REAL TIME TRAFFIC CONGESTION ANALYZER FOR ROAD SAFETY

Mini Project 2018

Done By

ELISA SANTHOSH

JABIR ALI V

Guided By

Prof.Sini S Raj

Assisstant Professor

Department of Computer Applications
College of Engineering
Trivandrum-695016

ABSTRACT

Expressways, highways and roads are becoming overcrowded with increasing of large number of vehicles. Vehicle detection and tracking plays an effective and significant role in the area of traffic surveillance system where efficient traffic management and safety is the main concern. Automatic recognition of vehicle data has been widely used in the vehicle information system and intelligent traffic system. It has acquired more attention of researchers from the last decade with the advancement of digital imaging technology and computational capacity. Automatic vehicle detection systems are keys to road traffic control nowadays; some applications of these systems are traffic response system, traffic signal controller, lane departure warning system, automatic vehicle accident detection and automatic traffic density estimation.

An Automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. There are several methods for vehicle detection and counting proposed so far. The proposed a real-time traffic analyzer includes, tracking, and counting the vehicles using Blob Detection methods. The proposed system, first, differentiate the foreground from background in frames by learning the background. Here, foreground detector detects the object and a binary computation is done to define rectangular regions around every detected object. To detect the moving object correctly and to remove the noise some morphological operations have been applied. Then the final counting is done by tracking the detected objects and their regions. Finally, vehicles in a predefined virtual detection zone are recorded and counted. The traffic details and counts are continuously updated in the server using a timer and users can register in an Android app to get notified about traffic in real time.

ACKNOWLEDGEMENT

If words are considered as symbols of approval and tokens of acknowledgment, then let words play the heralding role of expressing our gratitude.

First of all we would like to thank God almighty for bestowing us with wisdom,courage and perseverance which had helped us to complete this project ***REAL TIME TRAFFIC CONGESTION ANALYZER FOR ROAD SAFETY***.This project has been a reality as a result of the help given by a large number of personalities.

We would like to remember with gratitude **Prof. Jose T Joseph**,Head Of Department Department of Computer Applications, College of Engineering, Trivandrum for the encouragement and guidance rendered.

We express our profound and sincere gratitude to **Prof.Baby Syla L**, Assistant Professor, Department of Computer Applications, College of Engineering Trivandrum for the immense support and guidance provided by them.

We express our sincere thanks to **Prof.Sini S Raj**,Assistant Professor, Department of Computer Applications, College of Engineering Trivandrum for his valuable guidance, support and advices that aided in the successful completion of my project.

Finally, We wish to express our sincere gratitude to all our friends, who directly or indirectly contributed in this venture.

ELISA SANTHOSH
JABIR ALI V

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Vehicle detection and tracking plays an effective and significant role in the area of traffic surveillance system where efficient traffic management and safety is the main concern. The vehicle detections can be traditionally achieved through inductive loop detector, infrared detector, radar detector or video-based solution. Compared to other techniques, the video-based solutions based on surveillance camera mounted outdoor are easily influenced by environments such as weather, illumination, shadow, etc. However, because video-based systems can offer several advantages over other methods such as traffic flow undisturbed, easily installed, conveniently modified, etc., they have drawn significant attention from researchers in the past decade.

Automatic recognition of vehicle data has been widely used in the vehicle information system and intelligent traffic system. It has acquired more attention of researchers from the last decade with the advancement of digital imaging technology and computational capacity. Automatic vehicle detection systems are keys to road traffic control nowadays; some applications of these systems are traffic response system, traffic signal controller, lane departure warning system, automatic vehicle accident detection and automatic traffic density estimation.

An Automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. There are several methods for vehicle detection and counting proposed so far.

The proposed real time traffic analyzer for road safety includes, tracking, and counting the vehicles using Blob Detection methods. In this project, first, we differentiate the foreground from background in frames by learning the background. Here, foreground detector detects the object and a binary computation is done to define rectangular regions around every detected object. To detect the moving object correctly and to remove the noise some morphological operations have been applied. Then the final counting is done by tracking the detected objects and their regions. Finally, vehicles in a predefined virtual detection zone are recorded and counted. The traffic details and counts are continuously updated in the server using a timer and users can register in an Android app to get notified about traffic in real time. Thus, the proposed system consists of three main modules. The first module constitutes the Android application for user registration and to retrieve notifications regarding traffic in real time. The second module deals with the vehicle detecting and counting and the count is updated to the server with the help of a timer. The third module is the server, which stores user details, traffic details, routes and traffic updates.

Chapter 2

Requirement Analysis

2.1 Purpose

The purpose of this proposed work is to build a system to detect the traffic flow on the basis of monitoring video footages placed in roads for detecting congestions. The system will provide an app for the user to register about his route details. If the specified route has traffic, the server that store every traffic details in real time will notify him back through the app. Due to the result of the increase in vehicle traffic, many problems have appeared such as traffic accidents, traffic congestion, traffic induced air pollution and so on. Traffic congestion has been a significantly challenging problem. It has widely been realized that increases of preliminary transportation infrastructure, more pavements, and widened road, have not been able to relieve city congestion. To processes the information and monitors the results as better understand traffic flow, an increasing reliance on traffic surveillance is in a need for better vehicle detection at a wide-area. Automatic detecting vehicles in video surveillance data is a very challenging problem in computer vision with important practical applications, such as traffic analysis and security.

