR_YELLOW = (0.0, 255.0, 255.0)
SCALAR_GREEN = (0.0, 255.0, 0.0)
SCALAR_RED = (0.0, 0.0, 255.0)

showSteps = False
def main(img_path):

    blnKNNTrainingSuccessful = DetectChars.loadKNNDataAndTrainKNN()         # attempt KNN training

    if blnKNNTrainingSuccessful == False:                               # if KNN training was not successful
        print("\nerror: KNN traning was not successful\n")  # show error message
        return                                                  # and exit program
    # end if

    ###### IMAGE PATH
    imgOriginalScene  =  cv2.imread(img_path)           # open image

    if imgOriginalScene is None:                            # if image was not read successfully
        print("\nerror: image not read from file \n\n") # print error message to std out
        os.system("pause")                                  # pause so user can see error message
        return                                              # and exit program
    # end if
```

```
    listOfPossiblePlates = DetectPlates.detectPlatesInScene(imgOriginalScene)           # detect plates

    listOfPossiblePlates = DetectChars.detectCharsInPlates(listOfPossiblePlates)        # detect chars in plates

    #cv2.imshow("imgOriginalScene", imgOriginalScene)          # show scene image

    if len(listOfPossiblePlates) == 0:                      # if no plates were found
        print("\nno license plates were detected\n")  # inform user no plates were found
    else:                                               # else
            # if we get in here list of possible plates has at leat one plate

            # sort the list of possible plates in DESCENDING order (most number of chars to least number of chars)
        listOfPossiblePlates.sort(key = lambda possiblePlate: len(possiblePlate.strChars), reverse = True)

            # suppose the plate with the most recognized chars (the first plate in sorted by string length descending order) is the actual plate
        licPlate = listOfPossiblePlates[0]

        #cv2.imshow("imgPlate", licPlate.imgPlate)           # show crop of plate and threshold of plate
        #cv2.imshow("imgThresh", licPlate.imgThresh)

        if len(licPlate.strChars) == 0:                 # if no chars were found in the plate
            print("\nno characters were detected\n\n")  # show message
            return                                  # and exit program
        # end if
```

```
        print("\nlicense plate read from image = " + licPlate.strChars + "\n") # write license plate text to std out
        txt = licPlate.strChars
        print("----------------------------------------")

        writeLicensePlateCharsOnImage(imgOriginalScene, licPlate)           # write license plate text on the image

        # Using resizeWindow()
        #cv2.resizeWindow("imgOriginalScene", 1000, 900)

        #cv2.imshow("imgOriginalScdddene", imgOriginalScene)              # re-show scene image

        cv2.imwrite("C:/Users/Tarun/OneDrive/Desktop/Numberplate/SavedImg/R1.png", imgOriginalScene)          # write image out to file

    # end if else

    cv2.waitKey(0)                                      # hold windows open until user presses a key

    return txt
# end main

def writeLicensePlateCharsOnImage(imgOriginalScene, licPlate):
    ptCenterOfTextAreaX = 0                         # this will be the center of the area the text will be written to
    ptCenterOfTextAreaY = 0
```

```
    ptLowerLeftTextOriginX = 0                  # this will be the bottom left of the area
that the text will be written to
    ptLowerLeftTextOriginY = 0

    sceneHeight, sceneWidth, sceneNumChannels = imgOriginalScene.shape
    plateHeight, plateWidth, plateNumChannels = licPlate.imgPlate.shape

    intFontFace = cv2.FONT_HERSHEY_SIMPLEX                  # choose a plain jane
font
    fltFontScale = float(plateHeight) / 30.0                # base font scale on height of plate
area
    intFontThickness = int(round(fltFontScale * 1.5))       # base font thickness on font
scale

    textSize, baseline = cv2.getTextSize(licPlate.strChars, intFontFace, fltFontScale,
intFontThickness)       # call getTextSize

