CHAPTER 1

INTRODUCTION

No-Parking Vehicle Detection (NPVD) system has been a practical technique in the past decades. One type of intelligent transportation system (ITS) technology is the automatic number plate recognition which can distinguish each vehicle as unique by recognizing the characters of the number plates. Automatic number plate recognition system finds wide varieties of applications to fit itself beyond just controlling access to collect details of vehicle parked in no parking areas. In NPVD, a camera captures the vehicle images and a computer processes them and recognizes the information on the number plate by applying various image processing and optical character recognition techniques. Prior to the character recognition, the number plates must be separated from the background vehicle images. This task is considered as the most crucial step in the ANPR system, which influences the overall accuracy and processing speed of the whole system significantly. Since there are problems such as poor image quality, image perspective distortion, other disturbance characters or reflection on vehicle surface, and the color similarity between the number plate and the background vehicle body, the number plate is often difficult to be located accurately and efficiently.

Generally vehicle number plate recognition is divided into several steps including number plate extraction, image region which contains a number plate, character segmentation, and character recognition. Generally, in order to recognize a vehicle number plate, the region of the number plate should be extracted from a vehicle image. Accurate detection of the plate region is essential process to go over to the step of character recognition.

There are two major methods to extract number plate region,

- Edge Detection
- Finding Rectangles in a Vehicle Image

1. STATEMENT OF A PROBLEM

With decreasing costs of high quality surveillance systems, human activity detection and tracking has become increasingly practical. Accordingly, automated systems have been designed for numerous detection tasks, but the task of detecting illegally parked vehicles has been left largely to the human operators of surveillance systems. We propose NPVD for detecting this event in realtime by applying a novel image processing system that can perform the job quite easily and efficiently. After event detection, we extract the number plate (or otherwise called license plate) data for further processing. The proposed program is able to successfully recognize illegally parked vehicles in real-time and impose fine as per the traffic law in india.

CHAPTER 2

SYSTEM ANALYSIS

1.Present system

The problem that countries like India that are still trying to deal with is the uncontrollable traffic rule breaking. Most of the problem fails to find a solution beacause of slow and manual actions that authorities takes when someone breaks the rule. Vehicles that are parked in no parking area often didn't get any legal actions due to reasons like unability to watchout for local no parking areas and unavailability of that much of human working force. This results in many conditions like traffic jam, public nuisance etc. Thus the traditional present system in inefficient and non reliable.

2.Limitation of present system

- Manual process
- Time consuming
- Events may get unreported while authorities are abscent at the time of it's occurrence
- More human effforts needed

3. Proposed system

NPVD system is proposed for monitoring and imposing fine to vehicles that are parked in non-parking area via identifying vehicle license plate numbers. No additional equipment need to be installed for operating this system. The only requirement of this system is installing special cameras for identifying license numbers on the no-parking area. The images taken by these cameras are subsequently processed in a computer. The cameras used in the system can be deployed under all weather conditions and are equipped with powerful infrared radiation units for identifying vehicle license plates in absolute darkness. The system normally comprises a camera for monitoring the vehicle path, an identification system for recognizing license plate number and are used for further identification of the corresponding owner to impose fines for breaking the traffic rules.

The software program used within the system deliver high precision and provide great processing speeds and fully reliable system. The OpenCV library provide a great image processing engine that ensures powerful and precise processing capabilities. The system prepares reports of vehicle that are parked in terms of time of entry and departure, vehicle license number, duration of each vehicle's stay in the area , amount of fine imposed etc. The user interface of the system is designed for speedy access to system events

4. Advantages and Features of Proposed System

- The system can detect vehicle immediately
- Less human intervention needed
- .Fast, reliable and secure
- Less paper work needed
- Instant fine imposing

3. Feasibility Study

A feasibility study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as technical, economic, behavioural, operational factors. When a new project is proposed, it normally goes through the feasibility assessment. Feasibility Study is carried out to determine whether the proposed system is possible to develop with available resources & what should be the cost of consideration.

Various types of feasibilities are,

- Technical Feasibility
- Economic Feasibility
- Operational Feasibility

If the proposed system is not feasible to develop, it is rejected at this very step.

1. Technical Feasibility

The proposed system uses the language Python. Based on this criteria, we can strongly say that it is technically feasible, since there will not be much difficulty in getting required resources for the development & maintaining system as well. All the resources needed for the development of the software as well as the maintenance of the same is available in the organization. Here we are utilizing the resources which are already available so it's very well technically feasible that we can implement flood detection system.

2. Economic Feasibility

It is found that the benefit from our system would be more than the cost and time involved in its development. In our system the implementation cost over production is economically feasible. Economic analysis is the most frequently used techniques for evaluating the effectiveness of the proposed system more commonly known as cost/benefit analysis the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs.

3. Operational Feasibility

The proposed system satisfies operational feasibility in the way that the customers needs are satisfied. The system is adaptable to the customers and acceptable to the common people who use this. Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks:

- Determines whether the problems anticipated in user requirements are of high priority
- Determines whether the solution suggested by the software development team is acceptable
- Analyses whether users will adapt to a new software
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

CHAPTER 3

SYSTEM SPECIFICATION

On the system specification the analyst begins to learn about the present system and physical process related to the revised system. After obtaining the information, software engineer begins to collect data on the present system outputs, inputs and costs. In this phase a key question is, what are the user's needs and how does a candidate system meet them.In systems engineering and software engineering, requirement analysis encompasses those tasks that go into determining the requirements of a new or altered system. Requirement analysis is critical to the success of a project. Systematic requirements analysis is also known as requirement engineering. It is sometimes referred to loosely by names such as requirements gathering, requirements capture, or requirements specification. The term "requirement analysis" can also be applied to the analysis proper.

Requirements must be measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. Conceptually, requirement analysis includes three types of activity:

- Eliciting requirements: The task of communicating with customers and users to determine what their requirements are.
- Analysing requirements: Determining whether the stated requirements are unclear, incomplete, ambiguous or contradictory and then resolving these issues.
- Recording requirements: Requirements may be documented in various forms, such as natural-language documents, use cases, user stories or process specifications.

3.1 Software Requirements

1. Operating System : Windows 8/10
2. Language : Python
4. IDE : Spyder3
5. Libraries : Open CV, Numpy, tKinter

3.2 Hardware Requirements

1. Processor : Intel i5
2. RAM : 3 GB
3. Hard Disk Drive : 200 GB
4. Peripherals : Keyboard, Mouse, Monitor, Camera

CHAPTER 4

SYSTEM DESIGN

System design is the process or art of defining the hardware and software architecture, components, modules, interfaces, and data for a computer system to satisfy specified requirements. Software design is a process of problem –solving and planning for a software solution. After the purpose and specifications of software is determined, software developers will design or employ designers to develop a plan for a solution. The software requirements analysis step of a software development process yields specifications that are used in software engineering.

