A PROJECT REPORT ON

**INTRUSION DETECTION**

**SUBMITTED TO**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY COLLEGE OF ENGINEERING

**BY**

CH.B.Neeraj Singh

178297601003

In partial fulfilment of

**Internship Program of Bachelors of Technology**

**Computer Science and Engineering**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**UNIVERSITY COLLEGE OF ENGINEERING**

**ADIKAVI NANNAYA UNIVERSITY**

**December-2020**

**ADIKAVI NANNAYA UNIVERSITY**

**RAJAMAHENDRAVARAM**

**UNIVERSITY COLLEGE OF ENGINEERING**

**DECLARATION**

I, **CH.B.Neeraj Singh**, Reg.No.:**178297601003**, hereby declare that the project report entitled **Intrusion Detection System** done by me under the guidance of **Dr D.Latha** , Associate Professor, Department of Computer Science and Engineering, University College of Engineering, Adikavi Nannaya University, is submitted for the partial fulfilment of requirement for the award of the degree, Bachelors of Technology in Computer Science and Engineering in the academic year **2017-2021**.

**CH.B.Neeraj Singh**

**178297601003**

**ADIKAVI NANNAYA UNIVERSITY**

**RAJAMAHENDRAVARAM**

**UNIVERSITY COLLEGE OF ENGINEERING**

**CERTIFICATE**

This is to certify that the project report entitled **Intrusion Detection System** submitted by **Mr CH.B.Neeraj Singh** to University College of Engineering,AdikaviNannaya University, Rajamahendravaram, Andhra Pradesh, India, is a record of bonafide Project work carried out by her under my supervision and guidance and is worthy of consideration for the award of the degree of Bachelors of Technology in Computer Science and Engineering .

Head of Department Internal Guide

**ADIKAVI NANNAYA UNIVERSITY**

**RAJAMAHENDRAVARAM**

**UNIVERISTY COLLEGE OF ENGINEERING**

**ACKNOWLEDGEMENT**

I wish to express my deep sense of gratitude to the management of **K-Groups Software Solutions**, Kakinada, for giving me an opportunity to complete my **Intrusion Detection System** for the partial fulfilment of my degree in Bachelors of Technology in Computer Science and Engineering.

I would like express my gratitude to Mr.Madhavu, K-Groups, who has shown keen interest and patience and extended his valuable help during the project period. Also for his valuable suggestions and constant motivation that greatly helped the project work to get successfully completed.

**ABSTRACT**

An Intrusion Detection System  is a [software application](https://en.wikipedia.org/wiki/Software_application) that monitors a [network](https://en.wikipedia.org/wiki/Computer_network) or systems for malicious activity or policy violations. Any intrusion activity or violation is typically reported to an administrator. A Intrusion Detection System monitors the packets from the device and will alert the user or administrator if suspicious activity is detected. It takes a snapshot of existing system files and matches it to the previous snapshot.  It is used for detection of attacks by looking for specific patterns, such as byte sequences in network traffic, or known malicious instruction sequences used by malware.

Therefore, this project is carried out to build a prototype to detect the attack in network . Evolutionary methodology was implemented in this system pre-processing stages and training model

The Train and Test datasets are downloaded from the Kaggle dataset website. The train dataset contains over 125973 rows and 42 columns of data which is used train the required model. Two Machine Learning Classification algorithms named KNeighborsClassifier Model and Logistic Regression model are applied over the preprocessed data and the model got tested over 22544 rows and 42 columns present in the test data set. Finally the accuracy obtained using KNeighborsClassifier algorithm is 87% and over 84% of accuracy is obtained using Logistic Regression Algorithm.

