# TOLL COLLECTION SYSTEM USING IMAGE PROCESSING

***A Project report submitted in partial fulfillment of the requirements for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted by**


P.S.S.Hrithik (317126512155)
Sushma. I (318126512L36)
Ganesh Karthikeya .M (317126512142)
Meghana. T (318126512L34)
Anurag prudhvi. K (316126512081)

**Under the guidance of**

**Dr. J Bhaskar Rao,** M.E, Ph.D

**Assistant Professor, Department of ECE**

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES

(UGC AUTONOMOUS)

*(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade*)

Sangivalasa, Bheemili mandal, Visakhapatnam dist. (A.P)


2020-2021

**DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING**

**ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES**
**(UGC AUTONOMOUS)**
***(Permanently Affiliated to AU, Approved by AICTE and Accredited by NBA & NAAC with 'A' Grade)***
**Sangivalasa, Bheemili Mandal, Visakhapatnam dist. (A.P)**

PRAGNANAM BRAHMA

**ANITS**

## CERTIFICATE

*This is to certify that the project report entitled* **"TOLL COLLECTION SYSTEMS USING IMAGE PROCESSING"** submitted by **P.S.S.Hrithik (317126512155)** , **M.S.S.Ganesh Karthikeya (317126512146)** , **I.Sushma (318126512L36)** , **T.Meghana (318126512L34)** , **K.Anurag (316126512081)** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics & Communication Engineering** of Andhra University, Visakhapatnam is a record of bonafide work carried out under my guidance and supervision.

**Project Guide**

**Dr. J. Bhaskar Rao** M.E, Ph.D

Assistant Professor

Department of ECE

ANITS

Assistant Professor
Department of E.C.E
Anil Neerukonda
Institute of Technology & Sciences
Sangivalasa, Visakhapatnam-531 162

**Head of the Department**

**Dr. V Rajyalakshmi** M.E, Ph.D., MHRM, MIEEE, MIMIETE

Professor & HOD

Department of E.C.E

ANITS

Head of the Department
Department of E C E
Anil Neerukonda Institute of Technology & Sciences
Sangivalasa - 531 162

# ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Dr**. **J. Bhaskar Rao Assistant Professor, M.E, Ph.D**. Department of Electronics and Communication Engineering, ANITS, for his guidance with unsurpassed knowledge and immense encouragement. We are grateful to **Dr. V. Rajyalakshmi**, Head of the Department, Electronics and Communication Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the **Principal and Management, ANITS, Sangivalasa,** for their encouragement and cooperation to carry out this work.

We express our thanks to all **teaching faculty** of Department of ECE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank **all non-teaching staff** of the Department of ECE, ANITS for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last, but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.



**PROJECT STUDENTS:**

| | |
|---|---|
| P.S.S.Hrithik | (317126512155) |
| Sushma. I | (318126512L36) |
| Ganesh Karthikeya .M | (317126512142) |
| Meghana. T | (318126512L34) |
| Anurag prudhvi. K | (316126512081) |

# CONTENTS

**ABSTRACT**

## CHAPTER 1: INTRODUCTION — PAGE NUMBER

1.1 RFID TECHNOLOGY 1

1.1.1 APPLICATIONS OF RFID 4

1.2 RFID ROLE IN FASTAG 5

1.2.1 BENEFITS OF FASTAG 7

1.3 LITERATURE SURVEY 7

1.4 ORGANIZATION OF PROJECT REPORT 9

## CHAPTER 2: METHODOLOGY

2.1 INTRODUCTION 10

2.2 FLOW CHART 11

## CHAPTER 3: IMAGE PRE-PROCESSING

3.1 CONVERSION OF RGB TO GRAY SCALE IMAGE 13

3.2 SMOOTHING FILTERS 15

3.2.1 AVERAGE BLURRING 15

3.2.2 GAUSSIAN FILTER 16

3.2.3 BILATERAL FILTER 18

3.3 GRAY TO BINARY IMAGE 21

3.3.1 EDGE DETECTION 21

3.3.1.1 SOBEL EDGE DETECTION 22

3.3.1.2 LAPLACIAN OF GAUSSIAN(LoG) 22

3.3.1.3 CANNY EDGE DETECTION 23

3.3.2 THRESOLDING 30

3.3.3 EDGE DETECTION Vs THRESOLDING 35

3.4 CONTOURS 35

## CHAPTER 4: SOFTWARE DESCRIPTION


| | | |
|---|---|---|
| 4.1 | SOFTWARE TOOLS USED | 38 |
| 4.2 | OVERVIEW OF SOFTWARE USED | 38 |
| 4.2.1 | HTML | 38 |
| 4.2.2 | CSS | 39 |
| 4.2.3 | REACT | 40 |
| 4.2.4 | PYTHON | 42 |
| 4.2.5 | FLASK | 43 |
| 4.2.6 | PHP | 43 |
| 4.2.7 | MY SQL | 44 |
| 4.3 | WEB DEVELOPMENT | 45 |
| 4.4 | APP LAYOUT | 49 |
| 4.5 | DATABASE DETAILS | 55 |


## CHAPTER 5: SIMULATED RESULTS


| | | |
|---|---|---|
| 5.1 | IMAGE PROCESSING RESULT | 58 |
| 5.2 | MAIN RESULT | 63 |

**CHAPTER 6: CONCLUSION** 64

**REFERENCES** 65

# LIST OF FIGURES

**FIGURES** **PAGE NUMBER**


Fig 1.1 Binary tree method of identifying an RFID tag 4
Fig 1.2 Fastag 6
Fig 1.3 Toll collection using fastag 6
Fig 2.1 Flowchart of Image Pre-Processing 11
Fig 2.2 At server side 12
Fig 2.3 At user side 12
Fig 3.1 Weighted grayscale and luminance corrected grayscale image 13
Fig 3.2 compressed vs uncompressed luminance 15
Fig 3.3 1-D Gaussian distribution with mean 0 and $\sigma$ =1 17
Fig 3.4 The shape of a Gaussian filter with mean (0,0) 17
Fig 3.5 Cases in Bilateral filter 19
Fig 3.6 Spatial and range components of a Bilateral filter 20
Fig 3.7 Bilateral filtering to an image 20
Fig 3.8 Original image 24
Fig 3.9 Gaussian filter applied 24
Fig 3.10 Gradient of the image 25
Fig 3.11 Non-maximum suppressed image 27
Fig 3.12 Double thresholded image 28
Fig 3.13 Canny edge detection of the image 29
Fig 3.14 Thresholded image with two thresholds 30
Fig 3.15 Histogram of the image 31
Fig 3.16 Gaussian distribution of the image 32
Fig 3.17 Histogram using Optimal threshold. 33
Fig 3.18 variance vs threshold for otsu's model 34
Fig 3.19 Topological analysis by border 36
Fig 3.20 Surroundness around connected components 36
Fig 3.21 Contours of the image 37
Fig 4.1 Worker Login Page 45

| | |
|---|---|
| Fig 4.2 Toll plaza Index page | 46 |
| Fig 4.3 ADD-USER PAGE | 47 |
| Fig 4.4 Linking-Vehicle Layout | 48 |
| Fig 4.5 Login page layout | 49 |
| Fig 4.6 Content layout | 50 |
| Fig 4.7 Dashboard layout | 51 |
| Fig 4.8 User’s Profile layout | 52 |
| Fig 4.9 Customers Transactions | 53 |
| Fig 4.10 Notification layout | 54 |
| Fig4.11 Customers Table | 55 |
| Fig 4.12 Employee Table | 55 |
| Fig 4.13 Transaction Table | 56 |
| Fig 4.14 Vehicle Information Table | 56 |
| Fig 4.15 Rejected Transaction Table | 57 |
| Fig 5.1 vehicle1 | 58 |
| Fig 5.2 vehicle 2 | 58 |
| Fig 5.3 Gray-Scale image of vehicle1 | 59 |
| Fig 5.4 Gray-Scale image of vehicle2 | 59 |
| Fig 5.5 Edge detection for vehicle1 | 60 |
| Fig 5.6 Edge detection for vehicle2 | 60 |
| Fig 5.7 All possible Contours in the vehicle1 | 61 |
| Fig 5.8 All possible Contours in the vehicle2 | 61 |
| Fig 5.9 Required contour of vehicle1 | 62 |
| Fig 5.10 Required contour of vehicle2 | 62 |
| Fig 5.11 Transaction table in the Database | 63 |
| Fig 5.12 Rejected Transaction table in the Database | 63 |

# ABSTRACT

In India, from manual collection to electronic collection, drastic changes had been made at toll gates. The risks due to cash are high compared to digital money. The introduction of a digital payment system laid a foundation for the present toll collection system. In FASTAG, RFID technology is used for toll collection at national highways. RFID tag plays a key role in the automated deduction of toll charges. FASTAG main objectives are traffic control and time-saving. But there are few disadvantages, as FASTAG uses an RFID tag, which can be cloned, so there is a chance for misusing it and it is also somewhat inconvenient to the user as they have to renew the tag every 3 years for which they have to pay for it. In case of loss of FASTAG user has to wait for a whole day for new tag and they have to pay for it. In this paper, we implemented toll collection using image processing technology which doesn't require RFID scanning devices, RFID tags. The account details of the user are stored in the database using the user's vehicle number plate as a key. When a vehicle is passed through the tollgate, license plate details are retrieved using image processing. The transaction can be done by retrieving account details from database using the vehicle's number plate details. Using the Open Computer Vision (Open CV) and Optical Character Recognition (OCR), the system can extract characters from the captured vehicle image.

***Keywords: Electronic payment, image processing, toll collection, open CV, OCR***.

**Team Members:**

Shanmukh Hrithik.P. S (317126512155)
Sushma. I (318126512L36)
Ganesh Karthikeya .M (317126512142)
Meghana.T (3181265121L34)
Anurag prudhvi. K (316126512081)

**Project Guide:**

**Dr. J. Bhaskar Rao** Assistant Professor, M.E, Ph.D

# CHAPTER 1

# INTRODUCTION

## 1.1RFID TECHNOLOGY:

Radio-frequency identification uses electromagnetic field to automatically identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data, usually an identifying inventory number, back to the reader. This number can be used to track inventory goods. RFID tags are made out of three pieces: a microchip (an integrated circuit which stores and processes information and modulates and demodulates (RF) signals), an antenna for receiving and transmitting the signal and a substrate. The tag information is stored in a non-volatile memory. The RFID tag includes either fixed or programmable logic for processing the transmission and sensor data, respectively. RFID tags can be either passive, active or battery-assisted passive. An active tag has an on-board battery and periodically transmits its ID signal. A battery-assisted passive tag has a small battery on board and is activated when in the presence of an RFID reader. A passive tag is cheaper and smaller because it has no battery; instead, the tag uses the radio energy transmitted by the reader. However, to operate a passive tag, it must be illuminated with a power level roughly a thousand times stronger than an active tag for signal transmission. This makes a difference in interference and in exposure to radiation. Tags may either be read-only, having a factory-assigned serial number that is used as a key into a database, or may be read/write, where object-specific data can be written into the tag by the system user. Field programmable tags may be write-once, read-multiple; "blank" tags may be written with an electronic product code by the user. The RFID tag receives the message and then responds with its identification and other information. This may be only a unique tag serial number, or may be product-related information such as a stock number, lot or batch number, production date, or other specific information. Since tags have individual serial numbers, the RFID system design can discriminate among several tags that might be within the range of the RFID reader and read them simultaneously.

Signalling between the reader and the tag is done in several different incompatible ways, depending on the frequency band used by the tag. Tags operating on LF and HF bands are, in terms of radio wavelength, very close to the reader antenna because they are only a small percentage of a wavelength away. In this near field region, the tag is closely coupled electrically with the transmitter in the reader. The tag can modulate the field produced by the reader by changing the electrical loading the tag represents. By switching between lower and higher relative loads, the tag produces a change that the reader can detect. At UHF and higher frequencies, the tag is more than one radio wavelength away from the reader, requiring a different approach. The tag can backscatter a signal. Active tags may contain functionally separated transmitters and receivers, and the tag need not respond on a frequency related to the reader's interrogation signal.

An Electronic Product Code (EPC) is one common type of data stored in a tag. When written into the tag by an RFID printer, the tag contains a 96-bit string of data. The first eight bits are a header which identifies the version of the protocol. The next 28 bits identify the organization that manages the data for this tag; the organization number is assigned by the EPC Global consortium. The next 24 bits are an object class, identifying the kind of product. The last 36 bits are a unique serial number for a particular tag. These last two fields are set by the organization that issued the tag. Rather like a URL, the total electronic product code number can be used as a key into a global database to uniquely identify a particular product. Often more than one tag will respond to a tag reader, for example, many individual products with tags may be shipped in a common box or on a common pallet. Collision detection is important to allow reading of data. Two different types of protocols are used to "singulate" a particular tag, allowing its data to be read in the midst of many similar tags. In a slotted Aloha system, the reader broadcasts an initialization command and a parameter that the tags individually use to pseudo-randomly delay their responses. When using an "adaptive binary tree" protocol, the reader sends an initialization symbol and then transmits one bit of ID data at a time; only tags with matching bits respond, and eventually only one tag matches the complete ID string.

"Bulk reading" is a strategy for interrogating multiple tags at the same time, but lacks sufficient precision for inventory control. A group of objects, all of them RFID tagged, are read completely from one single reader position at one time. Bulk reading is a possible use of HF (ISO 18000-3), UHF (ISO 18000-6) and SHF (ISO 18000-4) RFID tags. However, as tags respond strictly sequentially, the time needed for bulk reading grows linearly with the number of labels to be read. This means it takes at least twice as long to read twice as many labels. Due to collision effects, the time required is greater. A group of tags has to be illuminated by the interrogating signal just like a single tag. This is not a challenge concerning energy, but with respect to visibility; if any of the tags are shielded by other tags, they might not be sufficiently illuminated to return a sufficient response. The response conditions for inductively coupled HF RFID tags and coil antennas in magnetic fields appear better than for UHF or SHF dipole fields, but then distance limits apply and may prevent success. Under operational conditions, bulk reading is not reliable. Bulk reading can be a rough guide for logistics decisions, but due to a high proportion of reading failures, it is not suitable for inventory management. However, when a single RFID tag might be seen as not guaranteeing a proper read, multiple RFID tags, where at least one will respond, may be a safer approach for detecting a known grouping of objects. In this respect, bulk reading is a fuzzy method for process support. From the perspective of cost and effect, bulk reading is not reported as an economical approach to secure process control in logistics.