        # unpack roatated rect into center point, width and height, and angle
    (    (intPlateCenterX, intPlateCenterY), (intPlateWidth, intPlateHeight),
fltCorrectionAngleInDeg ) = licPlate.rrLocationOfPlateInScene

    intPlateCenterX = int(intPlateCenterX)          # make sure center is an integer
    intPlateCenterY = int(intPlateCenterY)

    ptCenterOfTextAreaX = int(intPlateCenterX)      # the horizontal location of the text
area is the same as the plate

    if intPlateCenterY < (sceneHeight * 0.75):                                  # if the
license plate is in the upper 3/4 of the image
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) + int(round(plateHeight * 1.6))
# write the chars in below the plate
```

```
    else:                                               # else if the license plate is
in the lower 1/4 of the image
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) - int(round(plateHeight * 1.6))
# write the chars in above the plate
    # end if

    textSizeWidth, textSizeHeight = textSize                # unpack text size width and height

    ptLowerLeftTextOriginX = int(ptCenterOfTextAreaX - (textSizeWidth / 2))           #
calculate the lower left origin of the text area
    ptLowerLeftTextOriginY = int(ptCenterOfTextAreaY + (textSizeHeight / 2))          #
based on the text area center, width, and height

            # write the text on the image
    cv2.putText(imgOriginalScene, licPlate.strChars, (ptLowerLeftTextOriginX,
ptLowerLeftTextOriginY), intFontFace, fltFontScale, SCALAR_RED, intFontThickness)
# end function
```

## // Preprocesser.py

```
import cv2
import numpy as np
import math

GAUSSIAN_SMOOTH_FILTER_SIZE = (5, 5)
ADAPTIVE_THRESH_BLOCK_SIZE = 19
ADAPTIVE_THRESH_WEIGHT = 9

def preprocess(imgOriginal):
    imgGrayscale = extractValue(imgOriginal)
```

```
    imgMaxContrastGrayscale = maximizeContrast(imgGrayscale)

    height, width = imgGrayscale.shape

    imgBlurred = np.zeros((height, width, 1), np.uint8)

    imgBlurred = cv2.GaussianBlur(imgMaxContrastGrayscale, GAUSSIAN_SMOOTH_FILTER_SIZE, 0)

    imgThresh = cv2.adaptiveThreshold(imgBlurred, 255.0, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, ADAPTIVE_THRESH_BLOCK_SIZE, ADAPTIVE_THRESH_WEIGHT)

    return imgGrayscale, imgThresh
# end function

def extractValue(imgOriginal):
    height, width, numChannels = imgOriginal.shape

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

    imgHSV = cv2.cvtColor(imgOriginal, cv2.COLOR_BGR2HSV)

    imgHue, imgSaturation, imgValue = cv2.split(imgHSV)

    return imgValue
# end function
def maximizeContrast(imgGrayscale):

    height, width = imgGrayscale.shape
```

```
    imgTopHat = np.zeros((height, width, 1), np.uint8)
    imgBlackHat = np.zeros((height, width, 1), np.uint8)

    structuringElement = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))

    imgTopHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_TOPHAT, structuringElement)
    imgBlackHat = cv2.morphologyEx(imgGrayscale, cv2.MORPH_BLACKHAT, structuringElement)

    imgGrayscalePlusTopHat = cv2.add(imgGrayscale, imgTopHat)
    imgGrayscalePlusTopHatMinusBlackHat = cv2.subtract(imgGrayscalePlusTopHat, imgBlackHat)

    return imgGrayscalePlusTopHatMinusBlackHat
# end function
```

**// Detectchars.py**

```
import cv2
import numpy as np
import math
import Main
import random

import Preprocess
import DetectChars
import PossiblePlate
import PossibleChar
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()

    imgGrayscaleScene, imgThreshScene = Preprocess.preprocess(imgOriginalScene)
# preprocess to get grayscale and threshold images
    #cv2.imshow("ImgGrayscaleScene",imgGrayscaleScene)
    #cv2.imshow("ImgThersholdScene",imgThreshScene)

    if Main.showSteps == True: # show steps
        cv2.imshow("1a", imgGrayscaleScene)
        cv2.imshow("1b", imgThreshScene)

    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)

            # 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:
        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)
    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)

    return listOfPossiblePlates
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:
```

```
cv2.drawContours(imgContours, contours, i, Main.SCALAR_WHITE)

        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:
        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)
    return listOfPossibleChars



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)

    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 )
    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))

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

    possiblePlate.imgPlate = imgCropped

    return possiblePlate
```