**INDEX**

|  |  |  |
| --- | --- | --- |
| S.No. | Contents | Page no |
| 1. | Introduction  1.1Problem Statement  1.2 Objective  1.3 Scope and Applications | 7-8 |
| 2. | Software Requirements Specification  2.1 Functional Requirements  2.2 Non- Functional Requirements  2.3 Software Requirements  2.4 Hardware Requirements  2.5 Feasibility Study  2.5.a Technical Feasibility  2.5.b Operational Feasibility  2.5.c Economical Feasibility  2.5.d Schedule Feasibility | 9-10 |
| 3. | System Analysis and Design  3.1 Phases of License Plate Recognition  3.2 Flowchart- System Design  3.2.a System Design  3.2.b System Design  3.2.c Architecture | 11-12  13  14  15 |
| 4. | Implementation  4.1 Implementation tools  4.1.a Programming language and Coding tool  4.1.b Frame Work | 16-17  17-20 |
| 5. | Testing and Outputs | 21 |
| 6. | Conclusion | 22 |
| 7. | Bibliography | 23 |

**INTRODUCTION**

Now a day’s security become . Today we have video surveillance cameras in schools, hospitals and every other public place to make us feel secured. According to a survey by HIS it is estimated that there were around 245 million security cameras installed and functioning back on 2014, which is like having one security camera for every 30 people on this planet. With the advancement in technology especially in Image processing and Machine Learning, it is possible to make these cameras smarter by training them to process information from the Video feed.

License Plate Recognition is a computer vision technology to extract the license number of vehicles from images. It is an embedded system which has numerous applications and challenges. One of the notable contributions of the open source community to the scientific world is Java. Intel’s researches in Computer Vision bore the fruit called Open Computer Vision (OpenCV) library, which can support computer vision development.

License Plate Recognition was invented in 1976 at the Police Scientific Development Branch in Britain. Prototype systems were working by 1979, and contracts were awarded to produce industrial systems, first at EMI Electronics, and then at Computer Recognition Systems (CRS) in [Wokingham](https://en.wikipedia.org/wiki/Wokingham), UK. Early trial systems were deployed on the [A1 road](https://en.wikipedia.org/wiki/A1_road_(Great_Britain)) and at the [Dartford Tunnel](https://en.wikipedia.org/wiki/Dartford_Tunnel). The first arrest through detection of a stolen car was made in 1981. However, ANPR did not become widely used until new developments in cheaper and easier to use software were pioneered during the 1990s. The collection of ANPR data for future use (*i.e*., in solving then-unidentified crimes) was documented in the early 2000s. The first documented case of ANPR being used to help solve a murder occurred in November 2005, in [Bradford](https://en.wikipedia.org/wiki/Bradford), UK, where ANPR played a vital role in locating and subsequently convicting killers of [Sharon Beshenivsky](https://en.wikipedia.org/wiki/Murder_of_Sharon_Beshenivsky).

This system classify the Attack using multinomial classification.

**Problem Statement:**

Automatic vehicle license plate recognition is a key technique in most of traffic related applications and is an active research topic in the image processing domain. Different methods, techniques and algorithms have been developed for license plate detection and recognitions. Automatic car license plate recognition (CLPR) system is implemented to help the human to automatically detect plate number without human supervision. Previously, human is needed to observe and list the user car plate number manually. So this project project is developing to replace human to monitor the car and automatically capture the image.

**Objectives:**

1. To develop a model to detect intrusion.

2. To experiment machine learning algorithm in the Cyber security domain.

3. To classify the type of Attacks.

**Scope and Applications:**

The scope of this system is to tackle with the problems that can arise in day to day life. Some of the scopes are:

1.The system can be used in parking areas and parking fee can be calculated based on time.

2.The system can be used for access control. A gate automatically opens for authorized members in a secured area, thus replacing or assisting the security gaurd. The events are logged on a database and could be used to search the history of events.

3. The system can be installed at Toll collection areas. The car number is used to calculate the travel fee in a toll-road, or used to double-check the ticket.