System design is a multi-step process that focuses on data structure, software architecture procedural details and interface between modules. Computer software design changes continually as a new method, better analysis and boarder understanding evolve. System design is a solution “How to” approach to the creation of a new system. This is important phase composed of several steps. It provides understanding procedure details necessary for complementing the system recommended in the feasibility study. Here the emphasis on translating the requirements into abstraction, structure information hiding, modularity, concurrency, and verification and design aesthetics.

CHAPTER 4

SYSTEM DESIGN

Data Flow Diagram

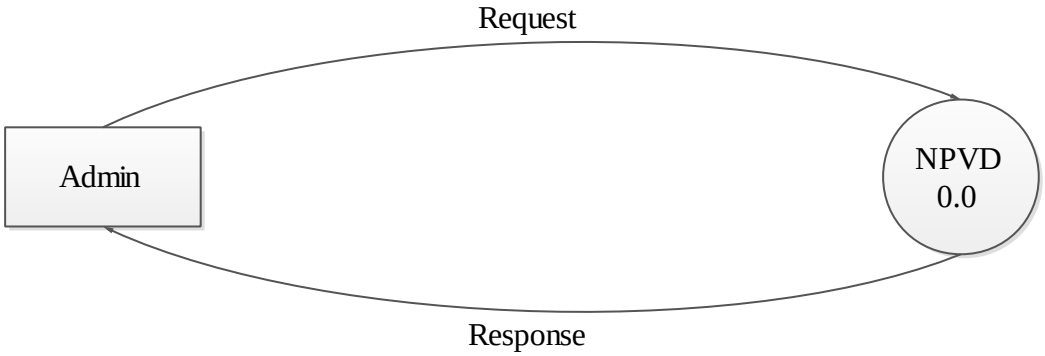


Fig 4.1 Context Level Diagram