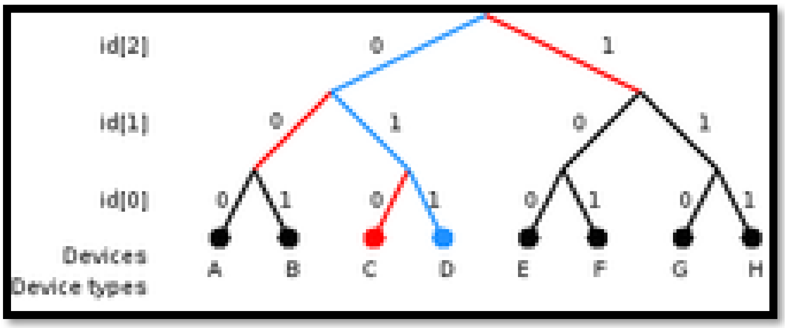



Fig 1.1Binary tree method of identifying an RFID tag

### 1.1.1 APPLICATIONS OF RFID TECHNOLOGY:

1. Access management
2. Tracking of goods
3. Tracking of persons and animals
4. Toll collection and contactless payments
5. Machine readable travel documents
6. Locating lost airport baggage
7. Tracking and billing processes
8. Monitoring the physical state of perishable goods

## 1.2 RFID ROLE IN FASTAG

The RFID technology uses an Electronic Produce Code (EPC) through which every vehicle can be uniquely identified. This code is different from the vehicle's registration number and exclusive to it on a global scale. Each EPC code, which is a 13-digit number, in the RFID-FASTAG is issued by GS1 India, a standards body, which ensures that each code is unique and in sync with the global standards put in place, in order for correct product identification. Which in the case of FASTAG, is a vehicle. The code needs to be standardized in order to ensure that the data coded inside is not read differently at different levels. However, unlike barcoding, which uses a pattern of black bars and white spaces, in which the information is coded, an RFID tag uses a small electronic chip for the same which is surrounded by an antenna. Also, unlike the barcode, an RFID tag does not need to be very close to the reader or, even in the line of sight of the same. One just simply has to be within a reading distance from the scanner. A FASTAG has what is called a passive RFID chip as it does not contain its own battery. It is energized only when the beam from the scanner strikes it. Whenever the vehicle passes through the ETC lane of the Toll Plaza, the Toll Plaza system captures the FASTAG details like (Tag ID, TID, Vehicle class, etc.) and sends it to the Acquiring bank for processing. The Acquiring bank sends a request to the NETC Mapper to validate the tag details. Once the Tag ID is validated, NETC Mapper responds with details like Vehicle class, VRN, Tag Status etc. If the Tag ID is absent in NETC Mapper, it will respond that the Tag ID is not registered. After successful validation of Tag ID from NETC Mapper, acquirer host calculates the appropriate toll fare and initiate a debit request to NETC system. NETC System will switch the debit request to the respective issuer bank for debiting the account of the customer. Issuer host shall debit the linked tag holder account and sends a SMS alert to the tag holder. The Issuer host shall send the response message to NETC system. If the response is not sent within the defined TAT, the transaction are considered as Deemed

Accepted. NETC system will notify the response to acquirer host. Acquirer host will notify to respective toll plaza system.

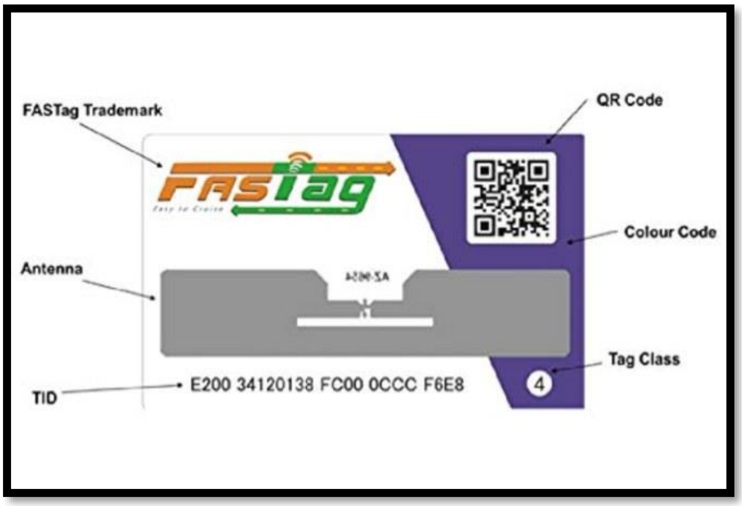



Fig 1.2 Fastag


Fig 1.3 Toll collection using fastag

### 1.2.1 BENEFITS OF FASTAG

1. Ease of payment No need to carry cash for the toll transactions saves time.
2. Near non-stop movement of vehicles leading to lower fuel cost.
3. Online Recharge FASTAG can be recharged online through Credit Card / Debit Card / NEFT/ RTGS or Net banking.
4. Environmental benefit like reduced air pollution and less use of paper is beneficial.
5. Economic benefit like the reduced effort in management at the toll plaza and the reduced effort in monitoring centrally is also beneficial.
6. Social benefit like less time taking toll payment is hassle-free and analytics for better highway management.

## 1.3 LITERATURE SURVEY:

Typical VNR System consists of four modules: image acquisition, license plate extraction, character segmentation, and character recognition. The efficiency & accuracy of the system largely depends on the second module & various approaches have been used for this purpose. There are several common searching algorithms to locate vehicle license plate. Searching algorithm rely on color information [2]. In this method a color search algorithm is used to extract the likelihood ROI in an image [2]. These algorithms are usually fast but can detect only single colored standardized number plate. High license plate extraction rate is achieved in [5], [6] based on vertical edging and mathematical morphology operations; because of having vertical edges in English characters & digits, they can be easily classified. Several algorithms also utilize neural networks for license plate extraction [4]. There are also some algorithms designed for recognizing the number plates of Indian vehicles [2][5]. The system [2] utilizes color searching algorithms and effectively detects the number plates of Sindh only. The other systems [5] rely on the width to height ratio of the standard number plate & matches input vertical edges with that ratio for extracting number plates. Presently, there are several common algorithms for the segmentation of license plate characters, such as segmentation through dilation, template matching & projection analysis. In the segmentation through dilation, characters of number plate are dilated vertically for separating each character & smearing algorithm is used for finding character region [7], the license plate characters are also segmented by drawing vertical projection of number plate & finding the region of each character [8]. This algorithm is simple and rapid, but can create problem if the plate have dots or image is noisy. In this system [9] They developed the electronic toll collection system using barcode laser technology. This system uses barcode. Initially user information embedded on barcodes which are mounted on the number plate of vehicle. When the vehicle passed through the toll booth, information is read by barcode reader and toll is deducted from the user's account. This system make the payment without using the cash manually and save time. The disadvantage of this system is that it takes more time for sending the signal to barcode and retrieving the information from Barcode. In this system [10] they used RFID technique in which vehicle is identified with the help of radio frequency. Each vehicle has the RFID tag. And this RFID tag has an unique identification number which is assigned by the RTO or traffic governing authority. System will store all the basic information of the owner, and the amount that has to be paid for the toll tax along with this number. The RFID reader will be placed strategically at the toll plaza. When the vehicle crosses the toll plaza, the toll amount based on vehicle type will be deduced from user's account balance and new account balance will be updated. If the user has insufficient balance, they uses the alarm which will alert the authority that the vehicle have insufficient balance and that particular vehicle can be trapped. The advantage of this system is RFID tag cannot be cloned so cannot be cheated and it is less costly. The disadvantage of this system is RFID tag are vulnerable to electro static discharge damage.

## 1.4ORGANIZATION OF PROJECT REPORT:

This project report is organized in 6 chapters. The current chapter introduce the project and brief description on toll collection system. Chapter 2 provides information about methodology followed. Chapter 3 describes the image processing techniques and their advantages. Chapter 4 provides information some of the existed software tools and tools used in project. Chapter 5 contains simulated results and performance characteristics of images. Chapter 6 gives conclusion of this project.

# CHAPTER-2

# METHODOLOGY

## 2.1 INTRODUCTION:

We are introducing a new method for deducting toll fares at toll plazas which is more efficient than the existed models. Automation is a technology where human interference can be reduced. Identifying vehicles automatically has been preferred because of its applications: Toll payments, ticket issuing, theft control, traffic surveillance, etc. Highways or Toll Roads are provided to control the traffic and increase the mobility of vehicles. So manual collection of toll fare results in time delay, inessential fuel consumption, traffic congestion. At present, RFID technology is implemented in the field of toll payments. The RFID tag present in the FASTAG uses an Electronic Produce Code (EPC) so that every vehicle can be uniquely identified. The EPC code is a 13-digit number that ensures that every code is unique. The RFID tag embedded in the FASTAG is a passive tag that is energized only when the beam of the scanner hits it. So, when a vehicle comes near to the scanning device, the scanner sends RF signals and reads the information present in the tag. As the FASTAG is pre-paid with money, so it automatically deducts the corresponding toll fare. Because of the drawbacks of FASTSAG, which are previously stated, this paper provides a solution to reduce those problems by deducting the toll fares automatically using image processing. In this system, we reduce waiting time at toll plazas and make the entire process automated. The payment is proceeded by retrieving account details from the database using the vehicle's number plate details which are unique, then a notification is sent to the user's app, the amount is deducted if the user acknowledges the notification else the corresponding footage is sent to the police database. Because of this method vehicle thefts, criminal activities can also be reduced.

## 2.2 FLOW CHART:

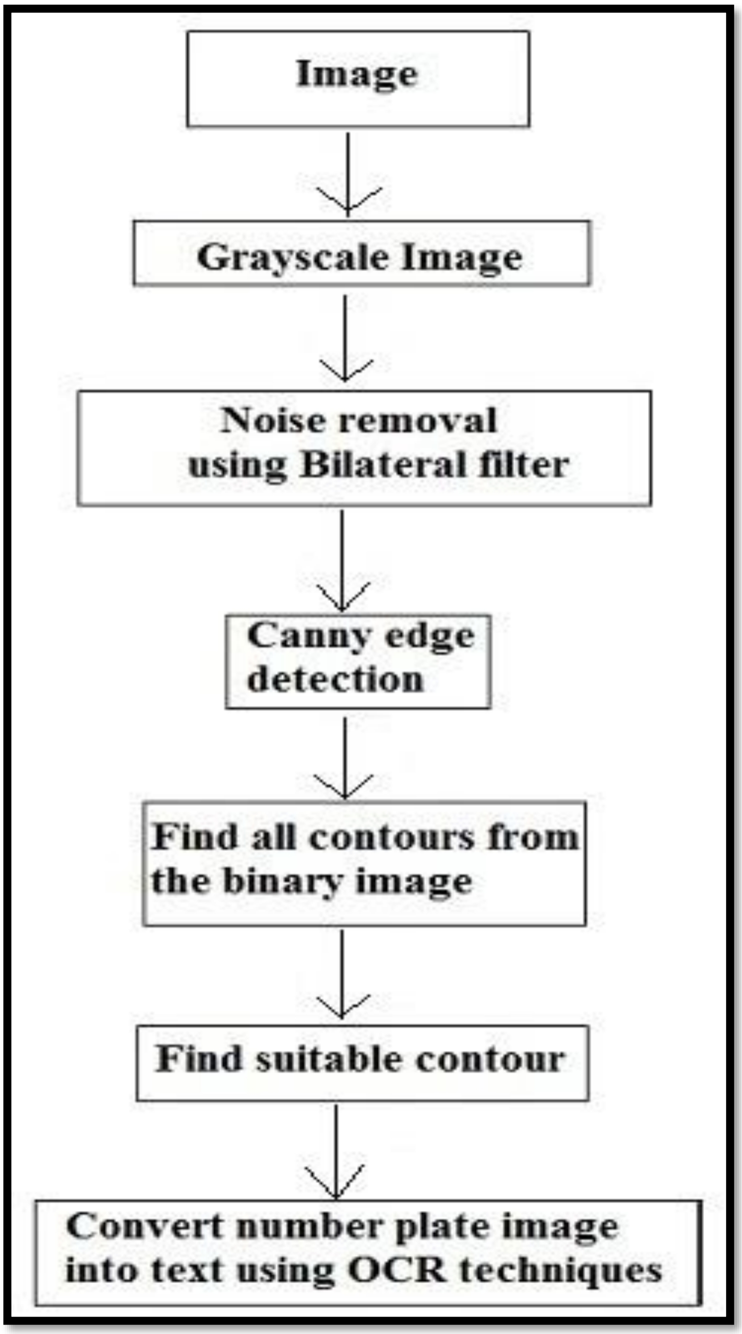



Fig 2.1 Flowchart of Image Pre-Processing

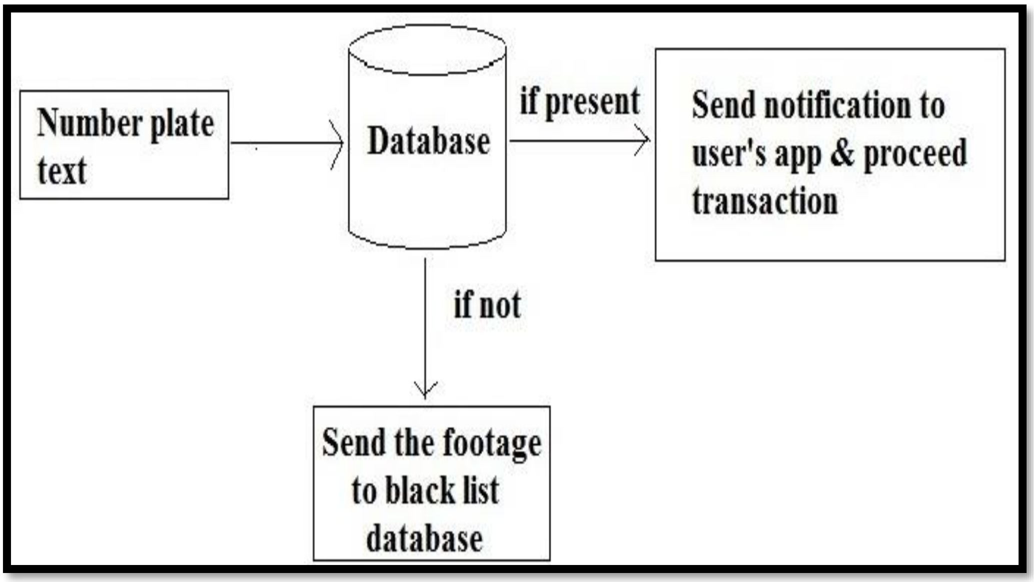



Fig 2.2 At server side

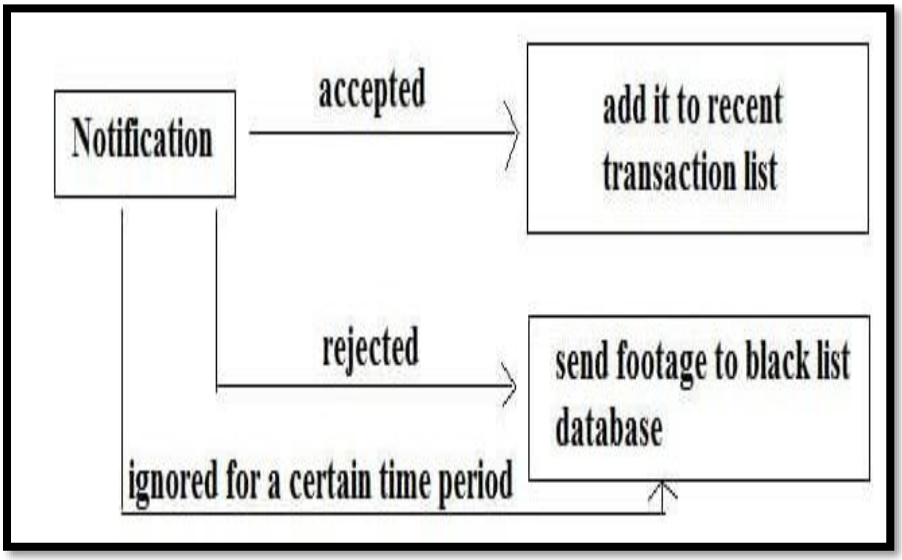



Fig 2.3 At user side

# CHAPTER-3
# IMAGE PRE-PROCESSING

## 3.1 CONVERSION OF RGB TO GRAY SCALE IMAGE:

A way to convert a color image 3D array to a grayscale 2D array is, for each pixel, take the average of the red, green, and blue pixel values to get the grayscale value. This combines the lightness or *luminance* contributed by each color band into a reasonable gray approximation. To our eyes green looks about ten times brighter than blue. Through many repetitions of carefully designed experiments, scientists have figured out how different we perceive the luminance or red, green, and blue to be. They have provided us a different set of weights for our channel averaging to get total luminance.