4. The system can also be used for Border Control. This installation covers the borders of the entire Country. Each vehicle is registered into a central database and linked to additional information such as the passport data. This is used to track all border crossings*.*

5. This system can be used for retrieving stolen cars.

6. This system can help in calculation of fines in case of violation of speed or crossing of traffic signals.

7. This system can be used to serve the purpose of marketing**.** The car plates may be used to compile a list of frequent visitors for marketing purposes, or to build a traffic profile (such as the frequency of entry verses the hour or day).

**SOFTWARE REQUIREMENTS SPECIFICATION**

**Functional Requirements:**

 A Functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between outputs and inputs. Some of the Functional components are Information source, Analysis and Response.

**Non-Functional Requirements:**

Nonfunctional requirements are requirements that are not directly concerned with the specified function delivered by the system. Some of the non-functional requirements are usability, reliability

**Software Requirements:**

Following are the software requirement necessary of the project:

a) Python programming language

b) Jupyter Notebook (Python editor)

d) Windows OS

**Hardware Requirements:**

Following are the hardware requirement that is most important for the project:

a) Fluently working Laptops (64 bit preferable)

b) RAM minimum 6GB

**Feasibility Study:**

Before starting the project, feasibility study is carried out to measure the viable of the system. Feasibility study is necessary to determine if creating a new or improved system is friendly with the cost, benefits, operation, technology and time. Following feasibility study is given as below:

**a) Technical Feasibility-**

Technical feasibility is one of the first studies that must be conducted after the project has been identified. Technical feasibility study includes the hardware and software devices. The required technologies (Python language and Python IDLE IDE) existed.

**b) Operational Feasibility-**

Operational Feasibility is a measure of how well a proposed system solves the problem and takes advantage of the opportunities identified during scope definition. The following points were considered for the project’s technical feasibility:

a. The system will detect and capture the image of License Plate of Car.

b. The captured image is then processed to give characters on license plate.

**c) Economic Feasibility-**

The purpose of economic feasibility is to determine the positive economic benefits that include quantification and identification. The system is economically feasible due to availability of all requirements such as collection of data from internet .

**d) Schedule Feasibility-**

Schedule feasibility is a measure of how reasonable the project timetable is. The system is found schedule feasible because the system is designed in such a way that it will finish with in prescribed time.

**SYSTEM DESIGN AND ANALYSIS**

**Phases of License Plate Recognition:**

License Plate Recognition System is trained using supervised learning approach in which it takes images of cars. The system includes the training and testing phase followed by Image Acquisition, Image Pre-Processing, Character Segmentation and Prediction. The model is trained using SVC (support vector classifiers). Phases in license plate recognition are outlined as below:

**a)Image Acquisition:**

Images used for license plate recognition are static images or image sequences. Images of car can be captured using camera.

**b)Image Pre-processing:**

Image pre-processing includes the preparing the image for license plate recognition. There are two steps in this:

a)Converting the input car image to a Grayscale image

b)Converting the Grayscale image to a Binary image

**c)Detect License Plate:**

The approach used for detection of license plate is Connected Component Analysis(CCA). Connected regions imply that all the connected pixels belong to the same object. A pixel is said to be connected to another if they both have the same value and are adjacent to each other. CCA is applied on this binary image of car using measure and regionprops module of python. measure.label method labels all the connected regions of binary car image and returns a labelled image. Assumptions are made that width of the license plate region to the full image ranges between 15% and 40% and height of the license plate region to the full image is between 8% and 20%.Regionprops method is applied on labelled image which returns a list of regions along with their properties like area, bounding box etc. The bounding box coordinates are obtained and are then compared with the maximum and minimum dimensions of plate (according to assumption). Once the labelled region satisfies the dimension constraints, it is pushed into a list. This list will hold the objects which are classified by our code as similar to license plate*.*

**d)Character Segmentation and Prediction:**

Output of above step is a license plate image detected with bounding box in red in a car image. This is provided as input to this step and CCA is applied on this image to bound the characters in plate. Each character identified is appended into a list.Here similar assumptions are made for the size of characters. Once characters are segmented, they are resized to a size of 20px X 20px and are pushed into a list.