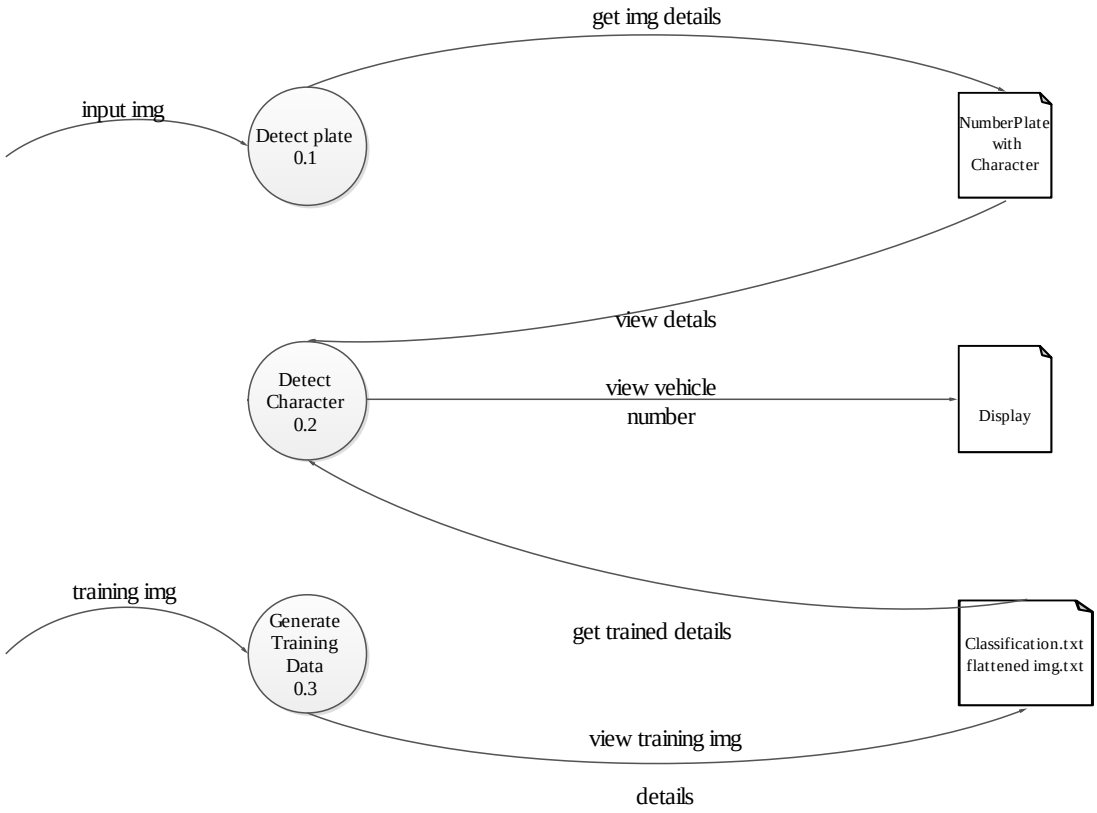


Fig 4.2 Level 1 DataFlow Diagram

Data Flow Diagram

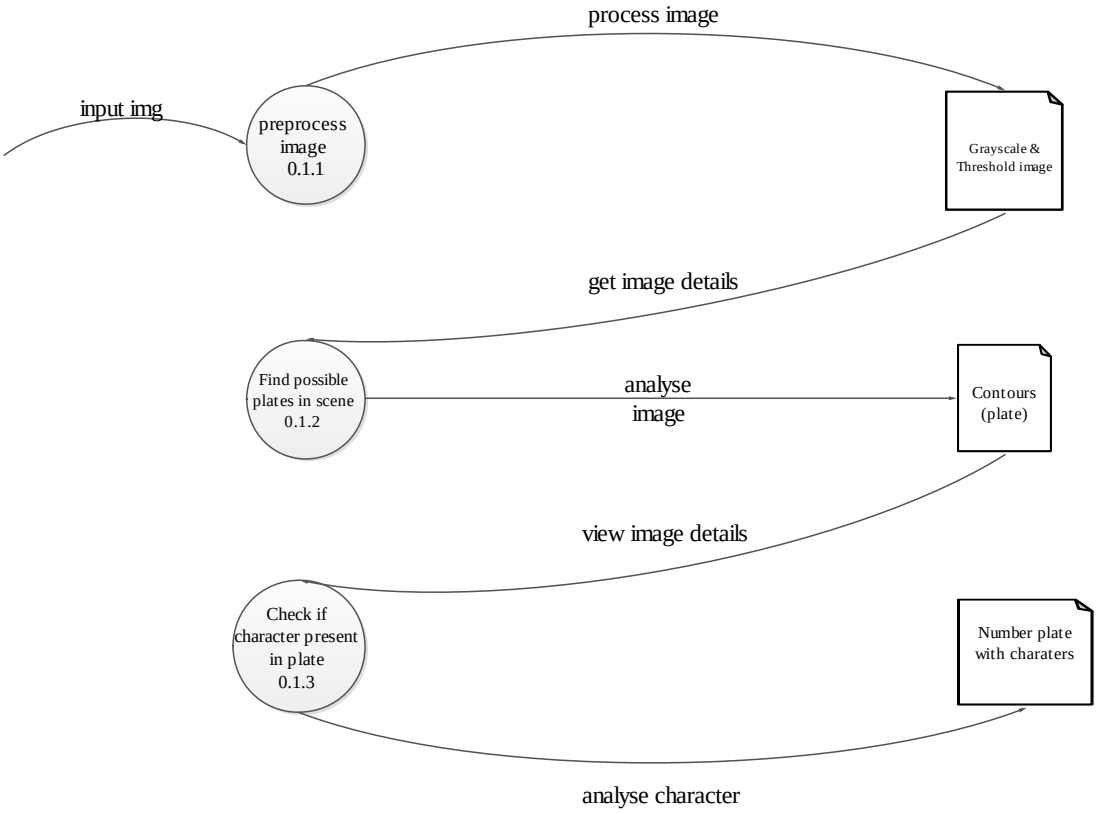


Fig 4.3 Level 2 DataFlow Diagram

Data Flow Diagram

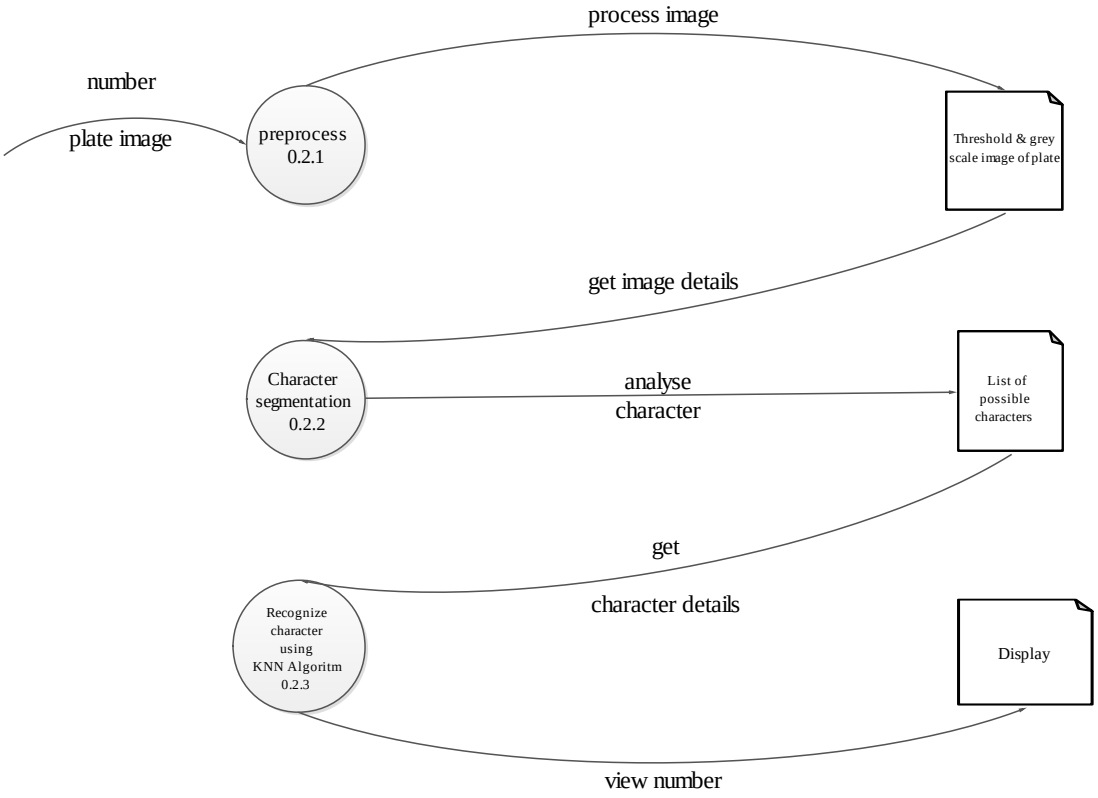


Fig 4.4 Level 2 DataFlow Diagram

Data Flow Diagram

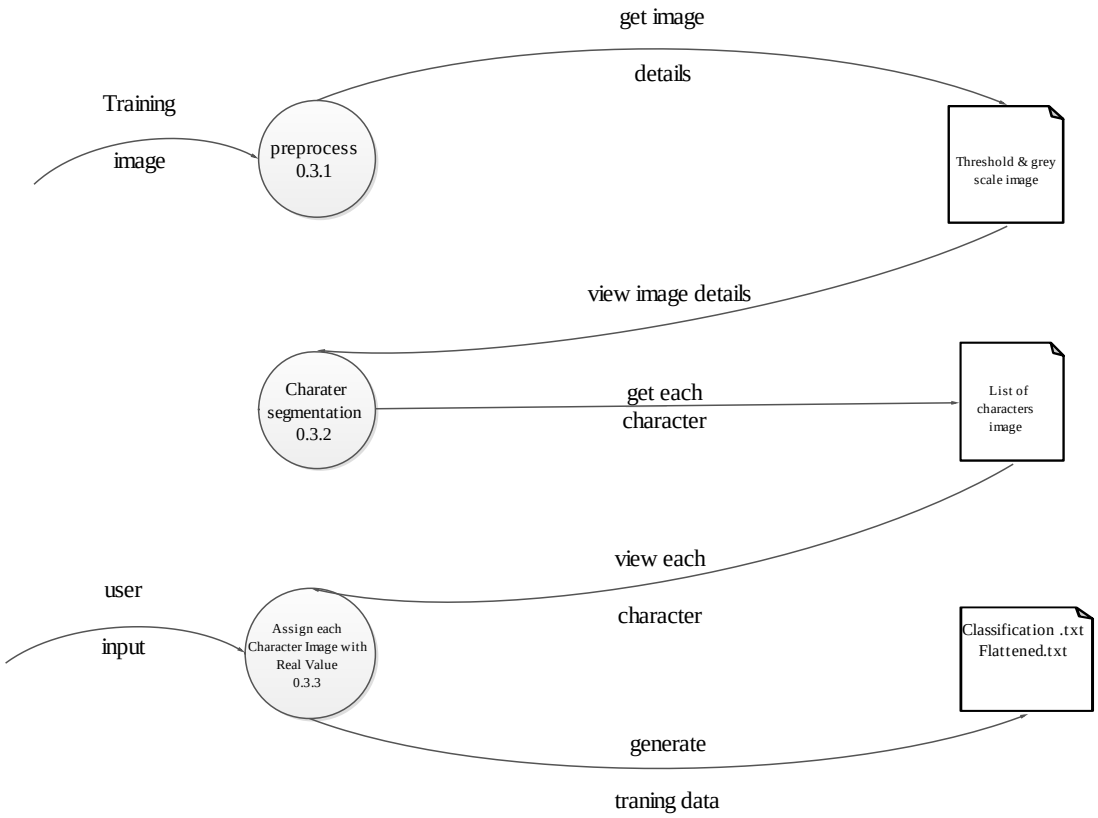
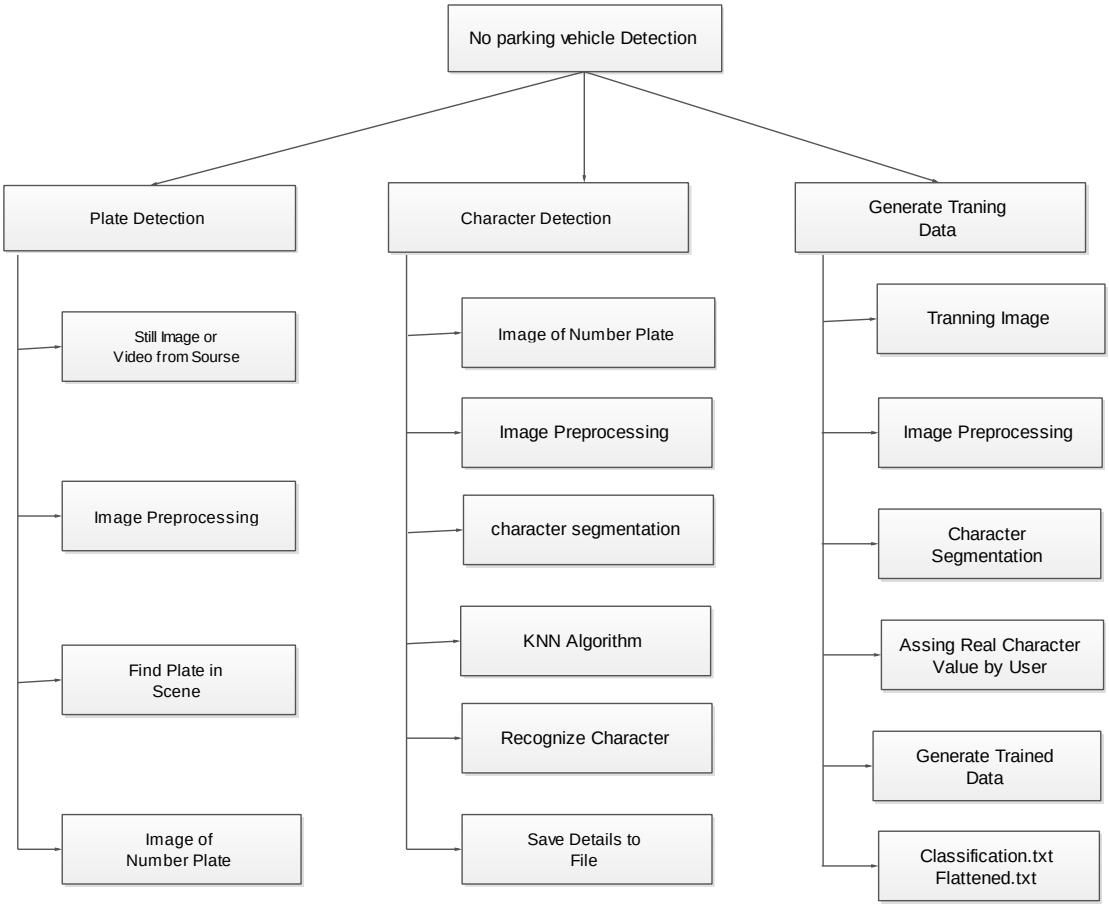


