Chapter 1

INTRODUCTION

1. INTRODUCTION

1.1 ABOUT THE PROJECT:

Vehicle Intrusion Detection is a software application that monitors a place or an area for malicious activity or access violations. **VID** is designed to provide human less security which has ability to alert the security services by triggering an alarm when an malicious activities takes place.

The main purpose of this project is to reduce the human effort by taking the advantage of computer vision and high speed computers. It is also a fact that humans are prone to make mistakes and on other hand it is important to automate the manual tasks to computers.

The system is simple and easy to use. It takes minimum efforts and cost to install anywhere.
It can be used 24x7 and also we can ensure 0 errors once installed.

It has built in security system which can automatically captures the intrusions and can alert the security services through an email within a span of seconds.

1.2 PURPOSE:

VID is fairly a new innovation in security. It is really important to constantly improve our security systems. VID system is designed in the situations where the security surveillance is heavily needed and in the situations like where traditional manual security check-ups become slow and dull. VID completely avoids the humans and function on fully automatic mode. It can troubleshoot its own problem and then can recover to normal mode when an unexpected crash in the software happens.

1.3 SCOPE:

VID has a lot of scope in future. It can be deployed to the areas where security is most important to the locality like military bases, nuclear sectors, defence secured areas….etc..,

It can also be deployed to small scale areas with minimum cost factor.

1.4 MOTIVATION:

The traditional security system might become outdated in upcoming years. Those manual techniques may not function well and also we cannot ensure the perfect security to the locality. An innovation is needed to overcome this deficit.

Chapter 2

SOFTWARE REQUIREMENTS SPECIFICATIONS

2. SOFTWARE REQUIREMENTS SPECIFICATIONS

2.1 PROBLEM DEFINITION:

Vehicle intrusion detection is a Computer Vision technique which is able to recognize a license plate number of the vehicle and make sure that the vehicle is authorized or not. If the vehicle is authorized then it allows the vehicle to proceed further otherwise it sends an email to the administrator with the details of the vehicle. This system is useful in many field likes parking lots, private and public entrances, theft control. In this project we designed such a system. First we capture the image from camera then load into system after that we used OpenCV library tools. Then we make the training set of different characters of different sizes. On the basis of these training set we extracted the character from images. When the license plate is detected, its digits are recognized and checks whether the acquired digits are in the database are not. If they exists in the database then it allows the vehicle otherwise sends an email to the security board.

2.2 EXISTING SYSTEMS AND ITS DRAWBACKS:

With rising traffic on roads and number of vehicles on roads, it is getting very difficult to manually handle laws, traffic rules and regulations for smooth traffic moment. Toll-booths are installed on freeways and parking complex, where the vehicle has to stop to pay the toll or parking charge fees. Also, Traffic maintenance systems are constructed on freeways to analyze for vehicles moving at speeds not allowed by law. All these operations have a scope of improve development. In the center of all these processes lies a vehicle. The vital question here is how to find a particular vehicle? The obvious response to this question is by using the vehicle's number plate. This number differentiate one vehicle from the other, which is effective especially when both are of same type of make and model. An automated system can be developed to find the license plate of a vehicle and recognize the characters from the region having a license plate. The vehicle license plate number which can be

utilized to fetch farther information data about the vehicle and its owner, which can be used for further processing.

Problems shortcomings of the current system

The current system presents a number of shortcomings that makes it ineffective.

This include;
(i)Storage of the records is problematic especially due to volume of data

(ii)Report generation is very difficult and inaccurate since some files get misplaced.

(iii)The system uses too many papers to maintain the records

(iv)The system is time wasting

(v)Data is not converted easily to information

(vi)Readiness of the information.

(vii)Data handling is a problem

2.3 PROPOSED SYSTEM AND ITS MERITS:

In India, basically there are two kinds of license plates, black characters in white plate and black characters in yellow plate. The process of proposed system consists of below steps:

1. **Capture**

The given image of the vehicle is captured by camera which is high resolution quality. We save the below image in the system

Fig. 2.1. Captured Image (Original image)

2. Pre-processing:

Pre-processing is the technique in which background illumination conditions and the number plate localization algorithms is used. In this phase mainly focuses on reduce background noise, enhancing of contrast. The system preprocessing uses two processes: Resize – In this section we have to change the size of object according to requirement. Convert Color Space – Images captured by cameras will be either in raw format or encoded into some multimedia standards. These images will be in RGB mode basically i.e. red, green and blue. There should be using OpenCV function in pre-processing phase.

Fig.2.2 Localization (Threshold image)

3. Number Plate Localization

The number plate localization is the phase in which mainly focuses on ROI (Region of Interest) where we find the contour region.

- **Contour Tracing**

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

- Since For better accuracy, use binary images. So before finding contours, apply threshold.

- In OpenCV, finding contours is like finding white object from black background

4. Character segmentation

Character segmentation is the technique in which individual character present in the image is separated out. Here all character is checked out individually.

5. Character Recognition

An automatic license plate recognition system must recognize alphanumeric characters. The character image is compared with the training set and the best similarity is measured and according to this recognized character is displayed..

Merits:

- **Secure & gated Entrances**
  For those who need their entrance fully secured, VID can offer a good benefit. Vehicles which are permitted to enter the premises can be identified with the help of license plate recognition system
- **It makes the work of security personnel easy**
- **Immediate action against the unauthorized vehicles**

2.4 FUNCTIONAL REQUIREMENTS:

In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behaviour, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish. Behavioural requirements describing all the cases where the system uses the functional requirements are captured in use cases. Functional requirements are supported by non-functional requirements (also known as quality requirements), which impose constraints on the design or implementation (such as performance requirements, security, or reliability).How a system implements functional requirements is detailed in the system design. In some cases a requirements analyst generates use cases after gathering and validating a set of functional requirements. Each use case illustrates behavioural scenarios through one or more functional requirements. Often, though, an analyst will begin by eliciting a set of use cases, from which the analyst can derive the functional requirements that must be implemented to allow a user to perform each use case.

2.4.1 Input Design:

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

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

2.4.2 Objectives:

Input Design is the process of converting a user-oriented description of the input into a computer-based system.

This design is important to avoid errors in the data input process and show the correct Direction to the management for getting correct information from the computerized system.

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

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

2.4.3 Output Design:

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

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

analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

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

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

- Convey information about past activities, current status or projections of the future.
- Signal important events, opportunities, problems, or warnings.

2.4 NON-FUNCTIONAL REQUIREMENTS:

In systems engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. This should be contrasted with functional requirements that define specific behaviour or functions In general; functional requirements define what a system is supposed to do whereas non-functional requirements define how a system is supposed to be.

Non-functional requirements are often called qualities of a system. Other terms for non-functional requirements are "constraints", "quality attributes", "quality goals" and "quality of service requirements," and "non-behavioural requirements.". Qualities, that is, non-functional requirements, can be divided into two main categories:

1. Execution qualities, such as security and usability, which are observable at run time.

2. Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

2.6 SOFTWARE ENVIRONMENT:

2.6.1 Python 3.6x

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

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object oriented, imperative, functional and procedural, it also has a comprehensive standard library.

2.6.2 Openpyxl:

Python has multiple 3rd party libraries for reading and writing Microsoft Excel files. For working with .xls files, there is xlrd for reading and xlwt for writing.