$$Y_{\text{linear}} = 0.2126R_{\text{linear}} + 0.7152G_{\text{linear}} + 0.0722B_{\text{linear}} \quad \text{-----(3.1)}$$

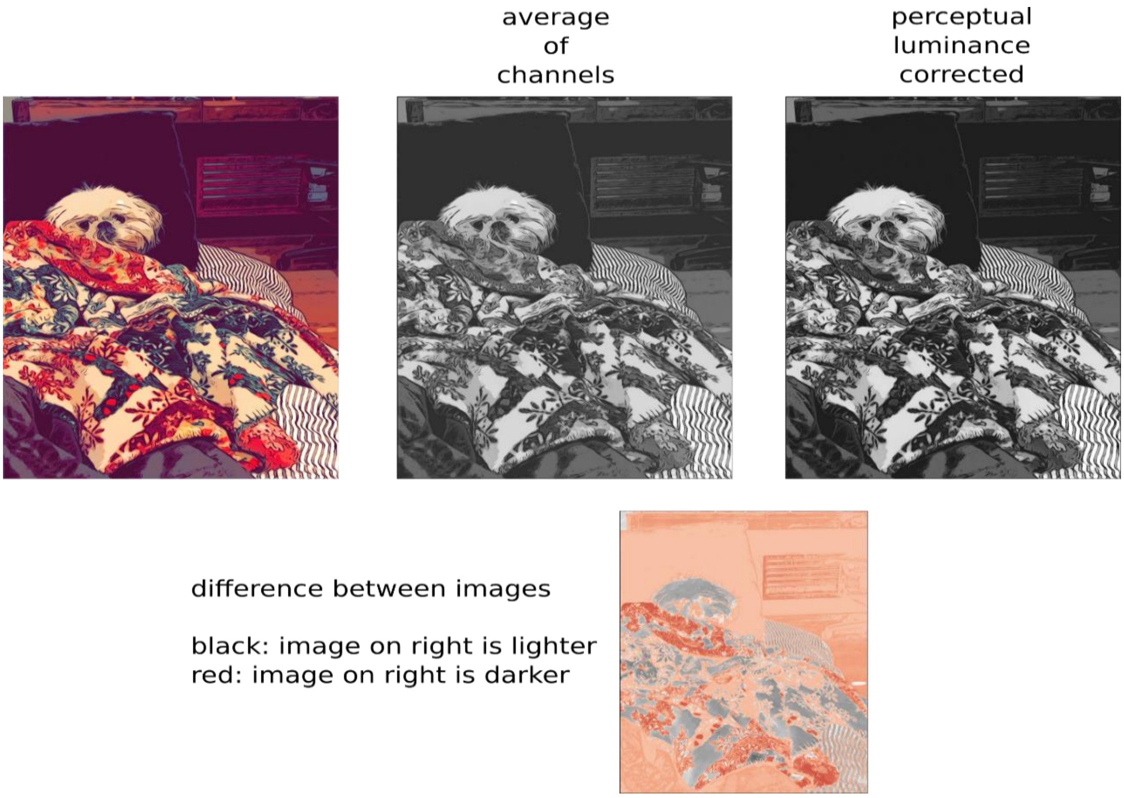



Fig 3.1 Weighted grayscale and luminance corrected grayscale image.

We are able to see small differences when luminance is low, but at high luminance levels, we are much less sensitive to them. In order to avoid wasting effort representing imperceptible differences at high luminance, the colour scale is warped, so that it concentrates more values in the lower end of the range, and spreads them out more widely in the higher end. This is called gamma compression.

To undo the effects of gamma compression before calculating the grayscale luminance, it's necessary to apply the inverse operation, gamma expansion:

$$C_{\text{linear}} = \begin{cases} \frac{C_{\text{srgb}}}{12.92}, & C_{\text{srgb}} \leq 0.04045 \\ \left(\frac{C_{\text{srgb}}+0.055}{1.055}\right)^{2.4}, & C_{\text{srgb}} > 0.04045 \end{cases}$$

**-------(3.2)**

The gamma decompression and re-compression rack up quite a large computation cost, compared to the weighted averages we were working with before. Sometimes speed is more desirable than accurate-as-possible luminance calculations.

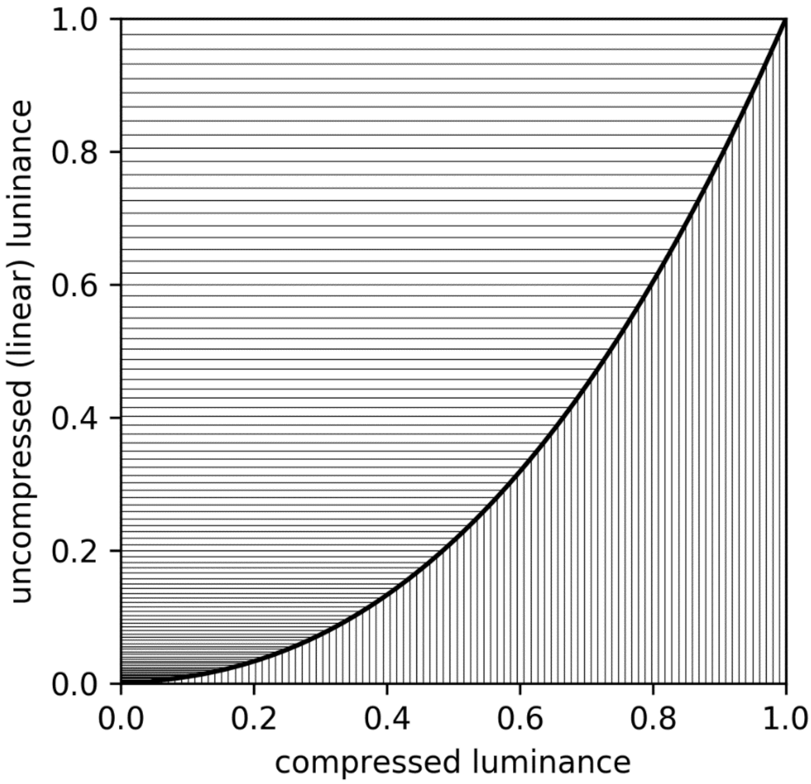



Fig 3.2 compressed vs uncompressed luminance

## 3.2 SMOOTHING FILTERS:

Image blurring is done by passing an image with the low-pass filter kernel. It is very useful for removing noise. It removes high-frequency content from the image. So, edges are blurred in this operation but there are also blurring techniques that don't blur the edges.

There are different blurring methods that can be used to blur the gray image.

### 3.2.1 AVERAGE BLURRING:

An average filter does exactly what you think it might do — takes an area of pixels surrounding a central pixel, averages all these pixels together, and replaces the central pixel with the average.

By taking the average of the region surrounding a pixel, we are **smoothing** it and replacing it **with the value of its local neighbourhood**. This allows us to reduce noise and the level of detail, simply by relying on the average.

- To accomplish our average blur, we'll actually be convolving our image with a **M x N** normalized filter where both **M** and **N** are both odd integers.
- This kernel is going to **slide** from **left-to-right** and from **top-to-bottom** for each and every pixel in our input image. The pixel at the centre of the kernel (and hence why we have to use an odd number, otherwise there woul d not be a true "centre") is then set to be the *average* of all other pixels surrounding it.

The 3 x 3 version of that kernel looks like this:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

### 3.2.2 GAUSSIAN FILTER:

Gaussian smoothing means using the **kernel** whose **values** follow a **Gaussian distribution**. The values are generated by a Gaussian function so it requires a sigma value ***σ*** for its parameter.

The equation for a Gaussian function in one direction is:

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{x^2}{2\sigma^2}}$$

**-------(3.3)**

The shape of the curve is controlled a single parameter called ***σ*** that controls the peakness of the bell curve.

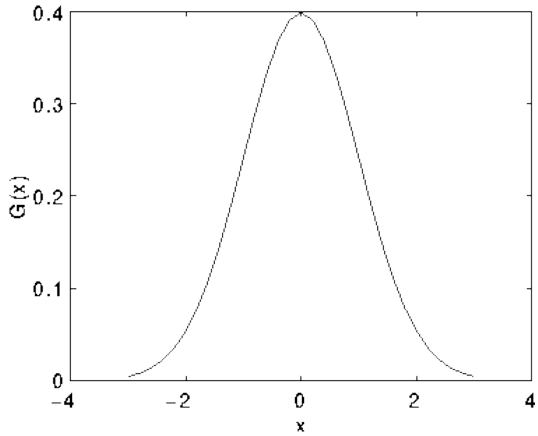



Fig 3.3 1-D Gaussian distribution with mean 0 and $\boldsymbol{\sigma}$ =1

We can extend this equation to two directions, one for the *x*-axis and the other for the *y*-axis, respectively:

$$G(x,\ y) = \frac{1}{2\pi\sigma^2}\exp^{-\frac{x^2\ +\ y^2}{2\sigma^2}}$$

**--------(3.4)**

where **x** and **y** are the respective distances to the horizontal and vertical center of the kernel and $\boldsymbol{\sigma}$ is the standard deviation of the Gaussian kernel.

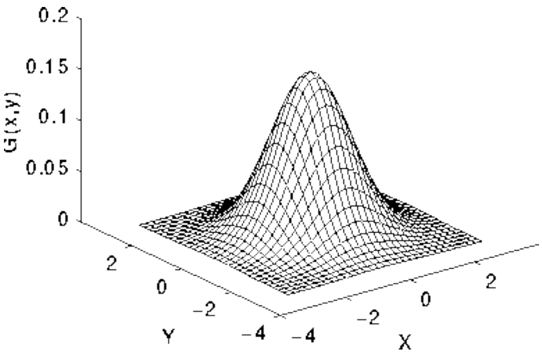



Fig 3.4  The shape of a Gaussian filter with mean (0,0)

The values of the kernel go higher near the center and go smaller near the corner.

**Note:** A bigger $\sigma$ creates a kernel that blurs more. The degree of smoothing is determined by the standard deviation of the Gaussian.

### 3.2.3 BILATERAL FILTER:

A **bilateral filter** is a non-linear, edge-preserving, and noise reducing smoothing filter for images. It replaces the intensity of each pixel with a weighted average of intensity values from nearby pixels. This weight can be based on a Gaussian distribution. Crucially, the weights depend not only on Euclidean distance of pixels, but also on the radiometric differences (e.g., range differences, such as color intensity, depth distance, etc.). This preserves sharp edges.

$$I^{\text{filtered}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|),$$

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

**----------------(3.5)**

$I^{\text{filtered}}$ is the filtered image;

$I$ is the original input image to be filtered;

$x$ are the coordinates of the current pixel to be filtered;

$\Omega$ is the window centered in $x$, so $x_i \in \Omega$ is another pixel;

$f_r$ is the range kernel for smoothing differences in intensities (this function can be a Gaussian function);

$g_s$ is the spatial (or domain) kernel for smoothing differences in coordinates (this function can be a Gaussian function).

The basic idea underlying bilateral filtering is to do in the range of an image what traditional filters do in its domain. Two pixels can be *close* to one another, that is, occupy nearby spatial location, or they can be *similar* to one another, that is, have nearby values, possibly in a perceptually meaningful fashion. The spatial distribution of image intensities plays no role in range filtering taken by itself. Combining intensities from the entire image, however, makes little sense, since the distribution of image values far away from x ought not to affect the final value at x. In addition, one can show that range filtering without domain filtering merely changes the color map of an image, and is therefore of little use. The appropriate solution is to combine domain and range filtering, thereby enforcing both geometric and photometric locality.

Combined domain and range filtering will be denoted as *bilateral filtering*. It replaces the pixel value at x with an average of similar and nearby pixel values. In smooth regions, pixel values in a small neighborhood are similar to each other, and the bilateral filter acts essentially as a standard domain filter, averaging away the small, weakly correlated differences between pixel values caused by noise. Consider now a sharp boundary between a dark and a bright region, as in figure ‘a’

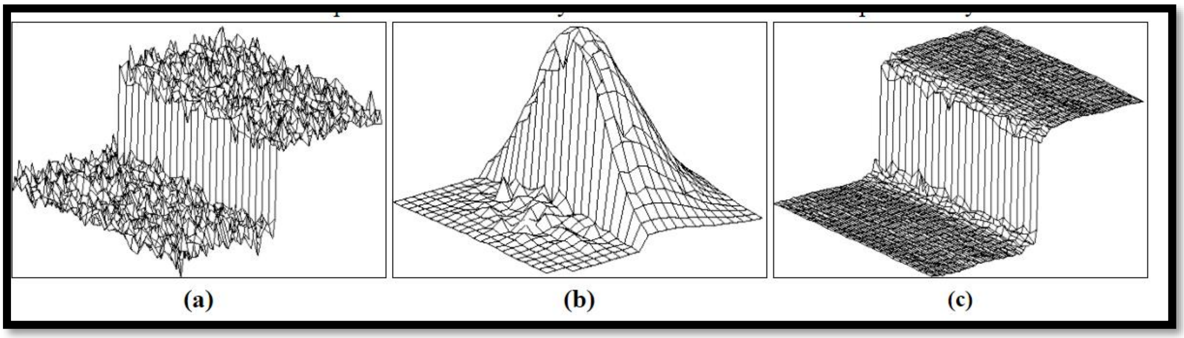



Fig 3.5 Cases in Bilateral filter

When the bilateral filter is centered, say, on a pixel on the bright side of the boundary, the similarity function *s* assumes values close to one for pixels on the same side, and values close to zero for pixels on the dark side. The similarity function is shown in figure (b) for a 23x23 filter support centered two pixels to the right of the step in figure (a). The normalization term *k(x)* ensures that the weights for all the pixels add up to one. As a result, the filter replaces the bright pixel at the center by an average of the bright pixels in its vicinity, and essentially ignores the dark pixels. Conversely, when the filter is centered on a dark pixel, the bright pixels are ignored instead. Thus, as shown in figure (c), good filtering behavior is achieved at the boundaries, thanks to the domain component of the filter, and crisp edges are preserved at the same time, thanks to the range component.