Once the characters of plate is obtained, then the model is loaded which predicts the characters.

**e)Training of model:**

Model is trained using SVC (4 cross-fold validation) on dataset of images of characters with size 20px X 20px. In the training of model, first the 20px X 20px images are flattened to image array of 1 X 400, then it is trained by using SVC and cross validation is used to check the accuracy of the model.

**f)System Evaluation:**

Evaluation of the system can be done using following methods:

1. Precision-83.80% b) F-score-93.70%

**System Design:**

Input image/video

**a)**

Grayscale Image

Yes

License Plate Detected

Compare ratio of height and width with the standard

Train ML model

Each character is identified and appended into the list

Predict Characters (output)

Error

Find maximum Area of the image

Apply Connected Component Analysis for character recognition

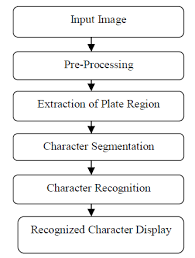
No

Apply Connected Component Analysis

Binary Image

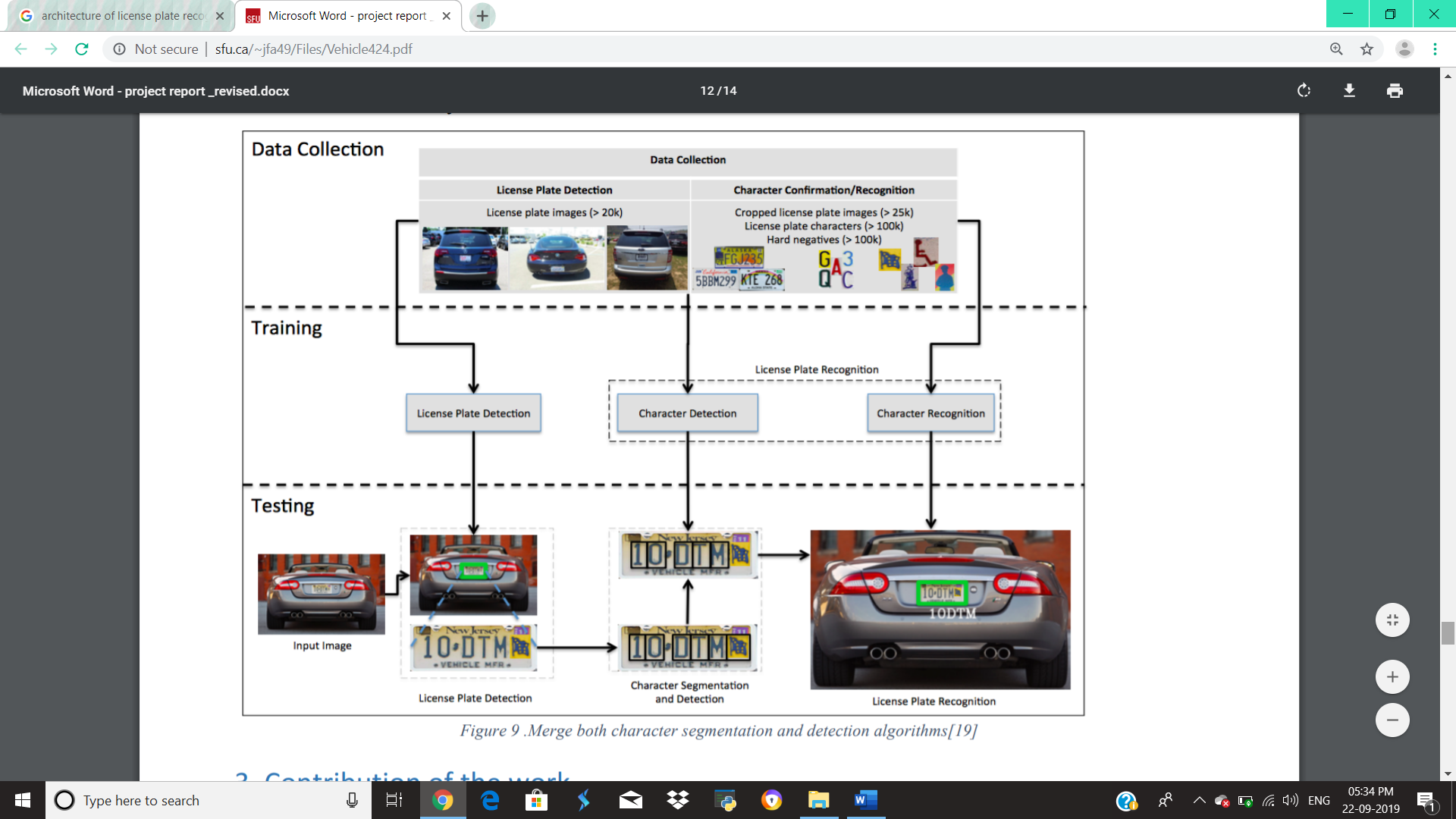
**System Design:**

**b)**

****

**Architecture:**

**c)**



10DTM

Character prediction

**IMPLEMENTATION**

**Implementation Tools:**

**Programming Language and Coding Tools**

**a)Python –** Python is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is [dynamically typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [procedural](https://en.wikipedia.org/wiki/Procedural_programming), object-oriented, and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

Python was conceived in the late 1980s as a successor to the [ABC language](https://en.wikipedia.org/wiki/ABC_(programming_language)). system capable of collecting [reference cycles](https://en.wikipedia.org/wiki/Reference_cycle). Python 3.0, released 2008, was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility), and much Python 2.0, released 2000, introduced