For working with .xlsx files, there is xlrd for reading, openpyxl for reading and writing, and XlsxWriter and PyExcelerate for writing.

2.6.3 OpenCV:

OpenCV (*Open source computer vision*) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel,[2] it was later supported by Willow Garage then Itseez (which was later acquired by Intel[3]). The library is cross-platform and free for use under the open-source BSD license.

The goals of the project were described as:

- Advance vision research by providing not only open but also optimized code for basic vision infrastructure. No more reinventing the wheel.
- Disseminate vision knowledge by providing a common infrastructure that developers could build on, so that code would be more readily readable and transferable.
- Advance vision-based commercial applications by making portable, performance-optimized code available for free – with a license that did not require code to be open or free itself.
-

Applications:

OpenCV's application areas include

- 2D and 3D feature toolkits
- Egomotion estimation
- Facial recognition system
- Gesture recognition
- Human–computer interaction (HCI)
- Mobile robotics
- Motion understanding
- Object identification
- Segmentation and recognition
- Motion tracking
- Augmented reality

To support some of the above areas, OpenCV includes a statistical machine learning library that contains:

- Boosting

- Decision tree learning
- Gradient boosting trees
- Expectation-maximization algorithm
- k-nearest neighbor algorithm
- Naive Bayes classifier
- Artificial neural networks
- Random forest
- Support vector machine (SVM)
- Deep neural networks (DNN)

2.6.4 Pymysql:

Pymysql is database framework which can act as a middle ware between python and it's associated database

It can perform all the crud operations required on the table with simpler syntaxes

2.6.5 SMTP Python:

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers. Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon

2.6.6 Python Deep Learning(python handle):

Deep learning is a class of machine learning algorithms that use several layers of nonlinear processing units for feature extraction and transformation. Each successive layer uses the output from the previous layer as input.

Deep neural networks, deep belief networks and recurrent neural networks have been applied to fields such as computer vision, speech recognition, natural language processing, audio recognition, social

network filtering, machine translation, and bioinformatics where they produced results comparable to and in some cases better than human experts have.

Deep Learning Algorithms and Networks –

- This are based on the unsupervised learning of multiple levels of features or representations of the data. Higher-level features are derived from lower level features to form a hierarchical representation.

- use some form of gradient descent for training.

2.6.6.1 Deep Learning With Python Libraries & Frameworks

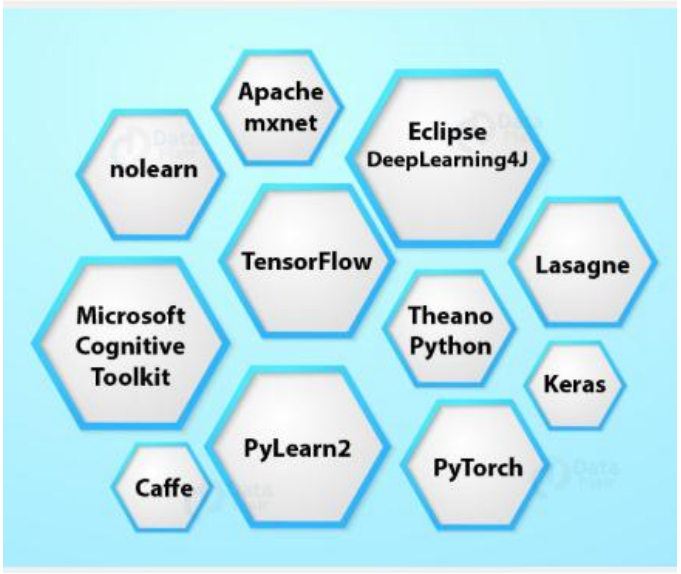


Fig 2.4 frameworks of deep learning

2.6.8 Webcam:

A webcam is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks traveling through systems such as the internet, and e-mailed as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there. Unlike an IP camera (which connects using Ethernet or Wi-Fi), a webcam is generally connected by a USB cable, or similar cable, or built into computer hardware, such as laptops.

2.7 HARDWARE AND SOFTWARE SPECIFICATION

2.7.1 HARDWARE SPECIFICATIONS

Component	Recommended Configuration
Processor	Intel Core i7 7800x Processor
Ram	12GB(8GB minimum)
Hard Disk	500GB(requires 150 mb for our application)
Input Device	Keyboard, Mouse, Webcam(optional)
Output Device	Monitor with internet connection
Operating system	Mac os,Window-07,08,10

2.7.2 SOFTWARE SPECIFICATIONS

Component	Configuration
Database	MySql
Software's	Workbench
Language	Python
Web Browser	Chrome

Chapter 3

FEASIBILITY STUDY

3. FEASIBILITY STUDY

All projects are feasible-given unlimited resources and infinite time! Unfortunately, the development of computer based system or product is more likely plagued by a scarcity of feasibility of a project at the earliest possible time. Months or years of effort, thousands or millions of dollars, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in definition phase

Feasibility and risk analysis are related in many ways. If project risk is great the feasibility of producing quality software is reduced. During product engineering, however, we concentrate our attention on four primary areas of interest

3.1 TECHNICAL FEASIBILITY:

Technical feasibility centres on the existing computer system, hardware, software etcetera and to what extent it can support the system. In examining the technical feasibility, the configuration of the system is given more importance than the actual hardware. The configuration should provide the complete picture of the system requirements, for example how many workstations are required and how these units are interconnected so that they would operate smoothly, etcetera. The result of the Technical Feasibility Study is the basis for the documents against which dealer and manufacturer can make bids. Specific hardware and software products can then be evaluated keeping in view the logical needs.

3.2 ECONOMIC FESIBILITY:

Economic analysis is the most frequently used method for evaluating the effectiveness of a new system. More commonly known as cost/benefit analysis, the procedure is to determine the benefits and savings that are expected from a candidate system and compare them with costs. If benefits outweigh costs, then the decision is made to design and implement the system.

In present system it is easy to manage the vehicles that are in the database and provides less complexity on addition and removal of vehicles.
Generates immediate response on detection of any unusual activities.

It should be required for both pilot and long-term activities, plans and projects.

3.3 OPERATIONAL FEASIBILITY:

In our application the admin who manages the database need to update whenever there is any change in the vehicles or the addition or removal of vehicles. So it is very easy to the administrator to enter the necessary information. But customer has some knowledge on using database before going to use our application.

It determines how acceptable the software is within the organization. The evaluations must then determine the general attitude and skills. Such restriction of the job will be acceptable. To the users are enough to run the proposed budget, hence the system is supposed to the feasible regarding all except of feasibility. In operational feasibility, we attempt to ensure that every user can access the system easily.

Chapter 4

HIGH LEVEL DESIGNS

4. HIGH LEVEL DESIGNS

High-level design (HLD) explains the architecture that would be used for developing a software product. The architecture diagram provides an overview of an entire system, identifying the main components that would be developed for the product and their interfaces.