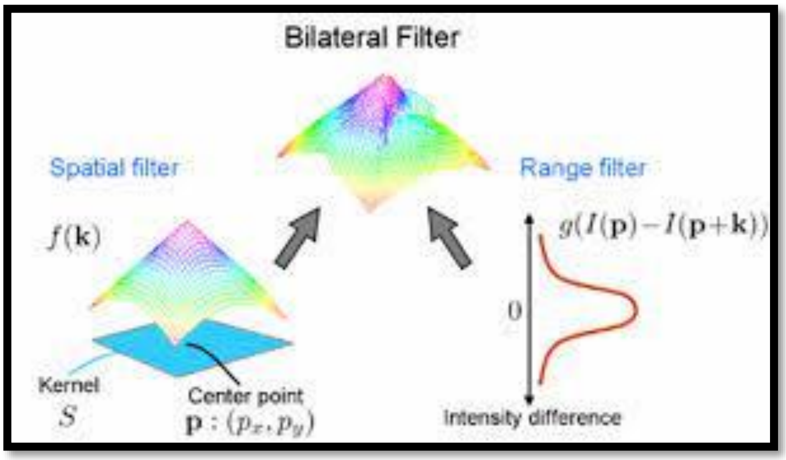



Fig 3.6 Spatial and range components of a Bilateral filter

Fig 3.7 Bilateral filtering to an image

- WHY BILATERAL FILTER IS USED?

We prefer the bilateral filter to blur the image because it actually preserves all strength, it removes noise quite well and strengthens the edges in the image when we deal with a single-layered image.

## 3.3 GRAY TO BINARY IMAGE:

A binary image can be obtained by either applying thresholding method or edge detection.

### 3.3.1 EDGE DETECTION:

**Edge Detection** is a method of segmenting an image into regions of discontinuity. It is a widely used technique in digital image processing:

- pattern recognition
- Image morphology
- feature extraction

Edge detection allows users to observe the features of an image for a significant change in the Gray level. This texture indicating the end of one region in the image and the beginning of another. It reduces the amount of data in an image and preserves the structural properties of an image.

There are three main types of Edge Detection**.**

1. **Sobel edge detection**
2. **Laplacian edge detection**
3. **Canny edge detection**

### 3.3.1.1 SOBEL EDGE DETECTION:

Sobel Edge Detection (first method) is a way to avoid the gradient calculated about an interpolated point between the pixels which uses 3 x 3 neighborhoods for the calculations of the gradient. It finds vertical or horizontal edges.

### 3.3.1.2 LAPLACIAN OF GAUSSIAN (LoG):

It is a gaussian-based operator which uses the Laplacian to take the second derivative of an image. This really works well when the transition of the grey level seems to be abrupt. It works on the zero-crossing method i.e when the second-order derivative crosses zero, then that particular location corresponds to a maximum level. It is called an edge location. Here the Gaussian operator reduces the noise and the Laplacian operator detects sharp edges. The Gaussian function is defined by the formula:

$$G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \left( \frac{x^2 + y^2}{2\sigma^2} \right) \quad \text{---------(3.6)}$$

The LoG operator is computed from:

$$\text{LoG} = \frac{\partial^2}{\partial x^2} G(x,y) + \frac{\partial^2}{\partial y^2} G(x,y) = \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \exp\left( - \frac{x^2 + y^2}{2\sigma^2} \right) \quad \text{---------(3.7)}$$

### 3.3.1.3 CANNY EDGE DETECTION:

The Canny edge detector is an edge_detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

It is composed of 5 steps:

1.Noise reduction

2.Gradient calculation

3.Non-maximum suppression

4.Double threshold

5.Edge Tracking by Hysteresis

1.NOISE REDUCTION:

One way to get rid of the noise on the image, is by applying Gaussian blur to smooth it. To do so, image convolution technique is applied with a Gaussian Kernel (3x3, 5x5, 7x7 etc…). The kernel size depends on the expected blurring effect. Basically, the smallest the kernel, the less visible is the blur. In our example, we will use a 5 by 5 Gaussian kernel.

**The equation for a Gaussian filter kernel of size (2*k*+1) × (2*k*+1) is given by:**

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \leq i, j \leq (2k+1)$$

**----------(3.8)**

Fig. 3.8 Original image

Fig. 3.9 Gaussian filter applied

2.Gradient Calculation:

The Gradient calculation step detects the edge intensity and direction by calculating the gradient of the image using edge detection operators. When the image is smoothed, the derivatives *Ix* and *Iy* with respect to *x* and *y* are calculated. It can be implemented by convolving *I* with Sobel kernels *Kx* and *Ky*, respectively:

$$K_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, K_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}.$$

Then, the magnitude *G* and the slope $\theta$ of the gradient are calculated as follow:

$$|G| = \sqrt{I_x^2 + I_y^2},$$

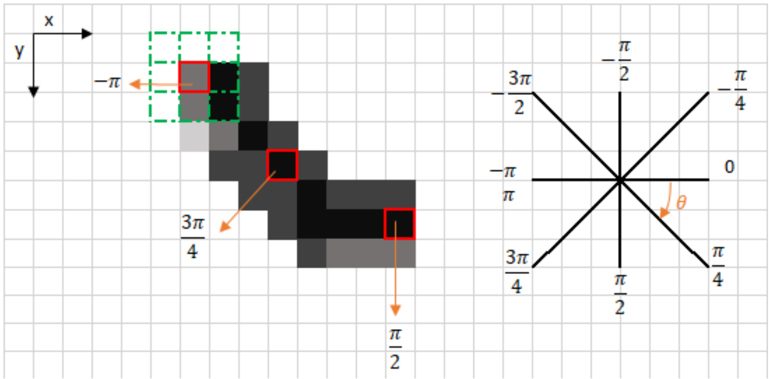
$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

Fig 3.10 Gradient of the image

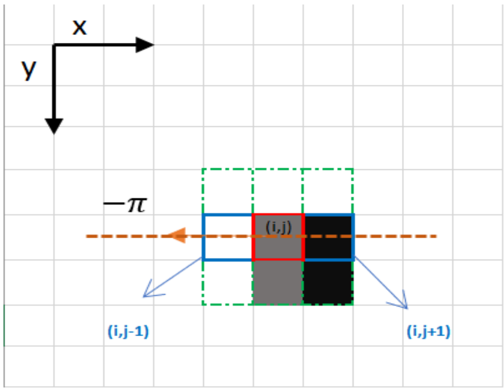
The result is almost the expected one, but we can see that some of the edges are thick and others are thin. Non-Max Suppression step will help us mitigate the thick ones.

3.NON-MAXIMUM SUPPRESSION:

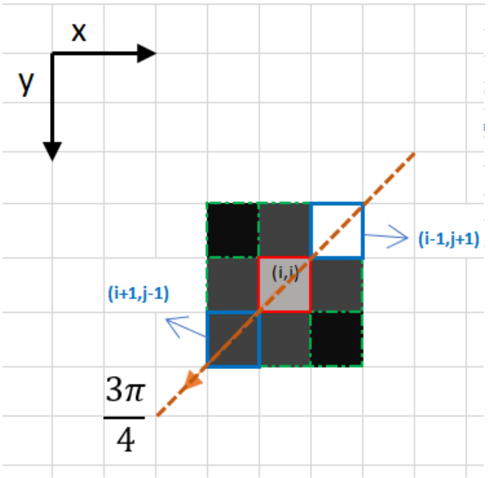
The principle is simple: the algorithm goes through all the points on the gradient intensity matrix and finds the pixels with the maximum value in the edge directions.




The upper left corner red box present on the above image, represents an intensity pixel of the Gradient Intensity matrix being processed. The corresponding edge direction is represented by the orange arrow with an angle of -pi radians (+/-180 degrees).




The edge direction is the orange dotted line (horizontal from left to right). The purpose of the algorithm is to check if the pixels on the same direction are more or less intense than the ones being processed. In the example above, the pixel *(i, j)* is being processed, and the pixels on the same direction are highlighted in blue *(i, j-1)* and *(i, j+1)*. If one those two pixels are more intense than the one being processed, then only the more intense one is kept. Pixel *(i, j-1)* seems to be more intense, because it is white (value of 255). Hence, the intensity value of the current pixel *(i, j)* is set to 0. If there are no pixels in the edge direction having more intense values, then the value of the current pixel is kept.




In this case the direction is the orange dotted diagonal line. Therefore, the most intense pixel in this direction is the pixel *(i-1, j+1)*.

Fig 3.11 Non-maximum suppressed image

4. DOUBLE THRESHOLD:

The double threshold step aims at identifying 3 kinds of pixels: strong, weak, and non-relevant: Strong pixels are pixels that have an intensity so high that we are sure they contribute to the final edge. Weak pixels are pixels that have an intensity value that is not enough to be considered as strong ones, but yet not small enough to be considered as non-relevant for the edge detection. Other pixels are considered as non-relevant for the edge. High threshold is used to identify the strong pixels (intensity higher than the high threshold). Low threshold is used to identify the non-relevant pixels (intensity lower than the low threshold). All pixels having intensity between both thresholds are flagged as weak and the Hysteresis mechanism (next step) will help us identify the ones that could be considered as strong and the ones that are considered as non-relevant.

Fig 3.12 Double thresholded image

5.EDGE TRACKING BY HYSTERESIS:

Based on the threshold results, the hysteresis consists of transforming weak pixels into strong ones, if and only if at least one of the pixels around the one being processed is a strong one, as described below:

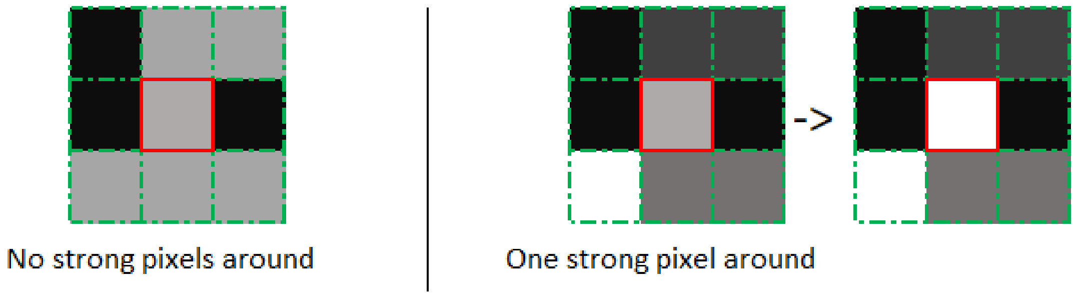



Fig. Edge tracking for example pixels

Fig 3.13 Canny edge detection of the image

## ❖ WHY CANNY EDGE DETECTION IS PREFERRED?

The Canny edge detection is highly preferred to extract the edges from the blurred image because of its optimal result, well-defined edges, and accurate detection.

### 3.3.2 THRESHOLDING:

The simplest thresholding methods replace each pixel in an image with a black pixel if the image intensity I(i,j) is less than some fixed constant T (that is I(i,j)<T), or a white pixel if the image intensity is greater than that constant. In the example image on the right, this results in the dark tree becoming completely black, and the white snow becoming completely white.

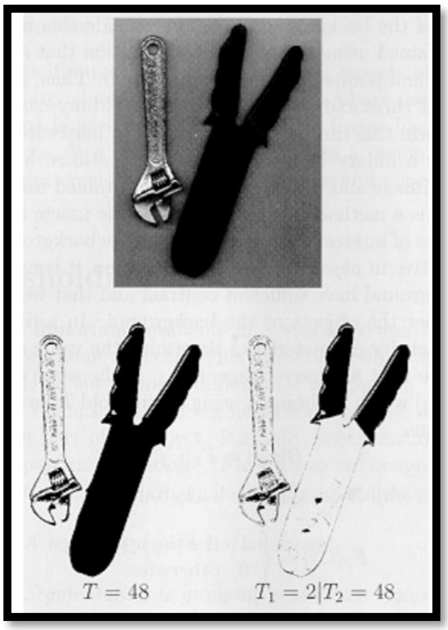



Fig 3.14 Thresholded image with two thresholds

1.HYSTERESIS THRESHOLDING:

If there is no clear valley in the histogram of an image, it means that there are several background pixels that have similar gray level value with object pixels and vice versa. Pixels above the high threshold are classified as object and below the low threshold as background. Hysteresis thresholding (i.e., two thresholds, one at each side of the valley) can be used in this case. Pixels between the low and high thresholds are classified as object only if they are adjacent to other object pixels.

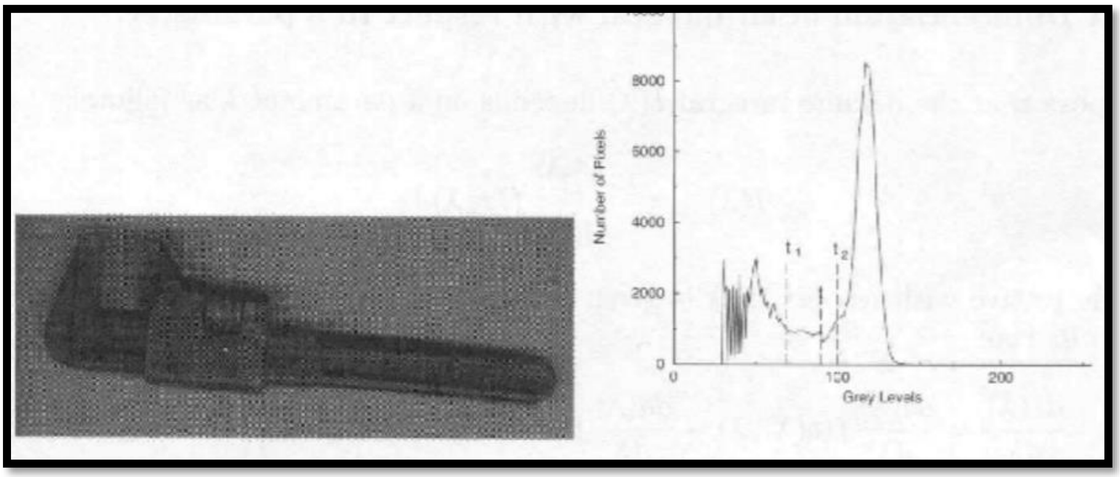



Fig 3.15 Histogram of the image

2.OPTIMAL THRESHOLDING

We can minimize the number of misclassified pixels if we have some prior knowledge about the distributions of the gray level values that make up the object and the background. Assume that the distribution of gray-level values in each region follows a Gaussian distribution.