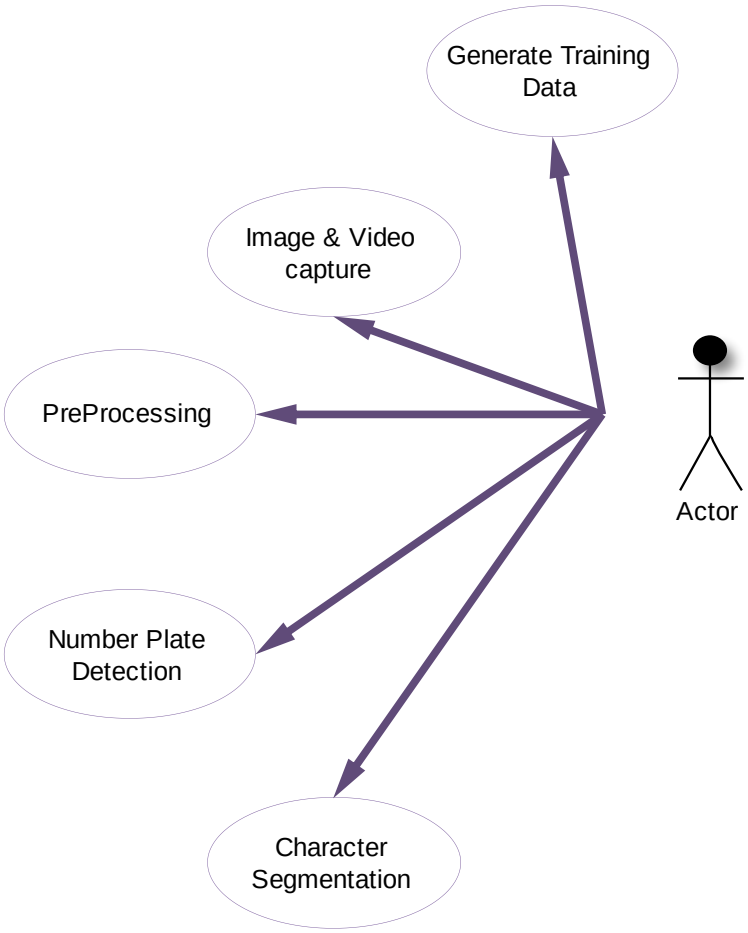
Fig 4.5 Level 2 DataFlow Diagram

Design of Each Subsystem

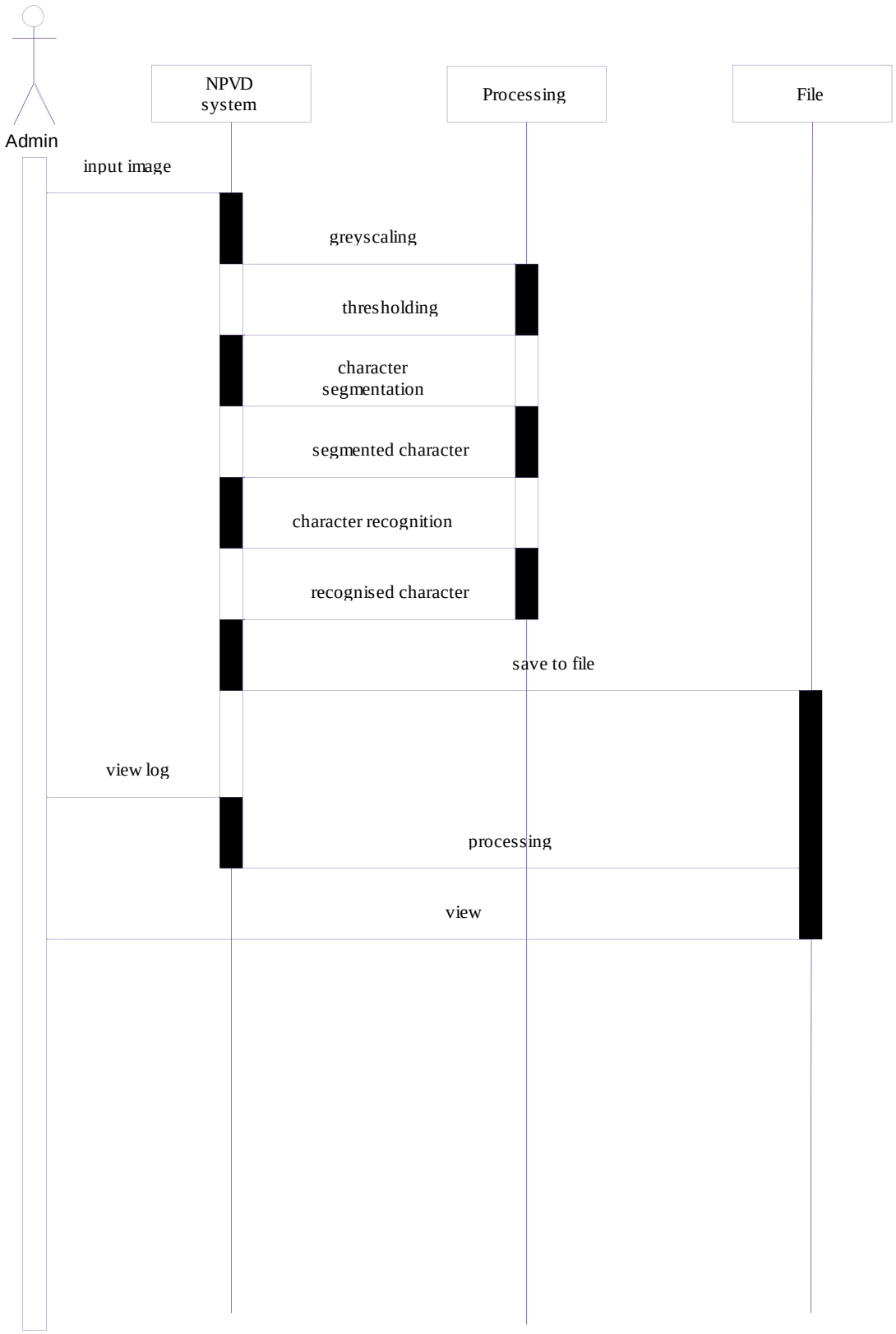


4.4 UML Diagram

4.4.1 Use case Diagram



4.4.2 Sequence Diagram



CHAPTER 5

CODING

5.1 Features of Language

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Below are some facts about Python

1. Python is a widely used general-purpose, high-level programming language.
2. Python allows programming in Object-Oriented and Procedural paradigms.
3. Python programs generally are smaller than other programming languages like Java. Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.
4. Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox and Uber… etc.
5. The biggest strength of the Python is large library which can be used for the following
    - Machine Learning
    - GUI Applications (like Kivy, Tkinter and PyQt etc.)
    - Web frameworks like Django (used by YouTube, Instagram, Dropbox)
    - Image processing (like OpenCV, Pillow)
    - Web scraping (like Scrapy, BeautifulSoup, Selenium)
    - Test frameworks
    - Multimedia
    - Scientific computing
    - Text processing and many more...

Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

OpenCV

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 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.

Functional Description

detectPlatesInScene():

#detect number plate in image.

detectCharsInPlates():

#detect number from the plate

preprocess():

#imgGrayscaleScene, imgThreshScene

findPossibleCharsInScene():

#listOfPossibleCharsInScene

findListOfListsOfMatchingChars():

#listOfListsOfMatchingCharsInScene

extractPlate():

#listOfPossiblePlates

loadKNNDataAndTrainKNN():

#list Of Possible Plates checked with training data

preprocess():

#possiblePlate.imgGrayscale, possiblePlate.imgThresh

findPossibleCharsInPlate():

#listOfPossibleCharsInPlate

findListOfListsOfMatchingChars():

#listOfListsOfMatchingCharsInPlate

removeInnerOverlappingChars():

#listOfListsOfMatchingCharsInPlate

recognizeCharsInPlate():

#character segmentation

CHAPTER 6
TESTING

System testing is the major quality control measure during software development. Testing is a set of activities that can be planned and conducted schematically. Testing begins at the module level and work towards the integration of entire computer based system. Testing is a process of executing a program with the intention of finding an error. A good test case is one that has a higher probability of finding an undiscovered error. A successful test case is one that uncovers an undiscovered error. Testing phase in the Smart Purchasing is supposed to verify that the system does exactly what it is designed to do. The Smart Purchasing is tested with the data at the extremes of the input range. This system is also being tested for various values outside the input range. The system provides different validity test strategies to validate the textboxes and entries in the system. Also it checks the system efficiency in terms of their input and output data's.