The HLD uses possibly nontechnical to mildly technical terms that should be understandable to the administrators of the system. A high-level design provides an overview of a solution, platform, system, product, service or process. Purpose: Preliminary design: In the preliminary stages of a software development, the need is to size the project and to identify those parts of the project that might be risky or time consuming. Design overview: As the project proceeds, the need is to provide an overview of how the various sub-systems and components of the system fit together. In both cases the high-level design should be a complete view of the entire system, breaking it down into smaller parts that are more easily understood. To minimize the maintenance overhead as construction proceeds and the lower-level design is done, it is best that the high-level design is elaborated only to the degree needed to satisfy these needs.

4.1 DESIGN CONSIDERATION:

General Constraints

The current constraints on the project are related to the provision of hardware resources to implement and test a high-performance. At present, we are using I3processor with 12GB ram as a machine. For better performance analysis, we have to use 16GB ram with I-7 processor.

4.1.2 Assumptions and dependencies:

A number of factors that may affect the requirements specified in the SRS include: The workability of the Sun Grid Engine cluster management system modules such as those dealing with process migration with the scheduling policies provided by the Libra Scheduler is assumed. A basic module of job accounting and payment considerations will be provided, as they are not the focus of the scheduler. Users are assumed to have a fair estimate of job execution times, so that the decision to accept or reject a job is facilitated. SDLC Models There are various software development life cycle models defined and designed which are followed during software development process. These models are also refer as "Software Development Process Models". Each process model follows a Series of steps unique to its type, in order to ensure success in process of software development.

Following are the most important and popular SDLC models followed in the industry:

4.1.3 Development Methods:

Waterfall model is the earliest SDLC approach that was used for software development. The waterfall Model illustrates the software development process in a linear sequential flow; hence it is also referred to as a linear-sequential life cycle model. This means that any phase in the development process begins only if the previous phase is complete. In waterfall model phases do not overlap Waterfall Model design Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project.

In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one the input for the next phase sequentially. Following is a diagrammatic representation of different phases of waterfall model.

4.2 UML DESIGNS:

4.2.1 UML:

UML stands for Unified Modelling Language. UML is a standardized general purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major component Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML. The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

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

4.2.2GOALS:

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

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

2. Provide extendibility and specialization mechanisms to extend the core concept

3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modelling language.

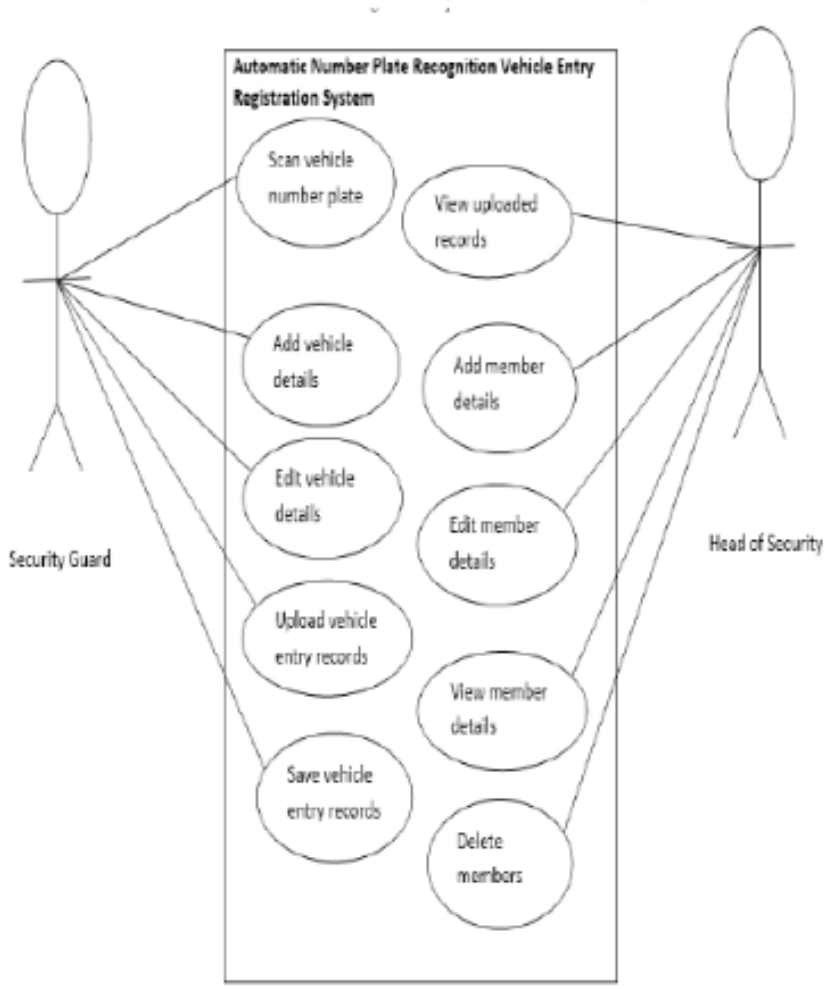
5. Encourage the growth of OO tools market.