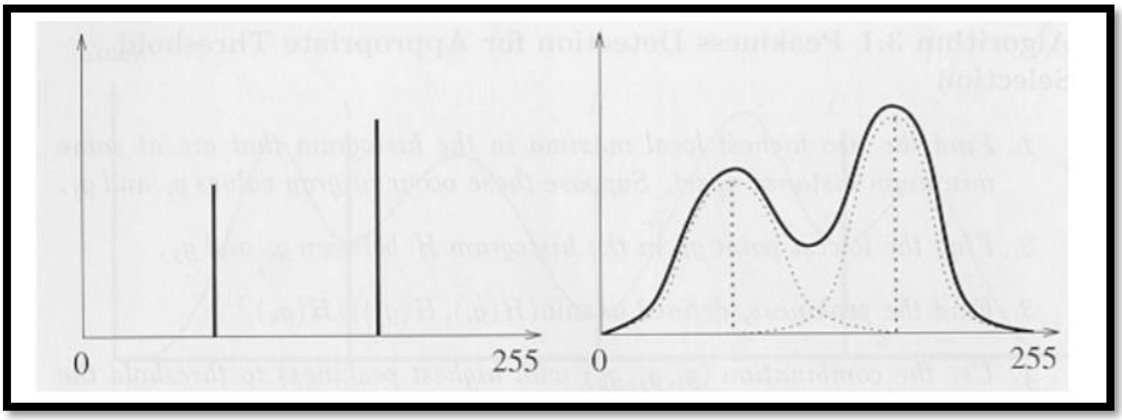



Fig 3.16 Gaussian distribution of the image

The probability of a pixel value is then given by the following mixture:

$$P(z) = p(z/background)\ P(background) + p(z/object)\ P(object)$$

$$\text{or } P(z) = P_b \frac{1}{\sqrt{2\pi\sigma_b}} e^{-\frac{(z-\mu_b)^2}{2\sigma_b^2}} + P_o \frac{1}{\sqrt{2\pi\sigma_o}} e^{-\frac{(z-\mu_o)^2}{2\sigma_o^2}}$$

$$\text{or } P(z) = P_b p_b(z) + P_o p_o(z)$$

$p_b(z)$, $p_o(z)$, prob. distributions of background, object pixels

$\mu_b$, $\mu_o$: the means of the distributions

$\sigma_b$, $\sigma_o$: the standard deviations of the distributions

$P_b$, $P_o$: the a-priori probabilities of background, object pixels

**----(3.9)**

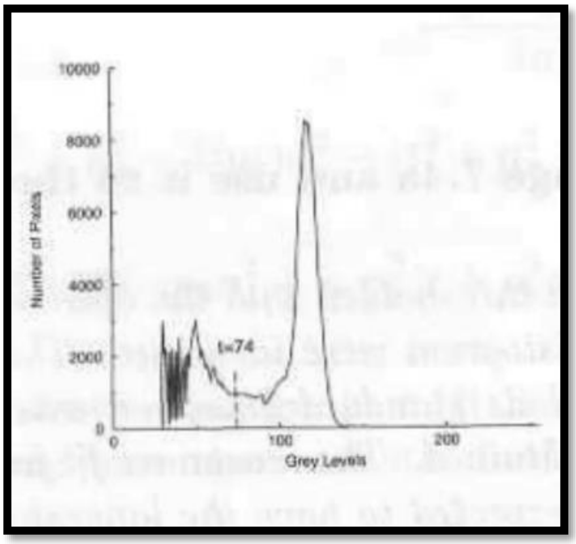



Fig 3.17 Histogram using Optimal threshold.

3.OTSU'S METHOD:

A measure of region homogeneity is variance (i.e., regions with high homogeneity will have low variance). Otsu's method selects the threshold by minimizing the within-class variance of the two groups of pixels separated by the thresholding operator. It does not depend on modeling the probability density functions, however, it assumes a bimodal distribution of gray-level values (i.e., if the image approximately fits this constraint, it will do a good job).

- Since the total variance $\sigma$ does not depend on $T$, the $T$ minimizing $\sigma_W^2$ will be the $T$ maximizing $\sigma_B^2$.

- Let's consider maximizing $\sigma_B^2$, we can rewrite $\sigma_B^2$ as follows:

$$\sigma_B^2 = \frac{[\mu(T) - \mu q_B(T)]^2}{q_B(T) q_o(T)}$$

where $\mu(T) = \sum_{i=1}^{T} iP(i)$

- Start from the beginning of the histogram and test each gray-level value for the possibility of being the threshold $T$ that maximizes $\sigma_B^2$

**-----(3.10)**

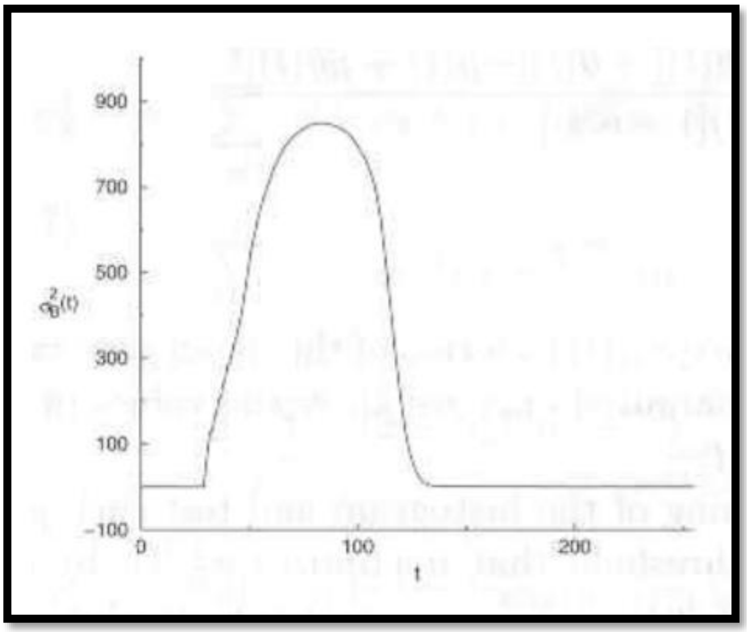



Fig 3.18 variance vs threshold for otsu's model

### 3.3.3 EDGE DETECTION VS THRESHOLDING :

Otsu thresholding method assumes that the histogram of the image is bimodal (i.e., two classes). It breaks down when the two classes are very unequal (i.e., the classes have very different sizes). In this case, variance may have two maxima. The correct maximum is not necessary the global one. The selected threshold should correspond to a valley of the histogram. The method does not work well with variable illumination.

## 3.4 CONTOURS:

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. Contours can be identified using Suzuki & Be’s algorithm.

Two border following algorithms are used for the topological analysis of digitized binary images. The first one determines the surroundness relations among the borders of a binary image. Since the outer borders and the hole borders have a one-to-one correspondence to the connected components of l-pixels and to the holes, respectively, the proposed algorithm yields a representation of a binary image, from which one can extract some sort of features without reconstructing the image. The second algorithm, which is a modified version of the first, follows only the outermost borders (i.e., the outer borders which are not surrounded by holes). These algorithms can be effectively used in component counting, shrinking, and topological structural analysis of binary images, when a sequential digital computer is used.

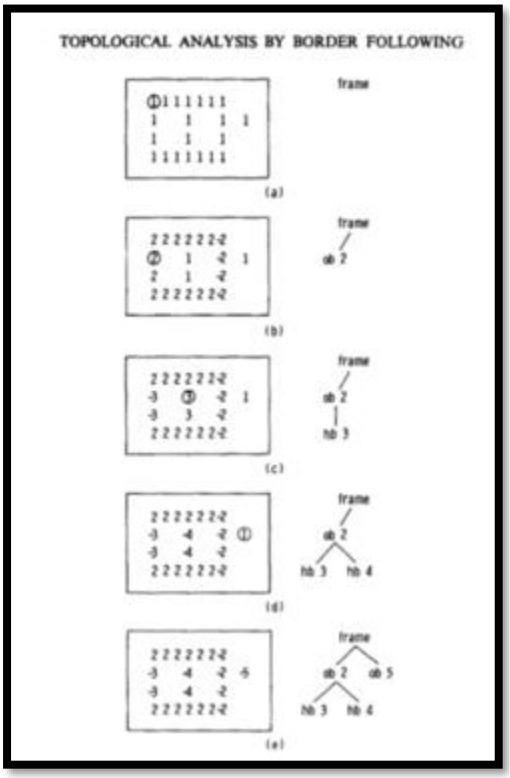



Fig 3.19 Topological analysis by border

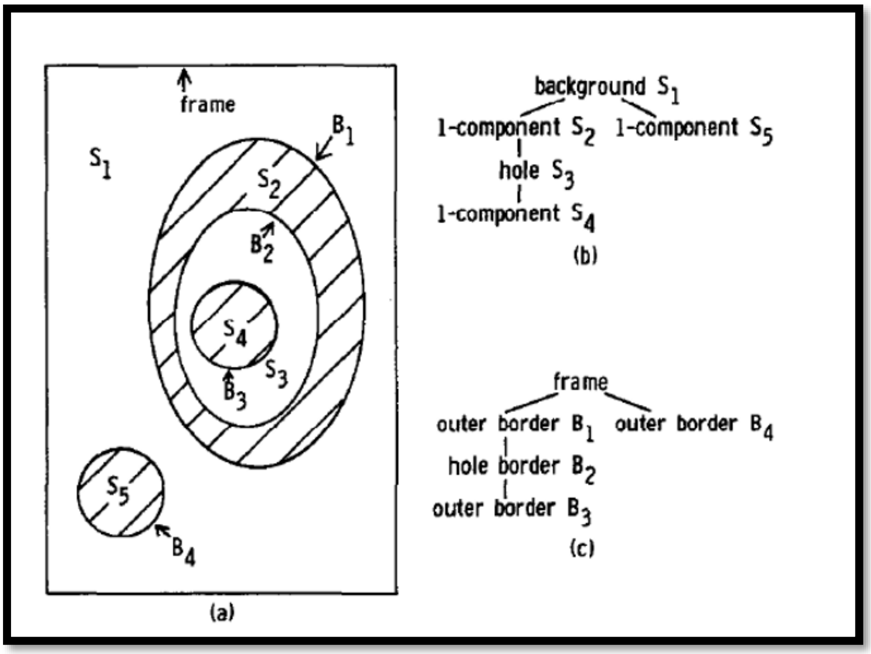



Fig 3.20 Surroundness around connected components.

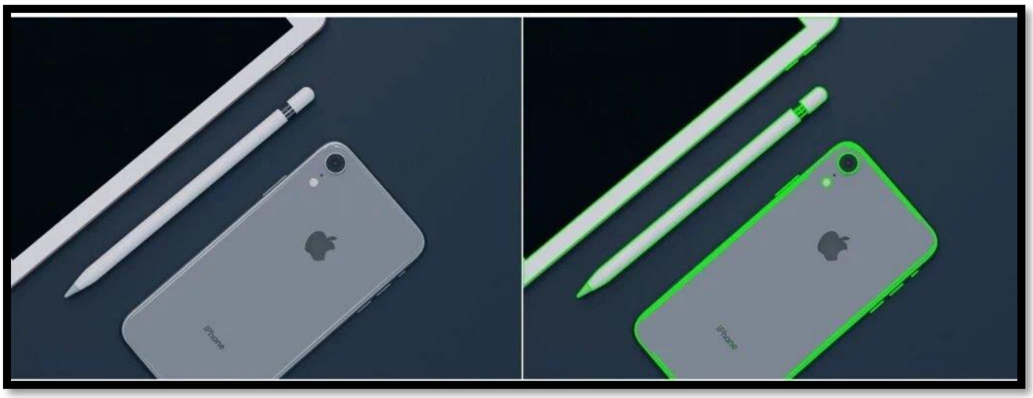


Fig 3.21 Contours of the image

# CHAPTER 4
# SOFTWARE DESCRIPTION

## 4.1 SOFTWARE TOOLS USED:

1. PYTHON (for image processing)
2. HTML (front-end)
3. CSS (front-end)
4. REACT (front-end)
5. PHP (Back-end)
6. MY SQL(Back-end)

## 4.2 OVERVIEW OF SOFTWARES USED:

### 4.2.1 HTML:

Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

HTML elements are delineated by tags, written using angle brackets. Tags such as <img> and <input/> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

### 4.2.2 CSS:

Stands for "Cascading Style Sheet". Cascading style sheets are used to format the layout of Web pages. They can be used to define text styles, table sizes, and other aspects of Web pages that previously could only be defined in a page's HTML.

CSS helps Web developers create a uniform look across several pages of a Web site. Instead of defining the style of each table and each block of text within a page's HTML, commonly used styles need to be defined only once in a CSS document. Once the style is defined in cascading style sheet, it can be used by any page that references the CSS file. Plus, CSS makes it easy to change styles across several pages at once. For example, a Web developer may want to increase the default text size from 10pt to 12pt for fifty pages of a Web site. If the pages all reference the same style sheet, the text size only needs to be changed on the style sheet and all the pages will show the larger text.

While CSS is great for creating text styles, it is helpful for formatting other aspects of Web page layout as well. For example, CSS can be used to define the cell padding of table cells, the style, thickness, and color of a table's border, and the padding around images or other objects. CSS gives Web developers more exact control over how Web pages will look than HTML does. This is why most Web pages today incorporate cascading style sheets

### 4.2.3 REACT:

ReactJS is JavaScript library used for building reusable UI components. React is a library for building composable user interfaces. It encourages the creation of reusable UI components, which present data that changes over time. Lots of people use React as the V in MVC. React abstracts away the DOM from you, offering a simpler programming model and better performance. React can also render on the server using Node, and it can power native apps using React Native. React implements one-way reactive data flow, which reduces the boilerplate and is easier to reason about than traditional data binding.

JSX is an optional syntax extension to JavaScript that makes writing your own components much easier. It accepts HTML quoting and makes a subcomponent rendering easier. In fact, it is a set of shortcuts for writing React. Create Element with a few rules to make your source cleaner and simpler.

The render method returns a *description* of what you want to see on the screen. React takes the description and displays the result. In particular, render returns a **React element**, which is a lightweight description of what to render. Most React developers use a special syntax called “JSX” which makes these structures easier to write.

JSX comes with the full power of JavaScript. You can put *any* JavaScript expressions within braces inside JSX. Each React element is a JavaScript object that you can store in a variable or pass around in your program.

FEATURES:

1. It facilitates the overall process of writing components
2. It boosts productivity and facilitates further maintenance
3. It ensures faster rendering
4. It guarantees stable code
5. It is SEO friendly
6. It comes with a helpful developer toolset
7. There is react native for mobile app development

Considering React js pros it can be easily summed up in three words: non-risky, responsive and advanced. The main idea behind this particular library is: “to build large-scale applications with data that changes repeatedly over time” and it tackles the challenge well. It provides developers with the capability of working with a virtual browser (DOM) that is much faster and user-friendly, than the real one. Apart from that, it offers the easier creation of interactive UIs, JSX support, component-based structure and much more. The combination of the above-mentioned factors makes it a reasonable choice for projects.