features like [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a [garbage collection](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science))on 2 code does not run unmodified on Python 3. Due to concern about the amount of code written for Python 2, support for Python 2.7 (the last release in the 2.x series) was extended to 2020. Language developer Guido van Rossum shouldered sole responsibility for the project until July 2018 but now shares his leadership as a member of a five-person steering council.

Python [interpreters](https://en.wikipedia.org/wiki/Interpreter_(computing)) are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). A global community of programmers develops and maintains [CPython](https://en.wikipedia.org/wiki/CPython), an [open source](https://en.wikipedia.org/wiki/Open-source_software) [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation). A [non-profit organization](https://en.wikipedia.org/wiki/Nonprofit_organization), the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation), manages and directs resources for Python and CPython development.

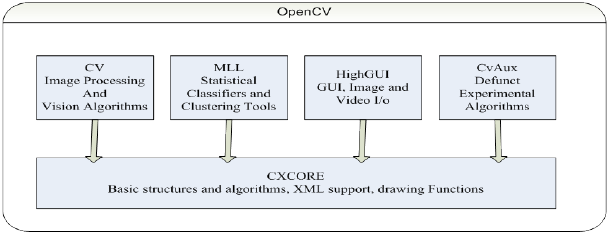
**b)Python IDLE IDE**- IDLE (short for integrated development environment or integrated development and learning environment) is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) for [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), which has been bundled with the default implementation of the language since 1.5.2b1. It is packaged as an optional part of the Python packaging with many [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions). It is completely written in Python and the [Tkinter](https://en.wikipedia.org/wiki/Tkinter) GUI toolkit ([wrapper](https://en.wikipedia.org/wiki/Wrapper_function) functions for [Tcl](https://en.wikipedia.org/wiki/Tcl)/[Tk](https://en.wikipedia.org/wiki/Tk_(framework))).