4.2.3 USECASE:

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

4.2.3.1 User Login:

Fig (4.1) Use case of User login



4.2.4 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

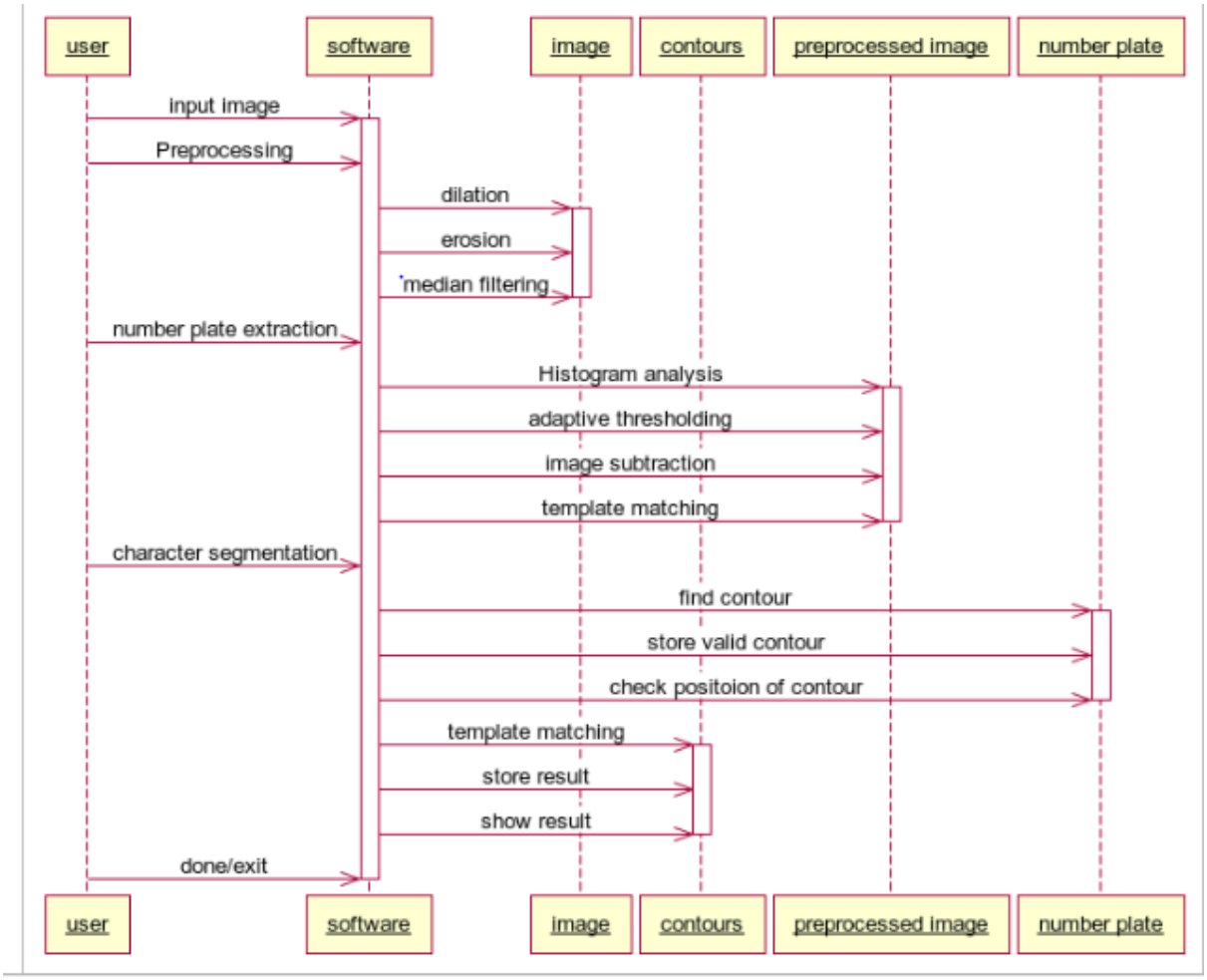


Fig (4.2) Sequence Diagram

Chapter 5

IMPLEMENTAION

5. IMPLEMENTATION

The first stage of implementation consists of rendering the video and produces the frames that
are generated by video. Then the frames are preprocessed by applying set of algorithms in
order to enhance the quality.

The next stage that follows is the number plate recognition phase that does several functions such as resizing of the image to a feasible aspect ratio. As well as converting the colored
image into a grey scale, image.

The number plate can be found anywhere within an image, it is impractical to check all the pixels of the image in order to locate the number plate. Therefore, we only focus on those pixels that have the number plate.

Then In order for proper text recognition to take place the line of text is first segmented, then from the segmented line the words are segmented and then from that the characters are segmented

Finally, Template matching, or matrix matching, is one of the most common classification methods. In template matching, individual image pixels are used as features. Classification is performed by comparing an input character image with a set of templates from each character class

5.1 Checkplate.py

```
import cv2
import numpy as np
import math
```

```
import Main
import random

import Preprocess
import DetectChars
import SearchPossiblePlates
import PossibleChar

PLATE_WIDTH_PADDING_FACTOR = 1.3
PLATE_HEIGHT_PADDING_FACTOR = 1.5

def detectPlatesInScene(imgOriginalScene):
    listOfPossiblePlates = []

    height, width, numChannels = imgOriginalScene.shape

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

    cv2.destroyAllWindows()

    if Main.showSteps == True:
        cv2.imshow("0", imgOriginalScene)

    imgGrayscaleScene, imgThreshScene = Preprocess.preprocess(imgOriginalScene)

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

    listOfPossibleCharsInScene = findPossibleCharsInScene(imgThreshScene)

    if Main.showSteps == True:
        print("step 2 - len(listOfPossibleCharsInScene) = " + str(
            len(listOfPossibleCharsInScene)))

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

        contours = []

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

        cv2.drawContours(imgContours, contours, -1, Main.SCALAR_WHITE)
```

```
        cv2.imshow("2b", imgContours)

    listOfListsOfMatchingCharsInScene =
DetectChars.findListOfListsOfMatchingChars(listOfPossibleCharsInScene)

    if Main.showSteps == True:
        print("step 3 - listOfListsOfMatchingCharsInScene.Count = " + str(
            len(listOfListsOfMatchingCharsInScene)))

        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)

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

        cv2.imshow("3", imgContours)

    for listOfMatchingChars in listOfListsOfMatchingCharsInScene:
        possiblePlate = extractPlate(imgOriginalScene, listOfMatchingChars)

        if possiblePlate.imgPlate is not None:
            listOfPossiblePlates.append(possiblePlate)

    print("\n" + str(len(listOfPossiblePlates)) + " possible plates found")

    if Main.showSteps == True:
        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)

        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 = []

    intCountOfPossibleChars = 0

    imgThreshCopy = imgThresh.copy()

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

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

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

        if Main.showSteps == True:
            cv2.drawContours(imgContours, contours, i, Main.SCALAR_WHITE)

        possibleChar = PossibleChar.PossibleChar(contours[i])

        if DetectChars.checkIfPossibleChar(possibleChar):
            intCountOfPossibleChars = intCountOfPossibleChars + 1
            listOfPossibleChars.append(possibleChar)

    if Main.showSteps == True:
        print("\nstep 2 - len(contours) = " + str(len(contours)))
        print("step 2 - intCountOfPossibleChars = " + str(intCountOfPossibleChars))
        cv2.imshow("2a", imgContours)

    return listOfPossibleChars

def extractPlate(imgOriginal, listOfMatchingChars):
```

```
    possiblePlate = SearchPossiblePlates.PossiblePlate()

    listOfMatchingChars.sort(key = lambda matchingChar: matchingChar.intCenterX)

    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

    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

    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)

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

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

    height, width, numChannels = imgOriginal.shape

    imgRotated = cv2.warpAffine(imgOriginal, rotationMatrix, (width, height))

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

    possiblePlate.imgPlate = imgCropped

    return possiblePlate
```