MAIN FUNCTIONS:

1. ReactDOM.render(element, container [callback]):

   Render a React element into the DOM in the supplied container and return a reference to the component (or returns null for stateless components).
   It currently returns a reference to the root React Component instance. However, using this return value is legacy and should be avoided because future versions of React may render components

asynchronously in some cases. If you need a reference to the root React Component instance, the preferred solution is to attach a callback ref to the root element.

2. Render():

A component with a render prop takes a function that returns a React element and calls it instead of implementing its own render logic. Components are the primary unit of code reuse in React, but it's not always obvious how to share the state or behavior that one component encapsulates to other components that need that same state.

**4.2.4 PYTHON** (for image processing)**:**

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as NUMPY which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in NumPy can be combined with OpenCV.

MAIN FUNCTIONS:

1. cv2.cvtcolor:

   cv2.cvtColor() method is used to convert an image from one color space to another. There are more than 150 color-space conversion methods available in OpenCV.
   Rgb to Gray color is one of them.

2. cv2.canny:

   The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images.

### 4.2.5 FLASK:

Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application.

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions.

### 4.2.6 PHP:

PHP (recursive acronym for PHP: Hypertext Preprocessor) is a widely-used open-source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

What distinguishes PHP from something like client-side JavaScript is that the code is executed on the server, generating HTML which is then sent to the client. The client would receive the results of running that script, but would not know what the underlying code was. You can even configure your web server to process all your HTML files with PHP, and then there's really no way that users can tell what you have up your sleeve.

The best things in using PHP are that it is extremely simple for a newcomer, but offers many advanced features for a professional programmer. Don't be afraid reading the long list of PHP's features. You can jump in, in a short time, and start writing simple scripts in a few hours.

**4.2.7 MYSQL:**

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons

1. MySQL is released under an open-source license. So you have nothing to pay to use it.
2. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySQL uses a standard form of the well-known SQL data language.
4. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
5. MySQL works very quickly and works well even with large data sets.
6. MySQL is very friendly to PHP, the most appreciated language for web development.
7. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
8. MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