IDLE is intended to be a simple [IDE](https://en.wikipedia.org/wiki/Integrated_development_environment) and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

**Framework:**

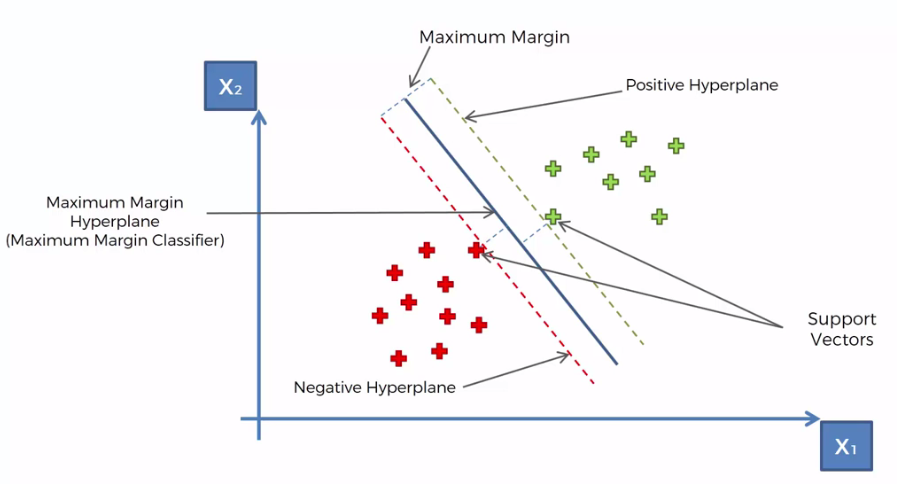
**a)OpenCV:**

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code. The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3Dmodels of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. It has C++, C, Python, Java and MATLAB interfaces and supports Windows, Linux, [Android](http://opencv.org/platforms/android.html) and Mac OS.



**b)Support Vector Classifiers (SVC):**

The maximal margin classifier is a very natural way to perform classification, is a separating hyperplane exists. However the existence of such a hyperplane may not be guaranteed, or even if it exists, the data is noisy so that maximal margin classifier provides a poor solution. In such cases, the concept can be extended where a hyperplane exists which almost separates the classes, using what is known as asoft margin. The generalization of the maximal margin classifier to the non-separable case is known as the support vector classifier, where a small proportion of the training sample is allowed to cross the margins or even the separating hyperplane. Rather than looking for the largest possible margin so that every observation is on the correct side of the margin, thereby making the margins very narrow or non-existent,some observations are allowed to be on the incorrect side of the margins. The margin is softas a small number of observations violate the margin. The softness is controlled by slack variables which control the position of the observations relative to the margins and separating hyperplane. The support vector classifier maximizes a soft margin.



Algorithm :

Input: Set of data points

Output: Hyper Plane classifying the points

Steps:

Step 1: *A hyperplane is drawn that best classifies the data into classes and assume that all the data is perfectly classified.*

Step 2: *Calculate the perpendicular distance from the closest data points to the hyperplane. This is known as margin. An optimal hyperplane should be such that the margin is maximum.*

Step 3: *For a new data point, to be classified into the class to which it belongs, consider the following equation*

, i = 1, 2, 3,….

*This is an equation that involves calculating the inner products of a new input vector (x) with all support vectors in training data. The coefficients and (for each input) must be estimated from the training data by the learning algorithm.*

Step 3.a: *If (is greater than zero), then the new data point(input) is above the plane.*

Step 3.b: *If (is less than zero), then the new data point(input) is below the plane*.

Step 3.c: *A value close to the line returns a value close to zero and the point may be difficult to classify.*

Step 3.d: *If the magnitude of the value is large, the model may have more confidence in the prediction.*

Step 4: *The best or optimal plane is the one that has the maximum margin.*

**c)Connected Component Analysis (CCA):**

Connected Component Analysis is an algorithmic application of graph theory, where subsets of connected components are uniquely labelled. Connected component labelling is used in computer vision to detect connected regions in binary digital images.

Any set of pixels which are not separated by a boundary are called Connected. Each maximal region of connected components is called Connected Component. The set of connected components partition an image into segments. Image segmentation is an useful operation in many image processing applications.

In text detection, connected component plays a vital role. Each connected component ensures that the underlying pixels forms a single character.  This helps in identification of characters.