5.2 Database_util.py

```
import pymysql
import openpyxl
from helpers import *

class Database:

    def __init__(self):
        self.conn = pymysql.connect(host='localhost', user='root', password='')

    def reset_database(self):
        # Resets the database to normal state
        try:
            self.conn.cursor().execute('DROP DATABASE project')
        except:
            print(">>> Required database not found....")
            print(">>> Creating new database")
        self.conn.cursor().execute('CREATE DATABASE project')
        self.conn.cursor().execute('CREATE TABLE PROJECT.auth_user(USER_ID INTEGER, REG_NO
VARCHAR(20), CAR_MODEL VARCHAR(20),NAME VARCHAR(40),PHONE_NO
VARCHAR(15),DEPT_ID INTEGER)')
        self.conn.cursor().execute('CREATE TABLE PROJECT.department(DEPT_ID
INTEGER,DEPT_NAME VARCHAR(50))')
        print(">>> Database has been reset to empty")

    def load_data(self):
        # Open excel book and read the content to the database
        self.load_auth_users_automatically()
        self.load_department_automatically()
        pass

    def reset_auth_users(self):
        # Drop the existing PROJECT.AUTH_USERS and recreate the table
        self.conn.cursor().execute('DROP TABLE PROJECT.auth_user')
        self.conn.cursor().execute('CREATE TABLE PROJECT.auth_user(USER_ID INTEGER, REG_NO
VARCHAR(20), CAR_MODEL VARCHAR(20),NAME VARCHAR(40),PHONE_NO
```

```
VARCHAR(15),DEPT_ID INTEGER)')
        print(">>> auth_users has been reset to empty state")
        pass

    def reset_department(self):
        # Drop the existing PROJECT.DEPARTMENT and recreate the table
        self.conn.cursor().execute('DROP TABLE PROJECT.department')
        self.conn.cursor().execute('CREATE TABLE PROJECT.department(DEPT_ID
INTEGER,DEPT_NAME VARCHAR(50))')
        print(">>> department table has been reset to empty state")
        pass

    def load_auth_users_manually(self):
        # Read the content from the terminal
        row=input(">>> Enter values with comma seperated as ---
id,reg_no,car_model,name,phone_no,dept_id\n")
        sql="INSERT INTO PROJECT.auth_user values(" +row+")"
        print(sql)
        self.conn.cursor().execute(sql)
        self.conn.commit()
        print("query successful")
        pass

    def load_auth_users_automatically(self):
        # Read the workbook sheet1 only to the database
        wb = openpyxl.load_workbook(WORKBOOK_PATH)
        sheet = wb['Sheet1']

        for i in range(2,sheet.max_row):
            data=
str(sheet[i][0].value)+str(",'"+sheet[i][1].value+"'")+str(",'"+sheet[i][2].value+"'")+str(",'"+sheet[i][3].value+"
'")+str(",'"+str(sheet[i][4].value)+"',")+str(sheet[i][5].value)
            #print(data)
            sql="INSERT INTO PROJECT.auth_user values(" +data+")"
            print(sql)
            self.conn.cursor().execute(sql)
            self.conn.commit()
        pass

    def load_department_manually(self):
        # Read the content from the terminal
        pass

    def load_department_automatically(self):
        # Read the workbook sheet2 only to the database
```

```
        pass

    def check(self,reg_no):
        # Checks if a record exists or not in the database
        sql="select * from PROJECT.auth_user where REG_NO like '%"+reg_no+"%';"
        result=self.conn.cursor().execute(sql)
        return result
        pass

database = Database()
database.reset_database()
database.load_auth_users_automatically()
#database.reset_database()
#database.load_auth_users_manually()

def query(number):
    # Check if the car is registered or not
    return database.check(number)
```

5.3 Detecting Chars:

```
import os

import cv2
import numpy as np
import math
import random

import Main
import Preprocess
import PossibleChar

kNearest = cv2.ml.KNearest_create()

MIN_PIXEL_WIDTH = 2
MIN_PIXEL_HEIGHT = 8

MIN_ASPECT_RATIO = 0.25
MAX_ASPECT_RATIO = 1.0

MIN_PIXEL_AREA = 80
```

```
MIN_DIAG_SIZE_MULTIPLE_AWAY = 0.3
MAX_DIAG_SIZE_MULTIPLE_AWAY = 5.0

MAX_CHANGE_IN_AREA = 0.5

MAX_CHANGE_IN_WIDTH = 0.8
MAX_CHANGE_IN_HEIGHT = 0.2

MAX_ANGLE_BETWEEN_CHARS = 12.0

MIN_NUMBER_OF_MATCHING_CHARS = 3

RESIZED_CHAR_IMAGE_WIDTH = 20
RESIZED_CHAR_IMAGE_HEIGHT = 30

MIN_CONTOUR_AREA = 100

def loadKNNDataAndTrainKNN():
    allContoursWithData = []
    validContoursWithData = []

    try:
        npaClassifications = np.loadtxt("classifications.txt", np.float32)
    except:
        print("error, unable to open classifications.txt, exiting program\n")
        os.system("pause")
        return False

    try:
        npaFlattenedImages = np.loadtxt("flattened_images.txt", np.float32)
    except:
        print("error, unable to open flattened_images.txt, exiting program\n")
        os.system("pause")
        return False
    # end try

    npaClassifications = npaClassifications.reshape((npaClassifications.size, 1))

    kNearest.setDefaultK(1)

    kNearest.train(npaFlattenedImages, cv2.ml.ROW_SAMPLE, npaClassifications)

    return True
```

```
def detectCharsInPlates(listOfPossiblePlates):
    # Detects the characters present on all the rectangular plates
    intPlateCounter = 0
    imgContours = None
    contours = []

    if len(listOfPossiblePlates) == 0:
        return listOfPossiblePlates

    for possiblePlate in listOfPossiblePlates:

        possiblePlate.imgGrayscale, possiblePlate.imgThresh = Preprocess.preprocess(possiblePlate.imgPlate)

        if Main.showSteps == True:
            cv2.imshow("5a", possiblePlate.imgPlate)
            cv2.imshow("5b", possiblePlate.imgGrayscale)
            cv2.imshow("5c", possiblePlate.imgThresh)

        possiblePlate.imgThresh = cv2.resize(possiblePlate.imgThresh, (0, 0), fx = 1.6, fy = 1.6)

        thresholdValue, possiblePlate.imgThresh = cv2.threshold(possiblePlate.imgThresh, 0.0, 255.0,
cv2.THRESH_BINARY | cv2.THRESH_OTSU)

        if Main.showSteps == True:
            cv2.imshow("5d", possiblePlate.imgThresh)

        listOfPossibleCharsInPlate = findPossibleCharsInPlate(possiblePlate.imgGrayscale, possiblePlate.imgThresh)

        if Main.showSteps == True:
            height, width, numChannels = possiblePlate.imgPlate.shape
            imgContours = np.zeros((height, width, 3), np.uint8)
            del contours[:]

            for possibleChar in listOfPossibleCharsInPlate:
                contours.append(possibleChar.contour)

            cv2.drawContours(imgContours, contours, -1, Main.SCALAR_WHITE)

            cv2.imshow("6", imgContours)

        listOfListsOfMatchingCharsInPlate = findListOfListsOfMatchingChars(listOfPossibleCharsInPlate)

        if Main.showSteps == True:
            imgContours = np.zeros((height, width, 3), np.uint8)
            del contours[:]
```

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

                for matchingChar in listOfMatchingChars:
                    contours.append(matchingChar.contour)

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

            cv2.imshow("7", imgContours)


        if (len(listOfListsOfMatchingCharsInPlate) == 0):

            if Main.showSteps == True:
                print("chars found in plate number " + str(
                    intPlateCounter) + " = (none), click on any image and press a key to continue . . .")
                intPlateCounter = intPlateCounter + 1
                cv2.destroyWindow("8")
                cv2.destroyWindow("9")
                cv2.destroyWindow("10")
                cv2.waitKey(0)

            possiblePlate.strChars = ""
            continue

        for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
            listOfListsOfMatchingCharsInPlate[i].sort(key = lambda matchingChar: matchingChar.intCenterX)
            listOfListsOfMatchingCharsInPlate[i] =
    removeInnerOverlappingChars(listOfListsOfMatchingCharsInPlate[i])


        if Main.showSteps == True:
            imgContours = np.zeros((height, width, 3), np.uint8)

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

                del contours[:]

                for matchingChar in listOfMatchingChars:
```

```
                contours.append(matchingChar.contour)