Test Procedure

Software testing accounts for the largest percentage of technical effort in the software process. The objective of the software testing is to uncover errors. To fulfil this objective, a series of test steps unit, integration, validation and system tests are planned and executed. In this system, we can adopt various types of test strategies .These are checks such as the validity, accuracy of the data etc.

System Testing

System testing validates the Data Analyser once it has been incorporated into a large system. System testing is actually a series of different tests whose primary purpose is fully exercising the computer based system. All work to verify that Data Analyser elements have been properly integrated and perform allocated function. They can checks the functioning of processes with respect to their input data. Also the Data Analyser that test the system validity in a user friendly manner.

Unit Testing:

Unit testing means after complete each module to test the module. Unit Testing can be performed on the modules such that User details.

Integration testing:

Integration Testing means set of components interaction between the modules. In this system perform the integration testing on the one module to another module.

Test case and Output

The test case is a document that describes an input, action or event and an expected response, to determine if a feature of an application is working correctly. A test case should contain particulars such as test case identifiers, test case name, objectives, test conditions, input data requirements steps and expected results.

CHAPTER 7

IMPLEMENTATION

Implementation is the stage of the project where theoretical design is turned into a working system. If the implementation is not carefully planned and controlled, it can cause chaos and confusion. Proper implementation is essential to provide a reliable system to meet organization requirements. Successful implementation may not guarantee improvement in the organization using the system, but proper installation will prevent it. The process of putting the developed system in actual use is called system implementation. The system can only be implemented after thorough testing is done and if it is found to be working according to the specifications.

The implementation stage involves following tasks:

1. Careful Planning
2. Investigation of system and constraints.
3. Design of methods to achieve the changeover.
4. Training of the staff in the changeover phase.
5. Evaluation of the changeover method.

To host the Data Analyser system, the primary need is web based environments without which the system will not have a proper utilization. To install the system, it is must to setup a centralized server which can hold social networking website including the user's information database. The database is accessed through web pages using browser at the client end. In order to have the server setup for the device information system, the following components are needed at the server end.

1) Python Tool
2) A preferable operating system like windows 8 or windows 10
3) Spyder3

Parallel run is done and both the computerized and manual systems are executed in parallel manual result can be compared with the result of computerized system. For the case of demonstration of the success of this system, it was implemented with successfully running; manual systems results are verified.

CHAPTER 8

SECURITY, BACKUP AND RECOVERY MECHANISMS

Security is an important consideration in desktop application. The first step in securing our application is deciding where we need security and what will be needed to protect.

Security concepts:

1. Authentication
2. Authorization

Authentication

This is the process of determining users identify and forcing users to prove they are who they claim to be usually this involves entering username and password in login page.

Authorization

Once the user is authenticated, authorization is the process of determining whether the user has sufficient permission to perform a given action or not, such as viewing a page or retrieving information from the database.

8.1 User Manual

The user manual provides the detailed description regarding the usage of the software.

The main user tips are:

1. Never share your username and password.
2. Do not write your username and password down in an unsecured environment.
3. Do not send extremely confidential information as autographs to any user.
4. Change your password periodically.
5. If your browser prompts to save username and password, cancel as it is not safe to store your log on information in your browser.

Log in for the first time, please follow the steps below:

1. Go the user log on page of the website and click on the hyperlink “Register”
2. Enter a email and password and the other details are specified on the New Registration page.
3. The user database is checked to see whether the email entered is unique and the password has been confirmed.
4. Enter all the fields that are given as optional fields.

CHAPTER 9

CONCLUSION

Improvements in the performance of visual recognition systems in the past decade have in part come from the realization that finely sampled pyramids of image features provide a good front-end for image analysis. It is widely believed that the price to be paid for improved performance is sharply increased computational costs. It have shown that this is not necessarily so. Finely sampled pyramids may be obtained inexpensively by extrapolation from coarsely sampled ones. This insight decreases computational cost substantially.

CHAPTER 10

FUTURE ENHANCEMENT

Our project has a very vast scope in the future. The model may be expanded to use web services. In future mobile users downloading applications called apps from the web may also be entitled to such services and this enhancement may further reduce memory overheads and improve accuracy than the ones used at present. This enhancement will make it a better utility for mobile users as well in future