Connected component labelling works by scanning an image, pixel-by-pixel (from top to bottom and left to right) in order to identify connected pixel regions, *i.e.* regions of adjacent pixels which share the same set of intensity values.

Algorithm:

Input: Binary or Grayscale image

Output: Connected regions

Steps:

The algorithm works in two passes.

Step 1: *In the first step, scan the image pixel by pixel (left to right, top to bottom).*

Step 2: *If the pixel is background pixel, then leave it and move on to the next pixel.*

Step 3: *If the pixel is not a background pixel, then*

Step 3.a: *If none of the neighbors of this pixel are labelled, assign a new label to this pixel and enter it in the equivalence table.*

Step 3.b: *If only one of it’s upper and left neighbors has a label, then assign that label to the current pixel.*

Step 3.c: *If both upper and left neighbors have the same label, then assign that to the current pixel.*

Step 3.d: *If both upper and left neighbors have different labels, then assign the minimal label to the current pixel. This minimal label becomes the parent to the higher label. And the labels in the equivalence table as equivalent labels.*

Step 4: *If there are no more pixels to consider, then go to Step 1.*

Step 5: *After completing the scan, the equivalent label pairs are sorted into equivalence classes and a unique label is assigned to each class. Find the minimal label in the equivalence table.*

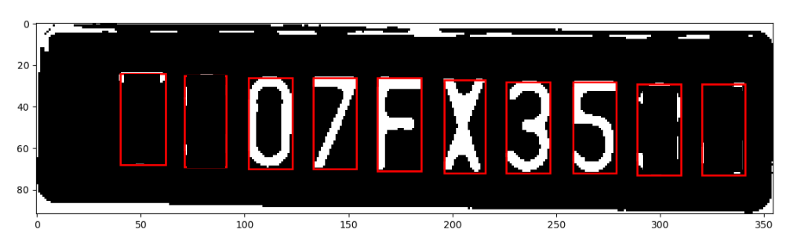
Step 6: *Scan the image again. Replace each label by the lowest label in it’s equivalent set(each label is replaced by the label assigned to its equivalence classes).*

**System Testing:**

System testing was done by giving different training and testing datasets. This test was done to evaluate whether the system was predicting accurate result or not. During the phase of the development of the system our system was tested again and again. The series of testing conducted are as follows:

Unit testing, integration testing, alpha testing, beta testing.

**Output:**

****

Output showing segmentation of characters after applying connected component analysis

**CONCLUSION**

The future lies in the hand of AI by this statement we can understand that everything will be overpowered and taken under control by machines and working on image processing gives a wider scope to look through multiple aspects that benefits the future. The multiple applications of it on smart parking, toll areas charging, plate detection to sum up for various surveys, track for emergency help during medical breakdown and tracking vehicles based on license plate detection makes it a larger beneficial source to work upon and make it benefits user applicable at ground usage. Using machine learning algorithms for better accuracy and more improvised detection will give way to new image processing techniques overall that is going to be a lot more beneficial. This project can detect vehicle number plate and can extract number from it. This system isidentifiescharacters in the license plate based on training dataset which consists of characters(i.e., Alphabets(A-Z),digits(0-9)). But there are certain assumptions while doing this project. The system now is efficient more than before to run down into many applications for navigation, smart parking and detection. The architecture thus designed is much simple and robust in its use. This System gave accuracy score of 93 percent which was relatively better as faster compared to existing modules. License Plate Recognition is a very challenging problem. More efforts should be made to improve performance for important applications. Our future work will focus on improving the performance of the system and deriving more appropriate results which maybe useful in many real applications.

**BIBLIOGRAPHY**

Code- <https://github.com/apoorva-dave/LicensePlateDetector>

OpenCV-<https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html>

Connected Component Analysis-<https://docs.opencv.org/3.4/d3/dc0/group__imgproc__shape.html>

Support Vector Classifier-<https://docs.opencv.org/3.4/d1/d73/tutorial_introduction_to_svm.html>