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

        cv2.imshow("8", imgContours)

    intLenOfLongestListOfChars = 0
    intIndexOfLongestListOfChars = 0

    for i in range(0, len(listOfListsOfMatchingCharsInPlate)):
        if len(listOfListsOfMatchingCharsInPlate[i]) > intLenOfLongestListOfChars:
            intLenOfLongestListOfChars = len(listOfListsOfMatchingCharsInPlate[i])
            intIndexOfLongestListOfChars = i

    longestListOfMatchingCharsInPlate = listOfListsOfMatchingCharsInPlate[intIndexOfLongestListOfChars]

    if Main.showSteps == True:
        imgContours = np.zeros((height, width, 3), np.uint8)
        del contours[:]

        for matchingChar in longestListOfMatchingCharsInPlate:
            contours.append(matchingChar.contour)

        cv2.drawContours(imgContours, contours, -1, Main.SCALAR_WHITE)

        cv2.imshow("9", imgContours)

    possiblePlate.strChars = recognizeCharsInPlate(possiblePlate.imgThresh,
longestListOfMatchingCharsInPlate)

    if Main.showSteps == True:
        print("chars found in plate number " + str(
            intPlateCounter) + " = " + possiblePlate.strChars + ", click on any image and press a key to continue
. . .")
        intPlateCounter = intPlateCounter + 1
        cv2.waitKey(0)

if Main.showSteps == True:
    print("\nchar detection complete, click on any image and press a key to continue . . .\n")
    cv2.waitKey(0)

return listOfPossiblePlates

def findPossibleCharsInPlate(imgGrayscale, imgThresh):
```

```
    # Filters the rectangular plates and returns the possible plates with characters
    listOfPossibleChars = []
    contours = []
    imgThreshCopy = imgThresh.copy()


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

    for contour in contours:
        possibleChar = PossibleChar.PossibleChar(contour)

        if checkIfPossibleChar(possibleChar):
            listOfPossibleChars.append(possibleChar)
    return listOfPossibleChars

def checkIfPossibleChar(possibleChar):
    # Method to check the printed font is a character with PIXELS width
    if (possibleChar.intBoundingRectArea > MIN_PIXEL_AREA and
        possibleChar.intBoundingRectWidth > MIN_PIXEL_WIDTH and possibleChar.intBoundingRectHeight >
MIN_PIXEL_HEIGHT and
        MIN_ASPECT_RATIO < possibleChar.fltAspectRatio and possibleChar.fltAspectRatio <
MAX_ASPECT_RATIO):
        return True
    else:
        return False

def findListOfListsOfMatchingChars(listOfPossibleChars):

    listOfListsOfMatchingChars = []

    for possibleChar in listOfPossibleChars:
        listOfMatchingChars = findListOfMatchingChars(possibleChar, listOfPossibleChars)

        listOfMatchingChars.append(possibleChar)

        if len(listOfMatchingChars) < MIN_NUMBER_OF_MATCHING_CHARS:
            continue


        listOfListsOfMatchingChars.append(listOfMatchingChars)

        listOfPossibleCharsWithCurrentMatchesRemoved = []

        listOfPossibleCharsWithCurrentMatchesRemoved = list(set(listOfPossibleChars) - set(listOfMatchingChars))
```

```
        recursiveListOfListsOfMatchingChars =
findListOfListsOfMatchingChars(listOfPossibleCharsWithCurrentMatchesRemoved)

        for recursiveListOfMatchingChars in recursiveListOfListsOfMatchingChars:
            listOfListsOfMatchingChars.append(recursiveListOfMatchingChars)

        break


    return listOfListsOfMatchingChars

def findListOfMatchingChars(possibleChar, listOfChars):

    listOfMatchingChars = []

    for possibleMatchingChar in listOfChars:
        if possibleMatchingChar == possibleChar:

            continue

        fltDistanceBetweenChars = distanceBetweenChars(possibleChar, possibleMatchingChar)

        fltAngleBetweenChars = angleBetweenChars(possibleChar, possibleMatchingChar)

        fltChangeInArea = float(abs(possibleMatchingChar.intBoundingRectArea -
possibleChar.intBoundingRectArea)) / float(possibleChar.intBoundingRectArea)

        fltChangeInWidth = float(abs(possibleMatchingChar.intBoundingRectWidth -
possibleChar.intBoundingRectWidth)) / float(possibleChar.intBoundingRectWidth)
        fltChangeInHeight = float(abs(possibleMatchingChar.intBoundingRectHeight -
possibleChar.intBoundingRectHeight)) / float(possibleChar.intBoundingRectHeight)

        if (fltDistanceBetweenChars < (possibleChar.fltDiagonalSize * MAX_DIAG_SIZE_MULTIPLE_AWAY)
and
            fltAngleBetweenChars < MAX_ANGLE_BETWEEN_CHARS and
            fltChangeInArea < MAX_CHANGE_IN_AREA and
            fltChangeInWidth < MAX_CHANGE_IN_WIDTH and
            fltChangeInHeight < MAX_CHANGE_IN_HEIGHT):

            listOfMatchingChars.append(possibleMatchingChar)

    return listOfMatchingChars

def distanceBetweenChars(firstChar, secondChar):
```

```
    intX = abs(firstChar.intCenterX - secondChar.intCenterX)
    intY = abs(firstChar.intCenterY - secondChar.intCenterY)

    return math.sqrt((intX ** 2) + (intY ** 2))

def angleBetweenChars(firstChar, secondChar):
    fltAdj = float(abs(firstChar.intCenterX - secondChar.intCenterX))
    fltOpp = float(abs(firstChar.intCenterY - secondChar.intCenterY))

    if fltAdj != 0.0:
        fltAngleInRad = math.atan(fltOpp / fltAdj)
    else:
        fltAngleInRad = 1.5708

    fltAngleInDeg = fltAngleInRad * (180.0 / math.pi)

    return fltAngleInDeg

def removeInnerOverlappingChars(listOfMatchingChars):
    listOfMatchingCharsWithInnerCharRemoved = list(listOfMatchingChars)

    for currentChar in listOfMatchingChars:
        for otherChar in listOfMatchingChars:
            if currentChar != otherChar:

                if distanceBetweenChars(currentChar, otherChar) < (currentChar.fltDiagonalSize *
MIN_DIAG_SIZE_MULTIPLE_AWAY):

                    if currentChar.intBoundingRectArea < otherChar.intBoundingRectArea:
                        if currentChar in listOfMatchingCharsWithInnerCharRemoved:
                            listOfMatchingCharsWithInnerCharRemoved.remove(currentChar)
                    else:
                        if otherChar in listOfMatchingCharsWithInnerCharRemoved:
                            listOfMatchingCharsWithInnerCharRemoved.remove(otherChar)

    return listOfMatchingCharsWithInnerCharRemoved

def recognizeCharsInPlate(imgThresh, listOfMatchingChars):
    strChars = ""

    height, width = imgThresh.shape

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

    listOfMatchingChars.sort(key = lambda matchingChar: matchingChar.intCenterX)
```

```
    cv2.cvtColor(imgThresh, cv2.COLOR_GRAY2BGR, imgThreshColor)

    for currentChar in listOfMatchingChars:
        pt1 = (currentChar.intBoundingRectX, currentChar.intBoundingRectY)
        pt2 = ((currentChar.intBoundingRectX + currentChar.intBoundingRectWidth),
(currentChar.intBoundingRectY + currentChar.intBoundingRectHeight))

        cv2.rectangle(imgThreshColor, pt1, pt2, Main.SCALAR_GREEN, 2)

        imgROI = imgThresh[currentChar.intBoundingRectY : currentChar.intBoundingRectY +
currentChar.intBoundingRectHeight,
                           currentChar.intBoundingRectX : currentChar.intBoundingRectX +
currentChar.intBoundingRectWidth]