APPENDIX

Input and Output Forms

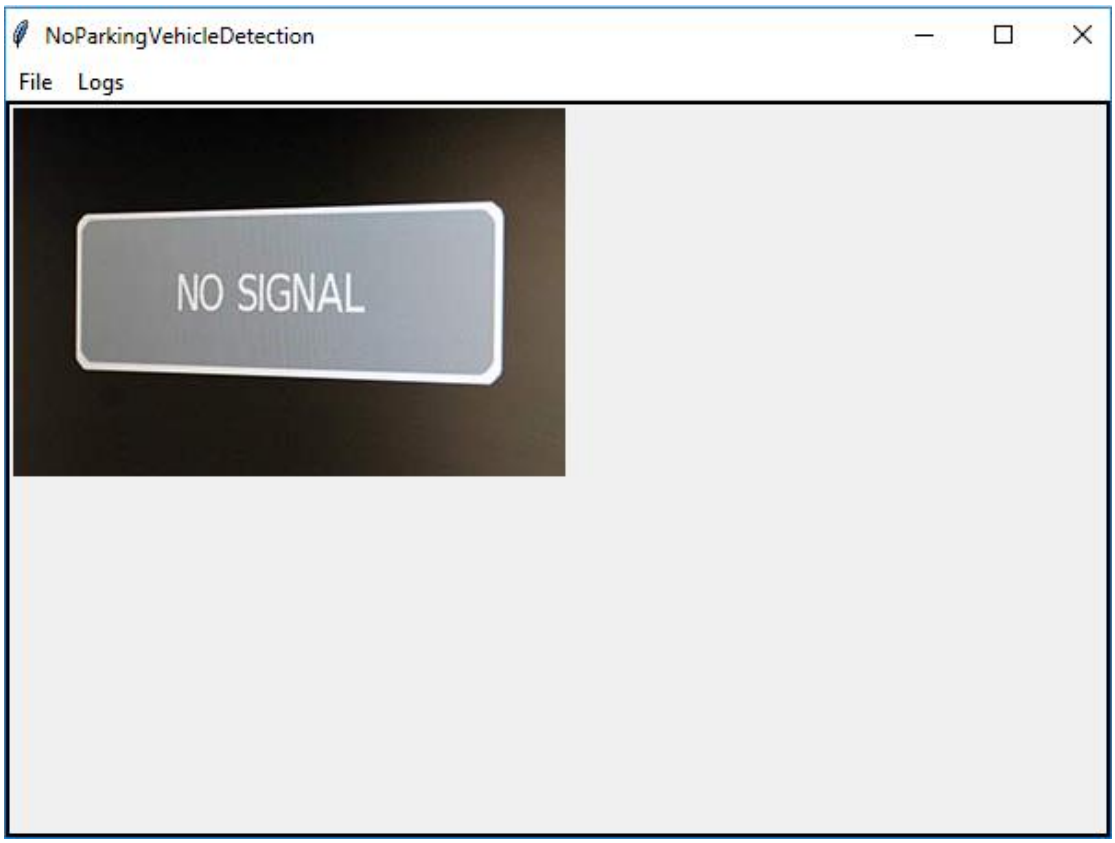


Fig 1: NPVD Home Page

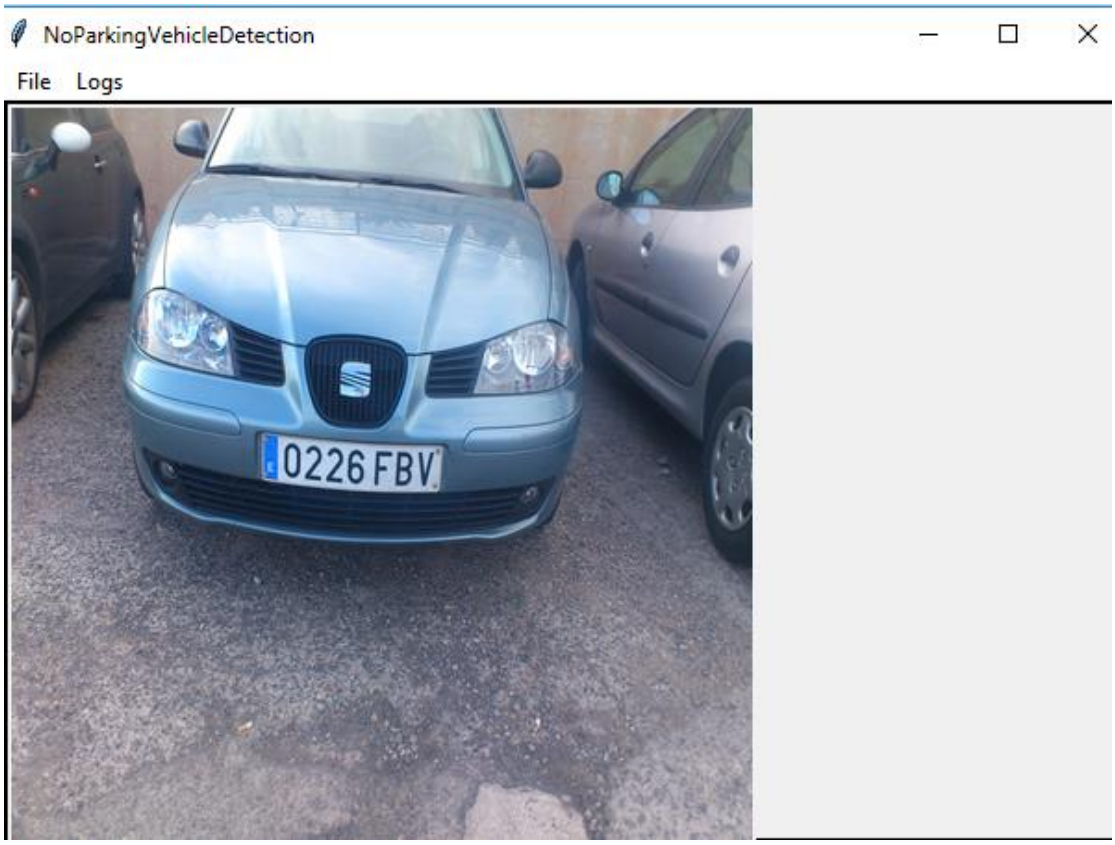


Fig 2: NPVD Live Stream Page

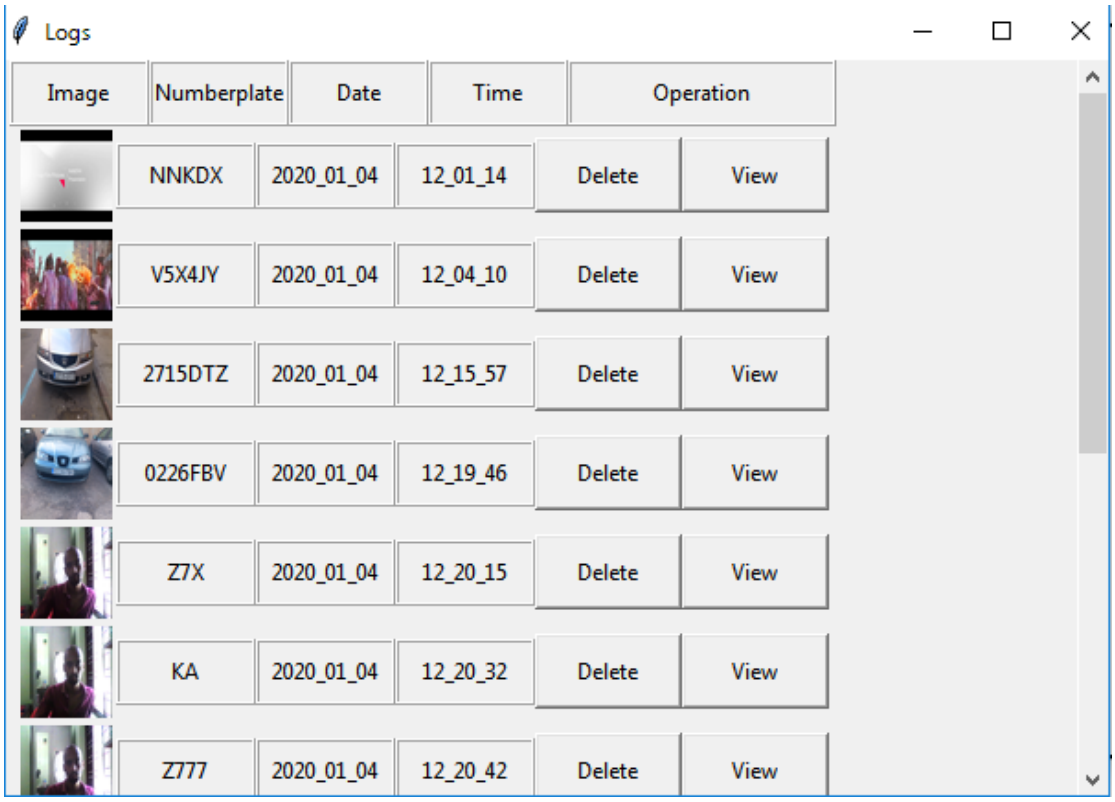


Fig 3: NPVD View Log Page

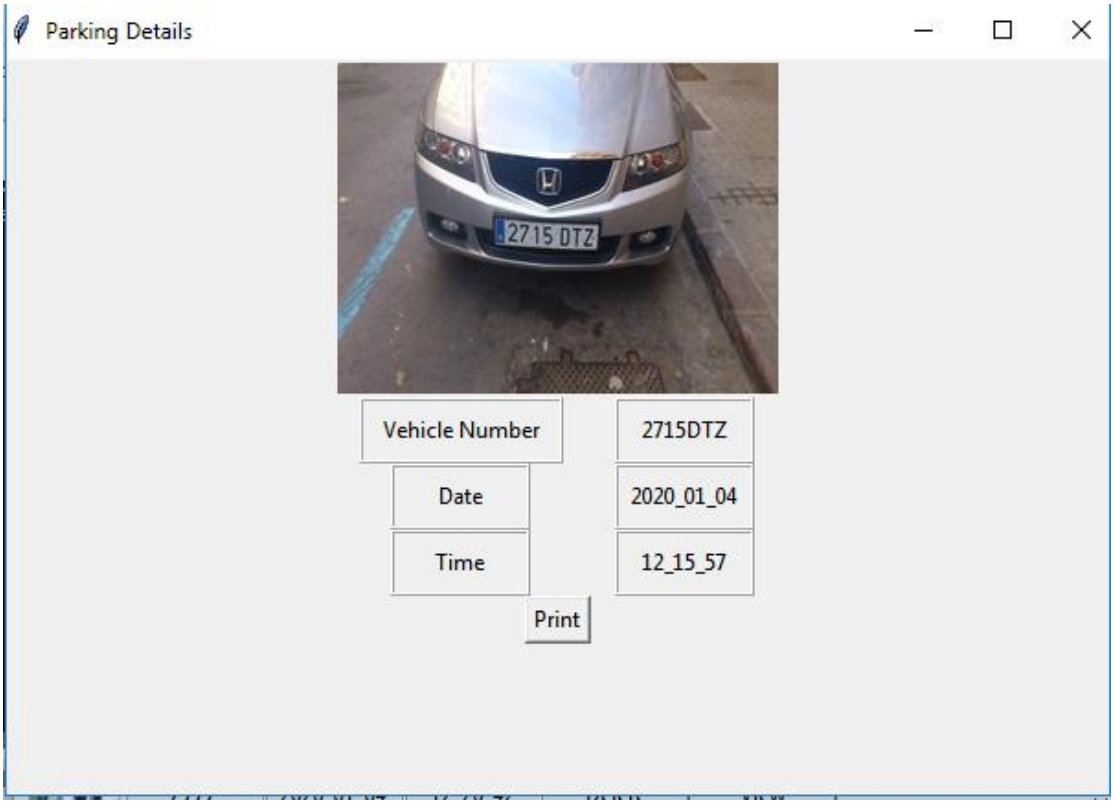


Fig 4: NPVD Vehicle Details

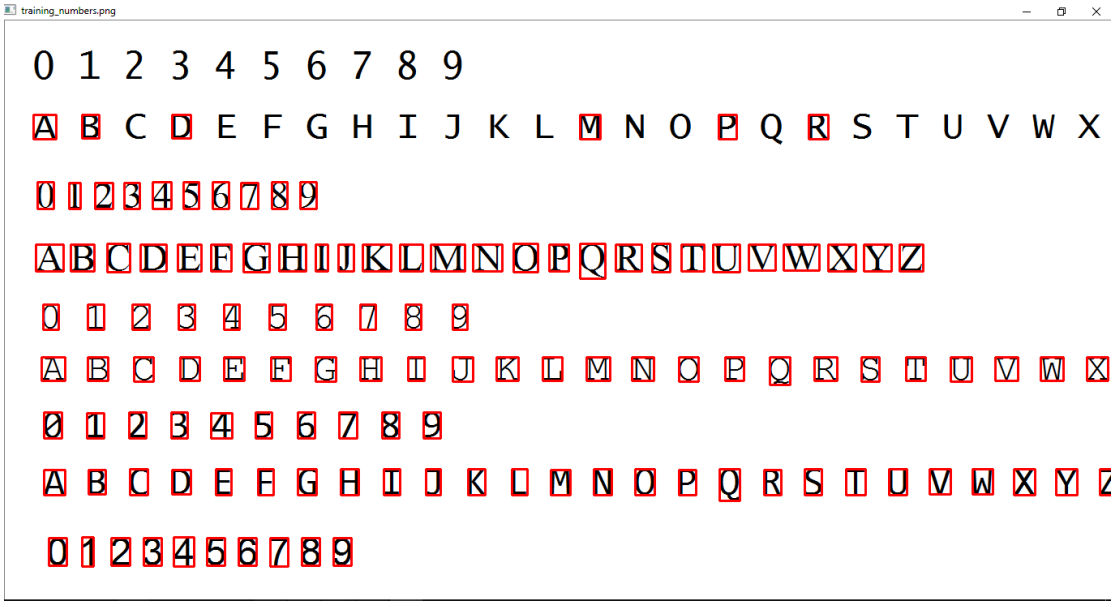


Fig 5: NPVD Training Data

BIBLIOGRAPHY

1. Munir Ahmad, Shabib Aftab and Iftikhar Ali - Sentiment Analysis of Tweets using SVM in International Journal of Computer Applications (0975 – 8887) Volume 177 – No.5, November 2017.
2. Ahmad, M., & Aftab, S. (2017). Analyzing the Performance of SVM for Polarity Detection with Different Datasets. International Journal of Modern Education and Computer Science (IJMECS), 9(10), 29- 36.
3. Zgheib, W. A., & Barbar, A. M. A Study using Support Vector Machines to Classify the Sentiments of Tweets.
4. Isa, D., Lee, L. H., Kallimani, V. P., & Rajkumar, R. (2008). Text document preprocessing with the Bayes formula for classification using the support vector machine. IEEE Transactions on Knowledge and Data engineering, 20(9), 1264-1272.
5. Malandrakis, N., Kazemzadeh, A., Potamianos, A., & Narayanan, S. (2013, June). SAIL: A hybrid approach to sentiment analysis. In SemEval@ NAACL-HLT (pp. 438-442).