## 4.3 WEB-LAYOUTS:

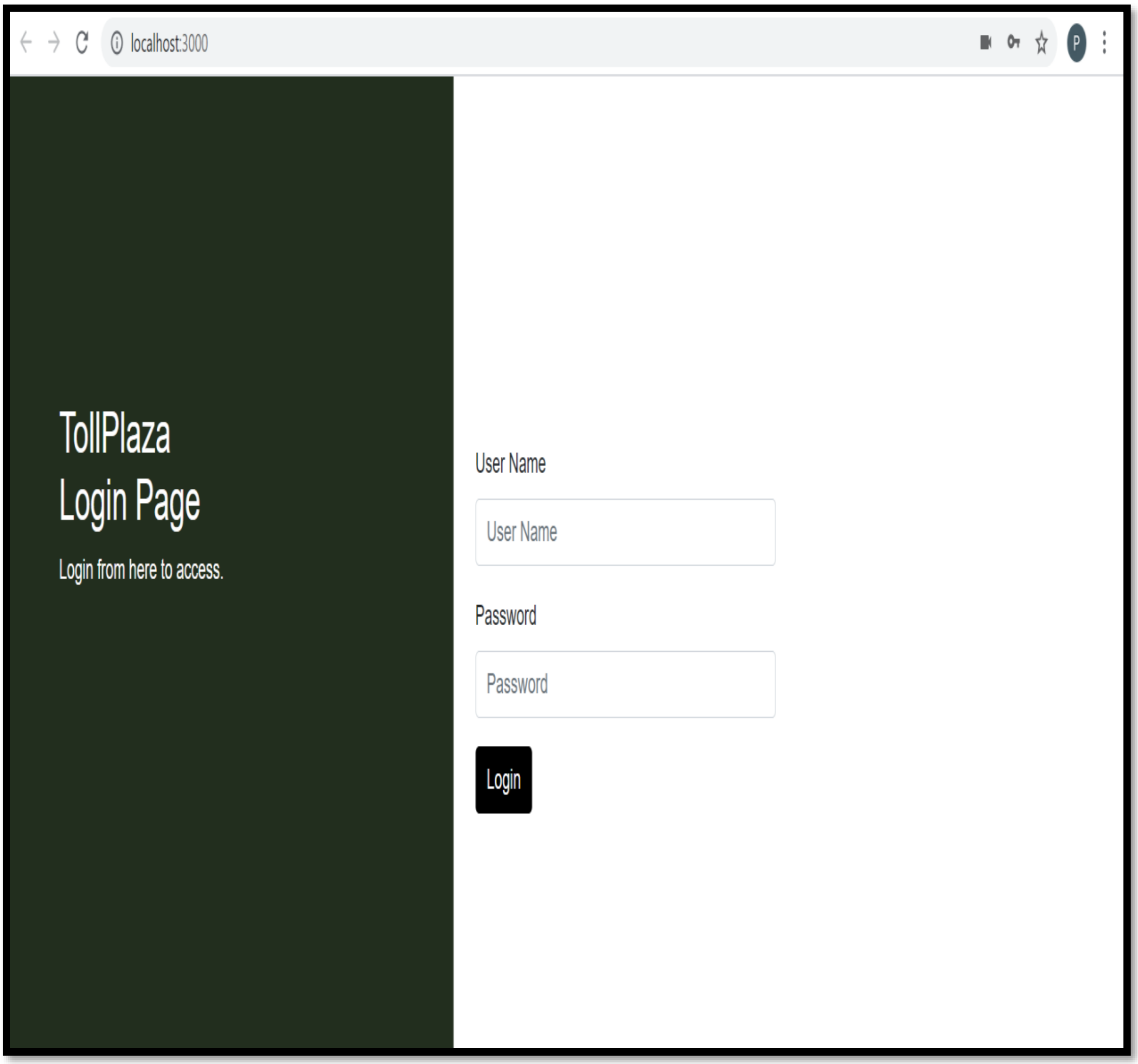



Fig 4.1 Worker Login Page

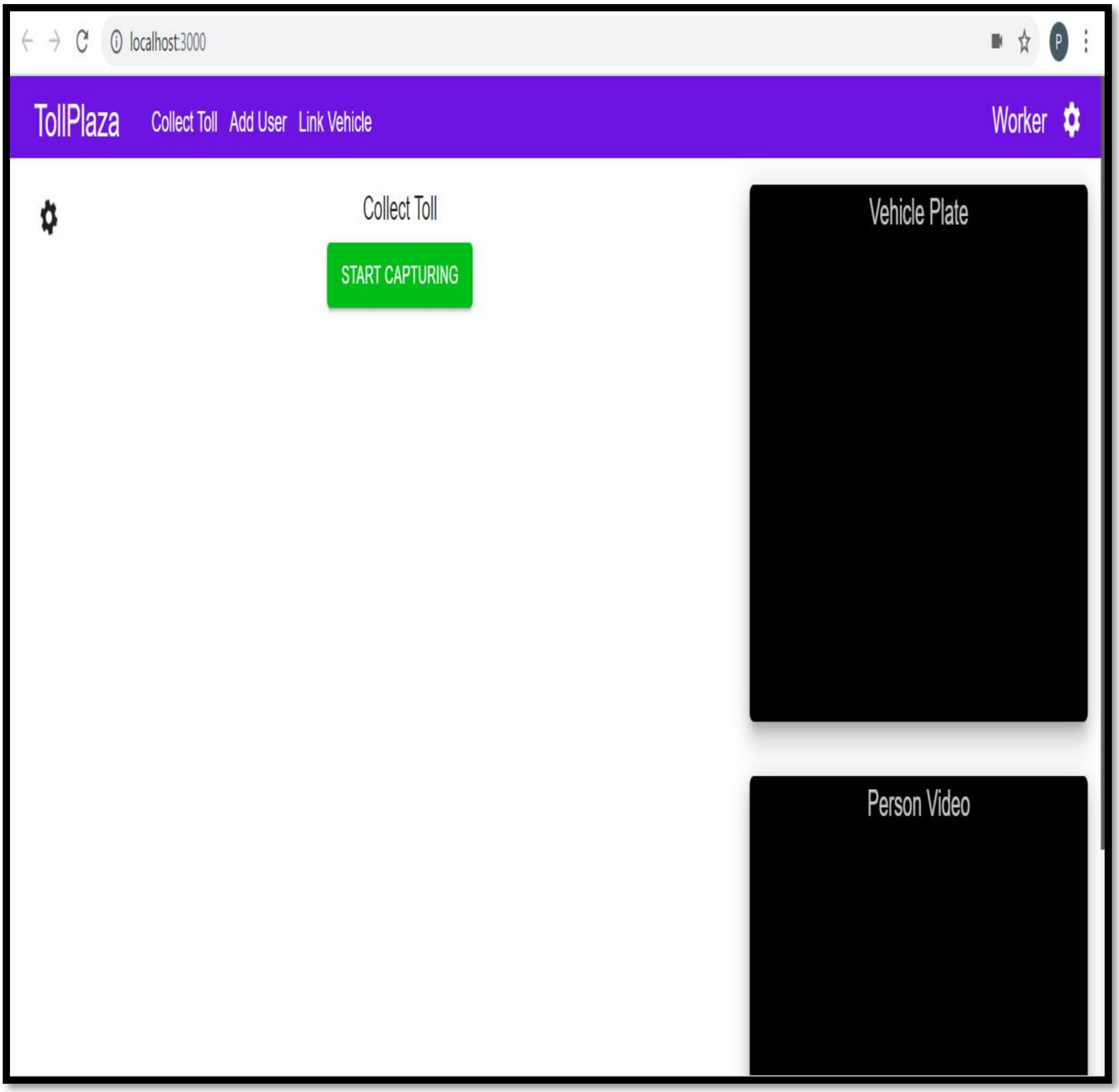



Fig 4.2 Toll plaza Index page

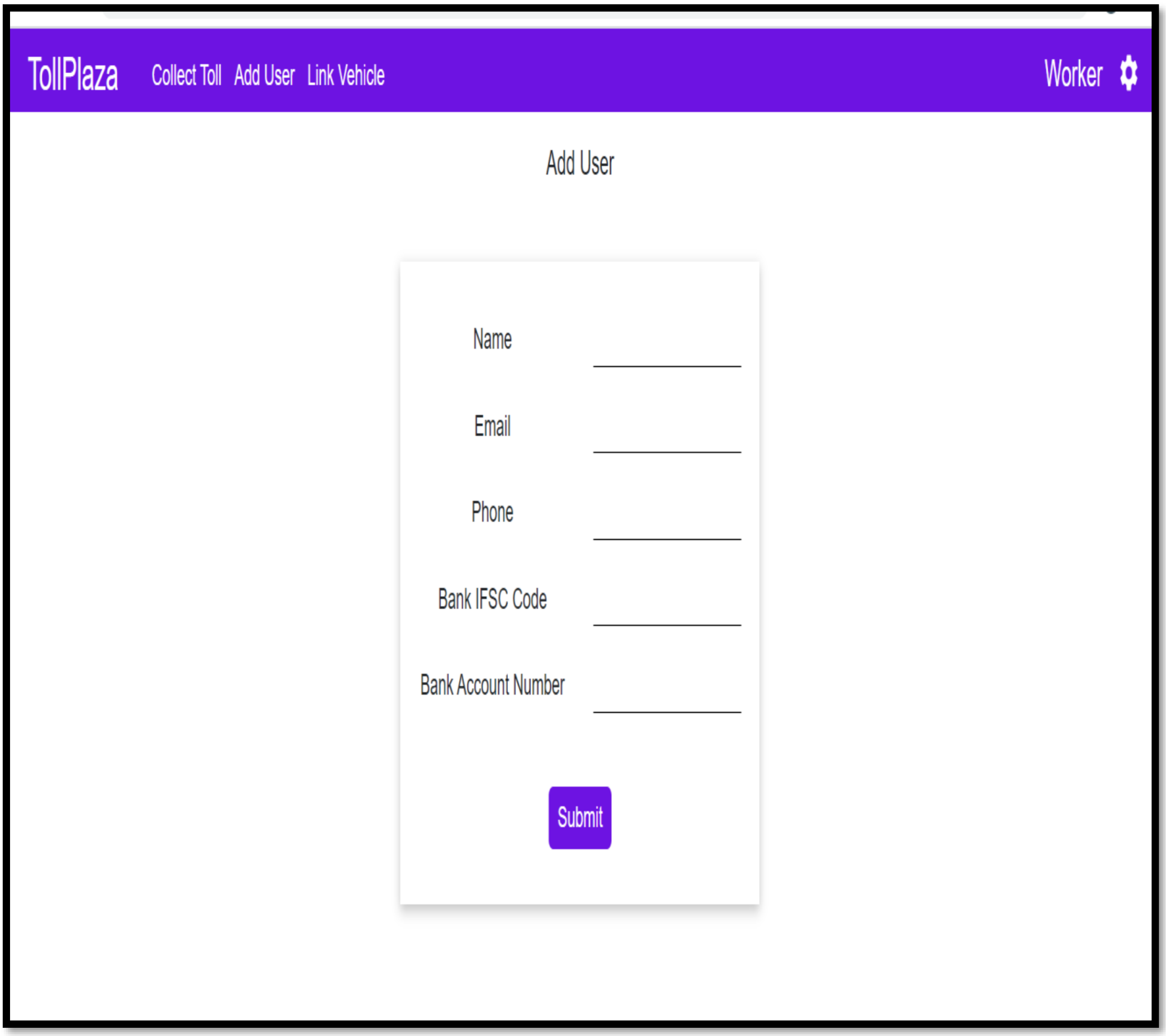



Fig 4.3 Add-User Page

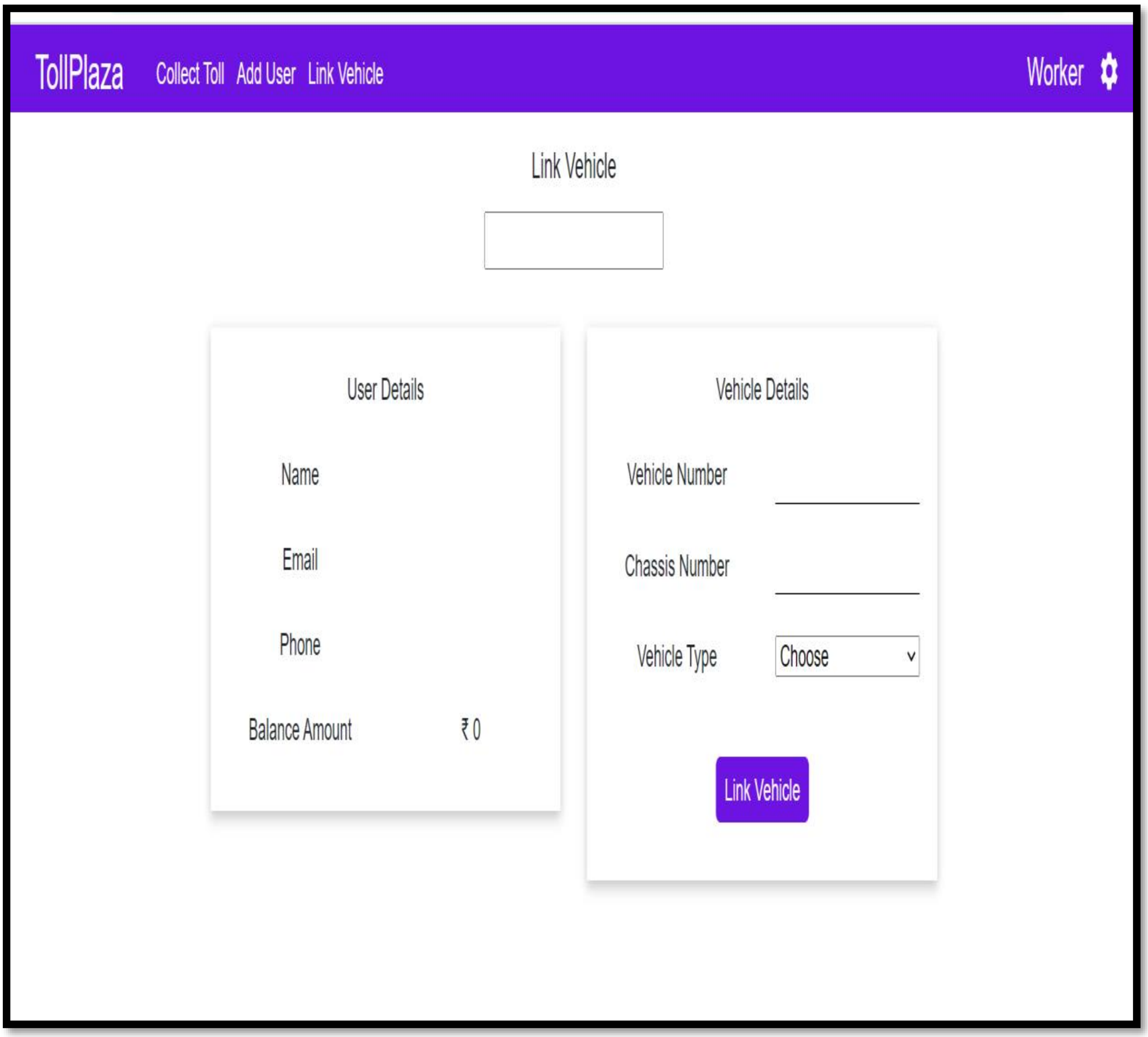



Fig 4.4 Linking-Vehicle Layout

## 4.4 APP LAYOUTS:

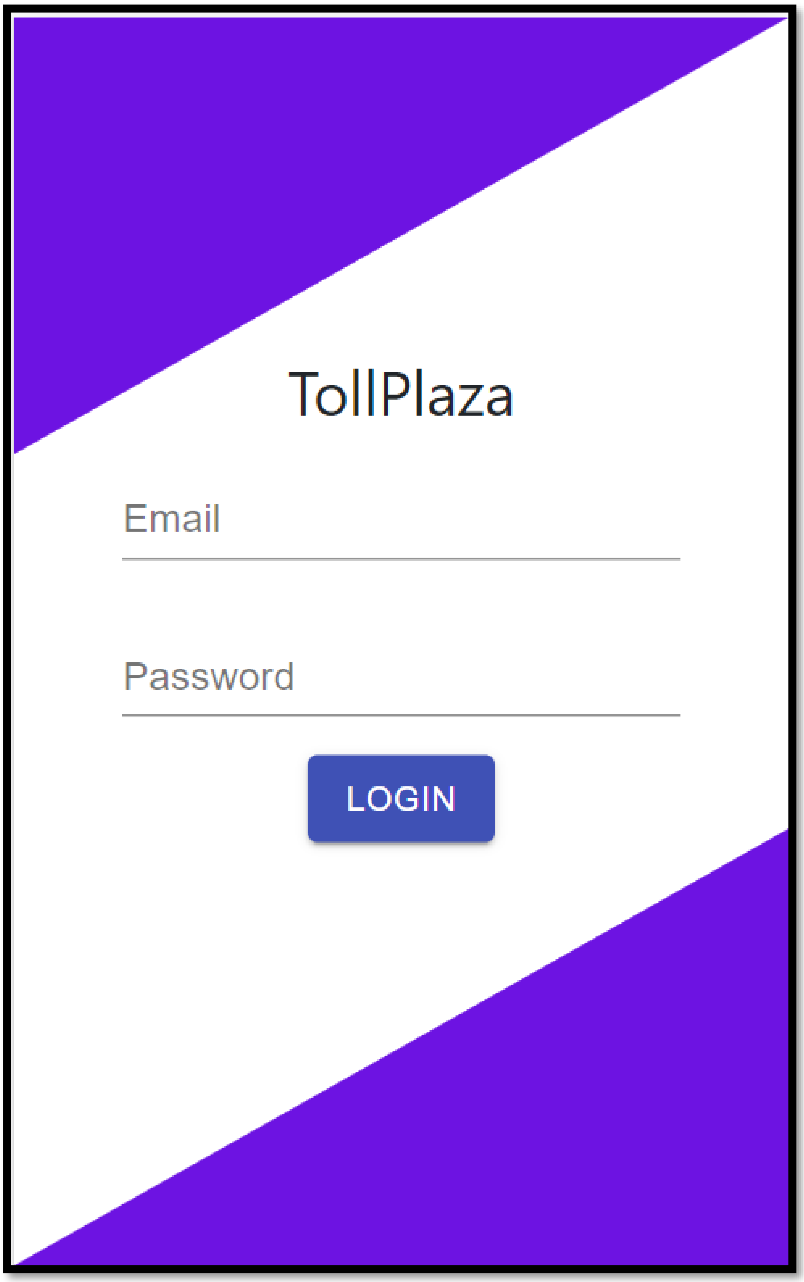



Fig 4.5 Login page layout

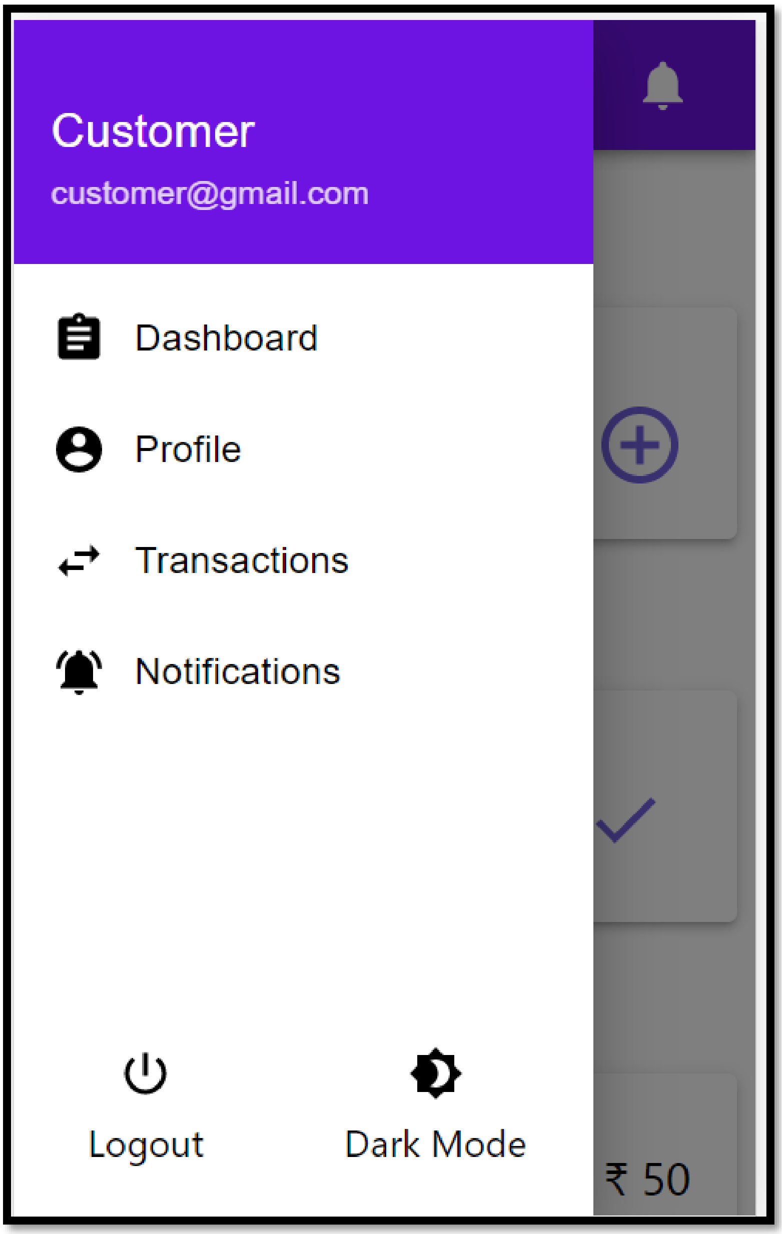



Fig 4.6 Content layout

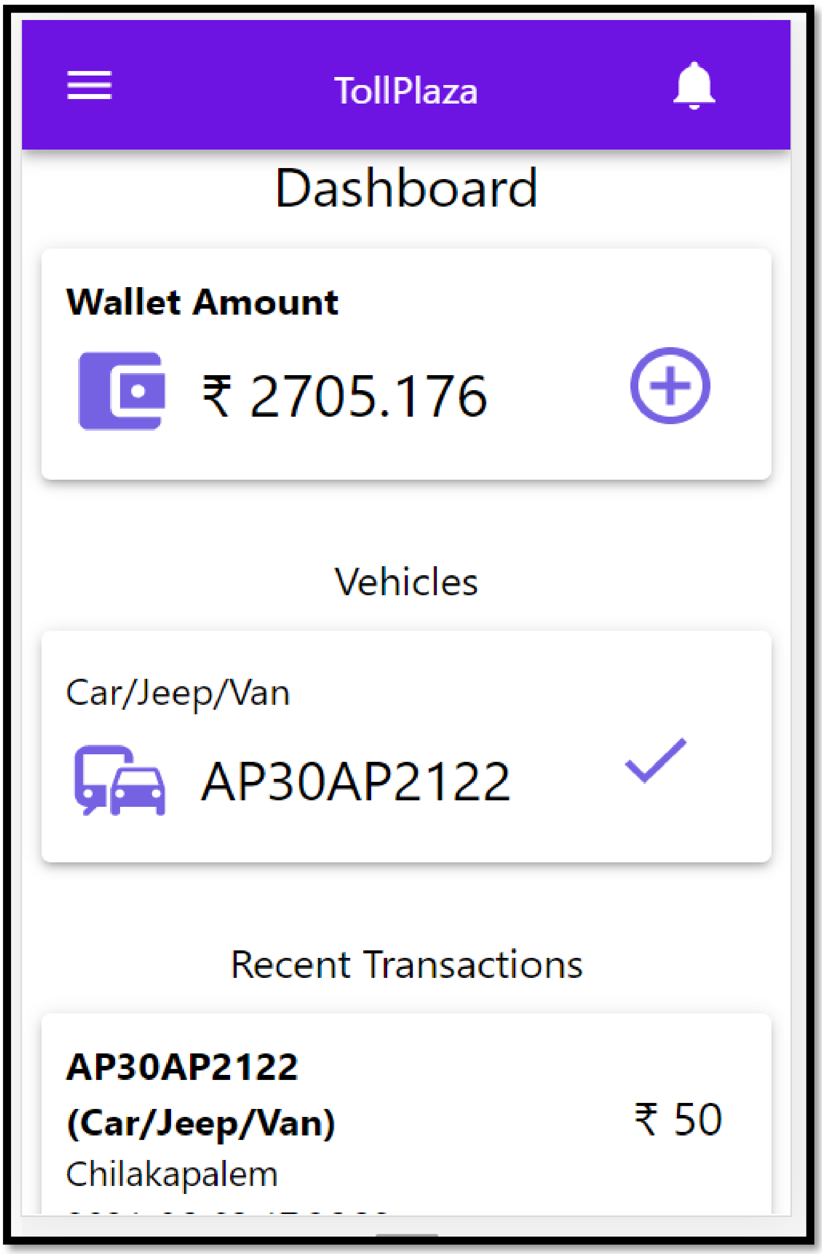



Fig 4.7 Dashboard layout

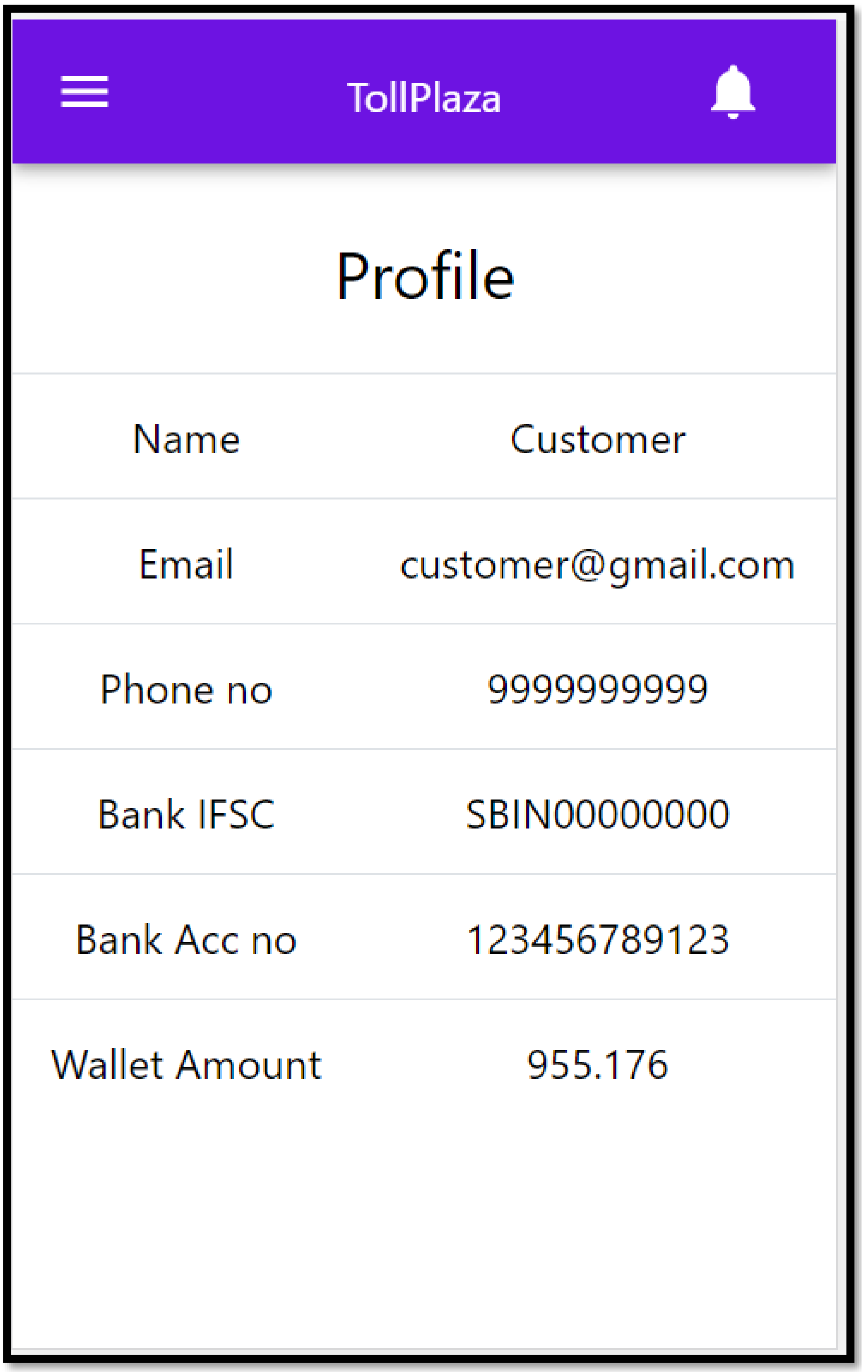



Fig 4.8 User's Profile layout

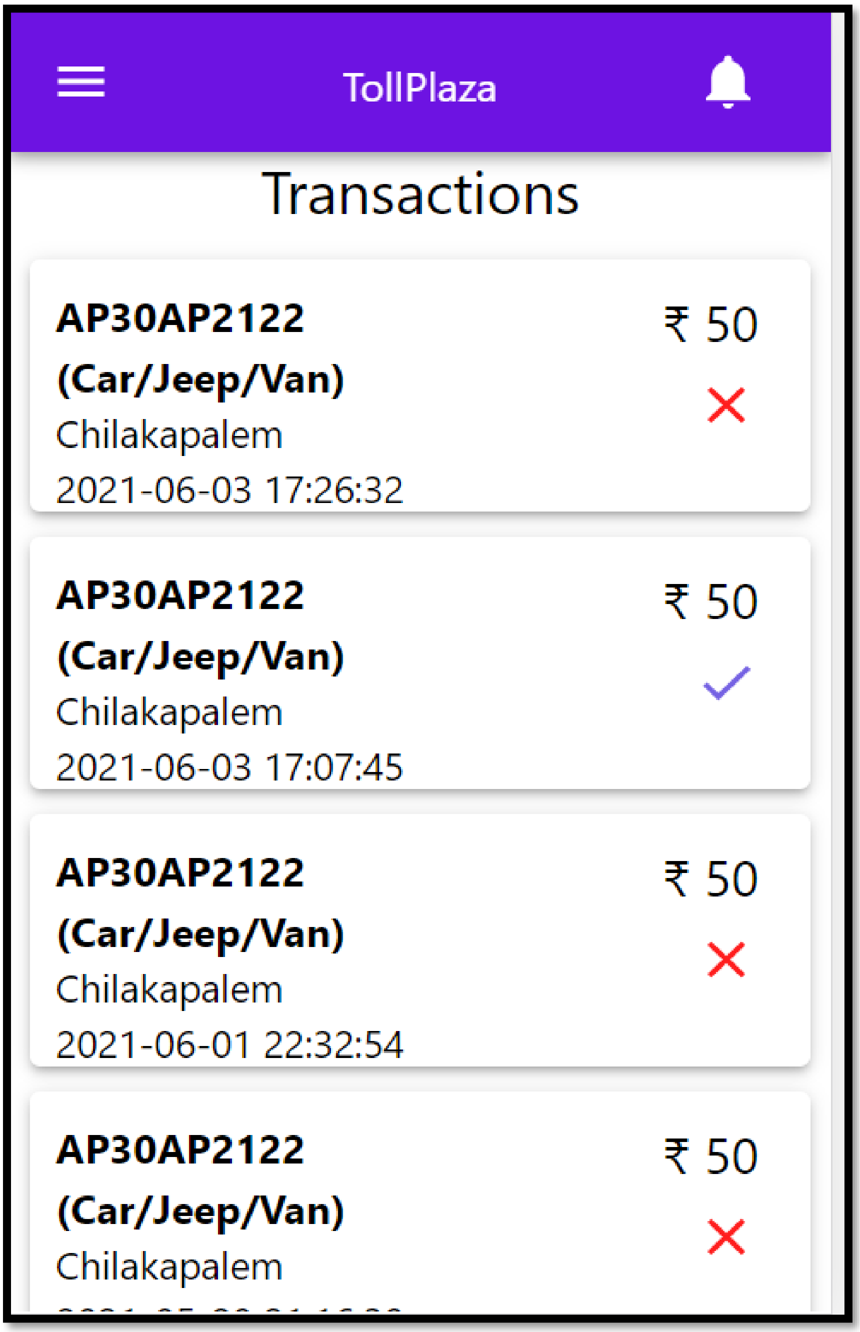



Fig 4.9 Customers Transactions

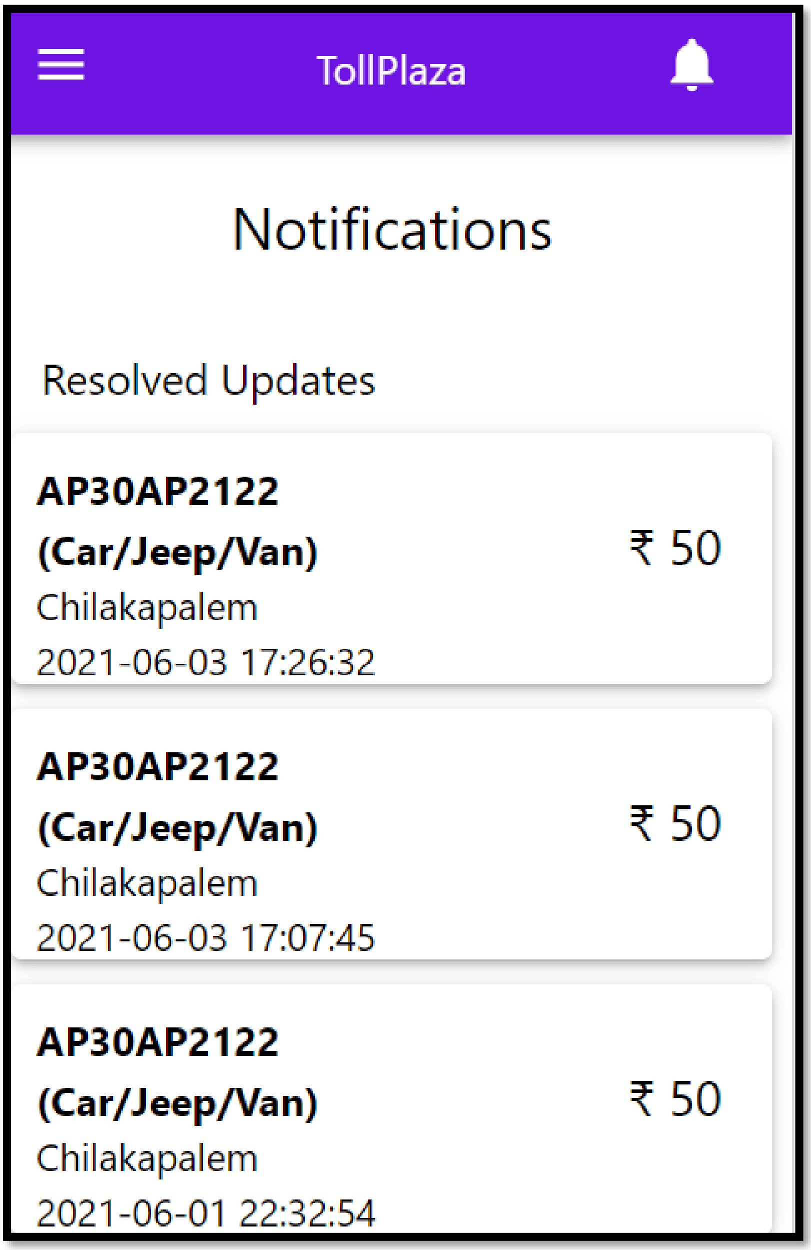



Fig 4.10 Notification layout

## 4.5 DATABASE TABLES:

| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | **cid** | int(255) | | | No | *None* |
| 2 | **email** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 3 | **password** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 4 | **name** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 5 | **phone** | varchar(15) | utf8mb4_general_ci | | No | *None* |
| 6 | **bankFSC** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 7 | **bankAccNo** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 8 | **balanceAmount** | double | | | No | *None* |

Fig 4.11 Customers Table

| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | **userid** | int(255) | | | No | *None* |
| 2 | **tid** | int(255) | | | No | *None* |
| 3 | **username** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 4 | **password** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 5 | **name** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 6 | **email** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 7 | **phoneno** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 8 | **usertype** | varchar(50) | utf8mb4_general_ci | | No | *None* |

Fig 4.12 Employee tabl

| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | **txnID** | int(255) | | | No | *None* |
| 2 | **timestamp** | timestamp | | | Yes | current_timestamp() |
| 3 | **vid** | int(255) | | | No | *None* |
| 4 | **vehicleNo** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 5 | **tid** | int(255) | | | No | *None* |
| 6 | **charge** | double | | | No | *None* |
| 7 | **confirmed** | int(10) | | | No | *None* |
| 8 | **images** | longblob | | | Yes | *NULL* |

Fig 4.13 Transaction Table

| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | **vid** | int(255) | | | No | *None* |
| 2 | **cid** | int(255) | | | No | *None* |
| 3 | **vehicleNo** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 4 | **chassisNo** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 5 | **vehicleType** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 6 | **lastTransaction** | timestamp | | | No | current_timestamp() |

Fig 4.14 Vehicle Information Table

| # | Name | Type | Collation | Attributes | Null | Default |
|---|---|---|---|---|---|---|
| 1 | **timestamp** | timestamp | | | Yes | current_timestamp() |
| 2 | **vehicleNo** | varchar(255) | utf8mb4_general_ci | | No | *None* |
| 3 | **reason** | varchar(16) | utf8mb4_general_ci | | Yes | *NULL* |
| 4 | **charge** | double | | | No | *None* |
| 5 | **bankFSC** | varchar(255) | utf8mb4_general_ci | | Yes | *NULL* |
| 6 | **bankAccNo** | varchar(255) | utf8mb4_general_ci | | Yes | *NULL* |
| 7 | **images** | longblob | | | Yes | *NULL* |

Fig 4.15 Rejected Transaction Table

# CHAPTER 5

# SIMULATED RESULTS

## 5.1 IMAGE PROCESSING RESULTS:

**1.**Original Image:


Fig 5.1 vehicle1


Fig 5.2 vehicle 2

2.Gray-scale Image:


Fig 5.3 Gray-Scale image of vehicle1


Fig 5.4 Gray-Scale image of vehicle2

3.Edge detection:


Fig 5.5 Edge detection for vehicle1


Fig 5.6 Edge detection for vehicle2

4.Contours:

Fig 5.7 All possible Contours in the vehicle1

Fig 5.8 All possible Contours in the vehicle2

5. Required Contour


Fig 5.9 Required contour of vehicle1