        imgROIResized = cv2.resize(imgROI, (RESIZED_CHAR_IMAGE_WIDTH,
RESIZED_CHAR_IMAGE_HEIGHT))

        npaROIResized = imgROIResized.reshape((1, RESIZED_CHAR_IMAGE_WIDTH *
RESIZED_CHAR_IMAGE_HEIGHT))

        npaROIResized = np.float32(npaROIResized)

        retval, npaResults, neigh_resp, dists = kNearest.findNearest(npaROIResized, k = 1)

        strCurrentChar = str(chr(int(npaResults[0][0])))

        strChars = strChars + strCurrentChar


    if Main.showSteps == True:
        cv2.imshow("10", imgThreshColor)

    return strChars
```

5.4 Smtp protocol:

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
import smtplib

msg=MIMEMultipart()
```

```
msg['From']='karthikvg1998@gmail.com'
msg['To']='karthikvg1998@gmail.com'
msg['Subject']='Security Alert'


def email(number):
    body ='<html><head>intrusion has been detected!!!</head><body>\n'
    body+='<br/><br/>'
    body+='<p>'+number+'</p>'
    msg.attach(MIMEText(body,'html'))
    server= smtplib.SMTP("smtp.gmail.com",587)
    server.starttls()
    server.login(msg['From'],'******')
    server.sendmail(msg['From'],msg['To'],msg.as_string())
    server.quit()
#email("karthik")
```

5.5 Main Script

```
import database_util
import mailling_util

import cv2
import os
import DetectChars
import CheckPlates

showSteps = False

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)

def main():

    blnKNNTrainingSuccessful = DetectChars.loadKNNDataAndTrainKNN()

    if blnKNNTrainingSuccessful == False:
        print("\nerror: KNN traning was not successful\n")
        return
    path=input()
```

```
imgOriginalScene  = cv2.imread("LicPlateImages/"+path)

if imgOriginalScene is None:
    print("\nerror: image not read from file \n\n")
    os.system("pause")
    return

listOfPossiblePlates = CheckPlates.detectPlatesInScene(imgOriginalScene)

listOfPossiblePlates = DetectChars.detectCharsInPlates(listOfPossiblePlates)

cv2.imshow("imgOriginalScene", imgOriginalScene)

if len(listOfPossiblePlates) == 0:
    print("\nno license plates were detected\n")
else:

    listOfPossiblePlates.sort(key = lambda possiblePlate: len(possiblePlate.strChars), reverse = True)

    licPlate = listOfPossiblePlates[0]

    cv2.imshow("imgPlate", licPlate.imgPlate)
    cv2.imshow("imgThresh", licPlate.imgThresh)

    if len(licPlate.strChars) == 0:
        print("\nno characters were detected\n\n")
        return

    drawRedRectangleAroundPlate(imgOriginalScene, licPlate)

    print("\nlicense plate read from image = " + licPlate.strChars + "\n")
    if database_util.query(licPlate.strChars)==0:
        print("intrusion is detected alerting the security system through email!!!")
        mailling_util.email(licPlate.strChars)
    else:
        print("safe to move!!!")

    writeLicensePlateCharsOnImage(imgOriginalScene, licPlate)

    cv2.imshow("imgOriginalScene", imgOriginalScene)

    cv2.imwrite("imgOriginalScene.png", imgOriginalScene)


cv2.waitKey(0)
```

```
    return

def drawRedRectangleAroundPlate(imgOriginalScene, licPlate):

    p2fRectPoints = cv2.boxPoints(licPlate.rrLocationOfPlateInScene)

    cv2.line(imgOriginalScene, tuple(p2fRectPoints[0]), tuple(p2fRectPoints[1]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[1]), tuple(p2fRectPoints[2]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[2]), tuple(p2fRectPoints[3]), SCALAR_RED, 2)
    cv2.line(imgOriginalScene, tuple(p2fRectPoints[3]), tuple(p2fRectPoints[0]), SCALAR_RED, 2)


def writeLicensePlateCharsOnImage(imgOriginalScene, licPlate):
    ptCenterOfTextAreaX = 0
    ptCenterOfTextAreaY = 0

    ptLowerLeftTextOriginX = 0
    ptLowerLeftTextOriginY = 0

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

    intFontFace = cv2.FONT_HERSHEY_SIMPLEX
    fltFontScale = float(plateHeight) / 30.0
    intFontThickness = int(round(fltFontScale * 1.5))

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

    ( (intPlateCenterX, intPlateCenterY), (intPlateWidth, intPlateHeight), fltCorrectionAngleInDeg ) =
licPlate.rrLocationOfPlateInScene

    intPlateCenterX = int(intPlateCenterX)
    intPlateCenterY = int(intPlateCenterY)

    ptCenterOfTextAreaX = int(intPlateCenterX)

    if intPlateCenterY < (sceneHeight * 0.75):
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) + int(round(plateHeight * 1.6))
    else:
        ptCenterOfTextAreaY = int(round(intPlateCenterY)) - int(round(plateHeight * 1.6))


    textSizeWidth, textSizeHeight = textSize
```

```
    ptLowerLeftTextOriginX = int(ptCenterOfTextAreaX - (textSizeWidth / 2))
    ptLowerLeftTextOriginY = int(ptCenterOfTextAreaY + (textSizeHeight / 2))


    cv2.putText(imgOriginalScene, licPlate.strChars, (ptLowerLeftTextOriginX, ptLowerLeftTextOriginY),
intFontFace, fltFontScale, SCALAR_YELLOW, intFontThickness)


if __name__ == "__main__":
    main()
```

5.6 Possible characters:

```
import cv2
import numpy as np
import math

class PossibleChar:

    def __init__(self, _contour):
        self.contour = _contour

        self.boundingRect = cv2.boundingRect(self.contour)

        [intX, intY, intWidth, intHeight] = self.boundingRect

        self.intBoundingRectX = intX
        self.intBoundingRectY = intY
        self.intBoundingRectWidth = intWidth
        self.intBoundingRectHeight = intHeight

        self.intBoundingRectArea = self.intBoundingRectWidth * self.intBoundingRectHeight

        self.intCenterX = (self.intBoundingRectX + self.intBoundingRectX + self.intBoundingRectWidth) / 2
        self.intCenterY = (self.intBoundingRectY + self.intBoundingRectY + self.intBoundingRectHeight) / 2

        self.fltDiagonalSize = math.sqrt((self.intBoundingRectWidth ** 2) + (self.intBoundingRectHeight ** 2))

        self.fltAspectRatio = float(self.intBoundingRectWidth) / float(self.intBoundingRectHeight)
```

5.7 Preprocess:

```
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

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

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
```

Chapter 6

TESTING

6. TESTING

Testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. The logical design and physical design is thoroughly and continually examined on paper to ensure that they will work when implemented.

Thus the system test in implementation was a confirmation that all is correct and an opportunity to show the users that the system works.

Testing of the online classified system was performed in three stages which are as follows :-

- Unit Testing
- Integration Testing
- System Testing

6.1 Unit Testing:

Unit testing is under taken when a module has been coded and successfully reviewed.