Fig 5.10 Required contour of vehicle2

## 5.2 MAIN RESULTS:

These are the two cases while transaction:

Case 1: If the user accepts the notification, then the user's transaction details are added to the transaction table in the database(as shown in below Fig ).

| txnID | timestamp | vid | vehicleNo | tid | charge |
|---|---|---|---|---|---|
| 1 | 2021-05-28 21:16:16 | 1 | AP30AP2122 | 1 | 100 |

Fig 5.11 Transaction table in the Database

Case 2: If the user rejects the notification or doesn't acknowledge the notification in a certain period, then the corresponding footage, transaction details are sent to the police database, or else the vehicle is unregistered, then automatically the corresponding footage is sent through the police database (as shown in below Fig ).

| timestamp | vehicleNo | reason | charge | bankFSC | bankAccNo | images |
|---|---|---|---|---|---|---|
| 2021-05-28 21:16:16 | AP30AP2122 | Rejected | 100 | SBIN00000000 | 123456789123 | NULL |
| 2021-05-28 21:16:26 | AP30AP2122 | Rejected | 50 | SBIN00000000 | 123456789123 | [BLOB - 49.8 KiB] |
| 2021-05-28 21:16:30 | AP30AP2122 | Not Acknowledged | 50 | SBIN00000000 | 123456789123 | [BLOB - 49.8 KiB] |
| 2021-05-28 21:42:16 | MH14BN7077 | unregistered | 50 | | | [BLOB - 49.8 KiB] |

Fig 5.12 Rejected Transaction table in the Database

# CHAPTER-6

# CONCLUSION

A literature survey is conducted on all proposed and existing toll collection system. Our project mainly focuses on maintain the advantages of FASTAG that is time saving, reduced traffic congestion, reduced consumption of inessential fuel and reducing the drawbacks of it. Along with it also reduces crime activities.

Thus, an efficient, less time-consuming method has been devised through this paper which can be more useful in toll fare collection. It promises to be less prone to crimes and errors if implemented in suitable conditions.

The proposed system works quite well however, there are still areas for improvement. The system robustness and speed can be increased if high resolution camera is used. High accuracy can be obtained by using neural networks, deep learning which is more accurate compared to image processing.

# REFERENCES:


[1] **Pushpajit A. Khaire and Dr. N. V. Thakur,** ʊA Fuzzy Set Approach for Edge Detection , International Journal of Image Processing (IJIP), Volume 6 Issue 6, pp. 403-412, 2012.

[2] **Xiaojun Zhai, Faycal Bensaali and Reza Sotudeh**, “OCR-Based Neural Network for ANPR", IEEE International Conference on Imaging Systems and Techniques (IST), pp. 393-397, 2012.

[3] **H. Nezamabadi-Pour, S. Saryazdi, and E. Rashedi,** “Edge detection using ant algorithms,” Soft Computing, vol. 10, pp. 623–628, May 2006.

[4] Automatic toll collection using RFIDfunctioning. **1.Pranoti Salunke, 2. Poonam Malle, 3.Kirti Datir, 4. Jayshree Dukale.** IOSR Journal ofComputer Engineering (IOSR-JCE), e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 2 (Jan. -Feb. 2013), PP 61-66.

[5] Smart highway electronic toll collection system. **1.Ganesh K. Andurkar, 2. Vidya R. Ramteke.**

[6] Automatic Detection and Counting of Vehicles Based on Image Processing **R. Rajamunipriya1, S. Dinesh2.International Journal of Science and Research (IJSR).**

[7] Multiple license plate detection for complex background; **Ching-Tang Hsieh, Yu-Shan Juan, Kuo-Ming Hung ,** \emph{Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA’05) 2005 IEEE}.

[8] **Zhen Bin, Mamoru Kobayashi, and Masashi Shimizu.** Framed aloha for multiple RFID objects identification. IEICE Transactions on Communications, 88(3):991–999, 2005.

[9] **M. Tahir Qadri, M. Asif** “Automatic Number Plate Recognition System for Vehicle Identification using OCR,” International Conference on Education Technology and Computer, pp 335 – 338, 2009.

[10] **V. Koval, V. Turchenko, V. Kochan, A. Sachenko, G. Markowsky** “Smart License Plate Recognition System Based on Image Processing

Using Neural Network," IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp 123 – 127, 2003.