This can be done by two methods:

- Black Box testing
- Equivalence Class Partitioning

6.1.1.1 Black Box Testing:

Test cases are designed from an examination of the input/output values only and no knowledge of designing or coding is required the following are the two main approaches of designing black-box test cases.

6.1.1.2 Equivalence Class Partitioning:

The domain of input values to a program is partitioned into a set of equivalence classes. This partitioning is done on such way that the behaviour of the program is similar to every boundary value analysis. Boundary value analysis leads to selection of the test cases at the boundaries of different equivalence classes Testing done by: Team Member

In our project particularly, first we create the chart table for vaccination & then by running the application we conclude & tested that whether it runs properly or not. So such a way we perform the Unit testing & in this way we have done the testing to all the forms.

6.1.2 Integration Testing:

During integration testing different modules of the system are integrated using integration plan. The integration plan specifies the steps and the order in which modules are combined to realize the full system.

Purpose:
- To test whether the module performs its intended task. - Once all the modules have been integrated and tested, system testing can start.

6.1.3 System Testing:

System tests are designed to validate a fully developed system with a view to assuring that it meets its requirements. There are three types of system testing which are as follows :-

6.1.3.1 Alpha Testing:

The initial testing of a computer program or system under actual usage conditions, it can be done in-house by the vendor, or outside by a customer or third party teaser. –

Acceptance Testing performed by the customer in a controlled environment at the developer's site. The software used by the customer in a setting approximating the target environment with the developer observing and recording errors and usage problems.

6.1.3.2 Beta Testing:

Beta Testing is done after alpha testing. The main purpose of Beta testing are as follows:-

- Testing done by the potential or existing users, customers and end users at the external site without developers involvement is known as beta testing. - It is operation testing i.e. it tests if the software satisfies the business or operational needs of the customers and end users.

- Beta Testing is done for external acceptance testing of COTS(Commercial off the Shelf) software

6.2 Test Case Design:

Any engineered product (and most other things) can be tested in one of the following two ways.

Knowing the specified function that a product has been designed to perform, tests can be conducted to demonstrate each function is fully operational.

Knowing the internal working of a product, tests can be conducted to ensure that ―all gears mesh‖, that is the internal operation of the product performs according to the specification and all internal components have been adequately exercised

Different Methods of Testing:

6.2.1 White Box Testing:

White-box testing is a methodology used to ensure and validate the internal framework, mechanisms, objects and components of a software application. White-box testing verifies code according to design specifications and uncovers application vulnerabilities.
White-box testing is also known as transparent box testing, clear box testing, structural testing and glass box testing. Glass box and clear box indicate that internal mechanisms are visible to a software engineering team.

White-box testing advantages include:

- Enables test case reusability and delivers greater stability
- Facilitates code optimization
- Facilitates finding of the locations of hidden errors in early phases of development
- Facilitates effective application testing
- Removes unnecessary lines of code.

6.2.2 Regression Testing :

It is a type of software testing i.e. carried out by software testers as functional regression tests & developers as Unit Regression Tests.

Objective of regression tests are to find defects that got introduced to detect fixes or introduction of new features. Regression tests are ideal candidate for automation.

6.2.3 Accessibility Testing:

This is a formal type of software testing that helps to determine whether the software can be used by people with disability. There are also companies & consultants that provide website accessibility audits.

6.2.4 Ad-hoc Testing:

Ad hoc testing is an informal and improvisational approach to assessing the viability of a product. An ad-hoc is usually only conducted once unless a defect is found.

Commonly used in software development, ad hoc testing is performed without a plan of action and any actions taken are not typically documented. Testers may not have detailed knowledge of product requirements. Ad hoc testing is also referred to as random testing and monkey testing.

6.3 Tester:

Testing done by Members of the team Testing tool debugging

6.3.1 Validation:

User id & password cannot be blank while logging into the site. - In the edit profile page or the new user account page there are some mandatory fields like login id, password, vacancy no, name, resume title, company name etc. which cannot be left blank. - In the modify password page user have to specify the login id as well as the old password & the new password. - In the email id field _@„ characters are mandatory.

6.4 Design of Test Cases and Scenarios:

Test Cases:

The Test cases for this project are as follows:

Table 6.1 Test cases

S.NO	TEST CASE	EXPECTED OUTPUT	ACTUAL OUTPUT	RESULT
1		Safe to move	Safe to move	Pass
2		Safe to move	Intrusion detected	Fail
3		Intrusion detected	Intrusion detected	pass
4		Intrusion detected	Intrusion detected	Pass

6.5 Maintenance:

Maintenance of a typical software product requires much more effort than the effort necessary to develop the product itself. The relative effort of development of a typical software product to its maintenance effort is roughly in the 40:60 ratios.

Maintenance involves performing any one or more of the following three kinds of activities:

- Correcting errors that were not discovered during the product development phase. This is called _Corrective Maintenance,,.

- Improving the implementation of the system, and enhancing the functionalities of the system according to the customer's requirements. This is called _Perfective maintenance,,.

- Porting the software to work in a new environment. For example, porting may be required to get the software to work on a new computer platform or with a new operating system. This is called Adaptive Maintenance,,.

6.5.1 Further enhancements:

Due to the diversity of number plate, the process of recognition faced a problem. For the improvement of number plate recognition various authors used neural network model such as RBF neural network model, BP neural network model and SOM neural network model. In this paper present the review of number plate recognition based on different neural network model. The processing of number plate recognition is also very difficult due to background and noise. Due to the problem of recognition faced a problem of road security surveillance.

Chapter 7

CONCLUSION

7.CONCLUSION

7.1 SUMMARY

Vehicle Intrusion Detection plays an important role in numerous real-life applications, such as automatic toll collection, traffic law enforcement, parking lot access control, and road traffic monitoring. VID recognizes a vehicle's license plate number from the frames generated by the Live sreaming video. It is fulfilled by the combination of a lot of techniques, such as object detection, image processing, pattern recognition. The variations of the plate types or environments cause challenges in the detection and recognition of license plates.

7.2 FUTURE SCOPE

Today advances technology took License Plate Recognization (LPR) systems from hard to set up, limited expensive, fixed based applications to simple mobile ones in which "point to shoot" method can be used. This is possible because of the creation of software which ran on cheaper PC based and also non specialist hardware in which their no need to give pre- defined direction, angels, speed and size in which the plate would be passing the camera field of view. Also Smaller cameras which can read license plates at high speed, along with smaller, more durable processors that can fit in police vehicles, allowed law enforcement officers to patrol daily with the benefit of license plate recognition in real time.

Chapter 8

REFERENCES

[References]:

[1]: https://docs.python.org/3/

[2]: https://pymysql.readthedocs.io/en/latest/

[3]: https://openpyxl.readthedocs.io/en/stable/

[4]: https://docs.python.org/3/library/smtplib.html

[5]: https://docs.opencv.org/2.4/doc/tutorials/tutorials.html

[6]: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[7]: https://docs.python.org/3/library/tk.html

[8]: https://www.coursera.org/learn/machine-learning/

[9]: https://www.coursera.org/learn/neural-networks-deep-learning/home/welcome

[10]: https://www.coursera.org/learn/deep-neural-network/home/welcome

[11]: https://www.coursera.org/learn/machine-learning-projects/home/welcome

[12]: http://www.numpy.org/