The scope of the proposed system is to develop an automatic vehicle counting system, which can process videos recorded from stationary cameras over roads e.g. CCTV cameras installed near traffic intersections or junctions and counting the number of vehicles passing a spot in a particular time for further collection of vehicle / traffic data. A simple approach was carried out to tackle the problem by using Pixel Based Adaptive Segmenter - based object detection, a non-predictive regional tracking and a counting of tracked objects based on simple rules. The proposed system includes an android application for the registration of users. They can update the user details and route details. The vehicle count calculated is sent to the server by the OpenCV application where registered users details, traffic details are also saved and will notify the user in real time.

2.2 Overall Description

This system structure is computationally efficient and can run in a real-time basis while retaining very respectable detection rates. The appearance of larger vehicle or vehicle's shadow occluding the adjacent lanes also is known to trigger false detection. Consequently, the merit of using computer vision as a surveillance tool has been limited by focusing strictly on building reliable systems that can perform in real-time.

The system comprises of three main modules. First module deals with an Android Application where users register with their details, update route details including source, destination, date and time for travel. Second module is an OpenCV application for tracking and detecting vehicles and vehicle count is updated to server in real time. The OpenCV application makes use of an existing video sequence. The first frame is considered as the reference frame. The subsequent frames are taken as the input frames. They are compared and the background is eliminated. If a vehicle is present in the input frame, itll be retained. The detected vehicle is thus tracked by various techniques. Vehicle detection is done by using Background Subtraction (BS) algorithm and for tracking blob tracker algorithm can be used. Third module is a PHP web server, where all the user and traffic details are stored and will notify the user in real time.

Product Functions

The main functions of the proposed product includes:

- Detects and count the vehicles
- Predicts Traffic flows
- Can be used to predict accident zone areas
- Ensures road safety
- Notify registered users about traffic jam

Operating Environment

The operating environment required are:

- **Hardware Requirements**

  Processor : Intel i3
  Storage : 5 GB hard disk space
  Memory : 8 GB RAM

- **Software Requirements**

  Operating system : Linux
  Database : MySQL
  Frameworks : Python,c++,java,php,cmake
  Server : Apache2

2.3 Functional requirements

Functional requirements represent the intended behavior of the system. This behavior may be expressed as services, tasks or functions that the specified system is required to perform. The following functional requirements have been identified for this project. The proposed system consists of 3 modules. They are given below:

- **ANDROID APPLICATION**

  In order to make the proposed system user friendly, an android application is used to collect the user details. Users can register through the app, can enter details of their journey including source, destination and the route they are going through. From the details stored in the RTA server, they will get an update about the traffic conditions and vehicle count through the android application.

- **OpenCV APPLICATION FOR VEHICLE DETECTION AND COUNTING**

  An Automatic vehicle detecting and counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. It is just the ability of automatically extract and recognize the traffic data e.g. total number of vehicles, vehicle number and label from a video. In each video frame, Pixel Based Adaptive Segmenter differentiates objects in motion from the background by tracking detected objects inside a specific region of the frame, and then counting is carried out. In this system, blob detection uses contrast in a binary image to compute a detected region, its centroid, and the area of the blob. The supplied pixels detected the foreground. These pixels are grouped, in current frame, together by utilizing a contour detection algorithm. The contour detection algorithm

groups the individual pixels into disconnected classes, and then finds the contours surrounding each class.
Each class is marked as a candidate blob (CB). These CB are then checked by their size and small blobs are removed from the algorithm to reduce false detections. The positions of the CB, in current frame, are compared using the k-Means clustering that finds the centers of clusters and groups the input samples CB around the clusters to identify the vehicles in each region. The moving vehicle is counted when it passes the base line. When the vehicle passes through that area, the frame is recorded. In each region the blob with the same label are analyzed and the vehicle count is incremented.

- **Real-time Traffic Analyzer SERVER**

  OpenCV application detects and counts the vehicle and the server gets updated with this count in every 15 minutes. Counting vehicles gives us the information needed to obtain a basic understanding over the flow of traffic in any region under surveillance. The total count of vehicles, including other traffic details such as source and destination of user are stored on the server. This will help to make a traffic analysis. The RTA server will notify the user in real time.

2.4 Non Functional requirements

Non-Functional requirements define the general qualities of the software product. Non-functional requirement is in effect a constraint placed on the system or the development process. They are usually associated with the product descriptions such as maintainability, usability, portability, etc. it mainly limits the solutions for the problem. The solution should be good enough to meet the non-functional requirements.

Performance Requirements

- Accuracy: Accuracy in functioning and the nature of user-friendliness should be maintained in the system.

- Speed: The system must be capable of offering speed.

Quality Requirements

- Scalability: The software will meet all of the functional requirements.
- Maintainability: The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.
- Reliability: The acceptable threshold for down-time should be long as possible. i.e.mean time between failures should be large as possible. And if the system is broken, time required to get the system back up again should be minimum.

Chapter 3

Design And Implementation

3.1 Overall Design

An Automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. Automatic vehicle detection systems are keys to road traffic control nowadays; some applications of these systems are traffic response system, traffic signal controller, lane departure warning system, automatic vehicle accident detection and automatic traffic density estimation. An Automatic vehicle counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene.

The system comprises of three main modules. First module deals with an Android Application where users register with their details, update route details including source, destination, date and time for travel. Second module is an OpenCV application for tracking and detecting vehicles and vehicle count is updated to server in real time. The OpenCV application makes use of an existing video sequence. The first frame is considered as the reference frame. The subsequent frames are taken as the input frames. They are compared and the background is eliminated. If a vehicle is present in the input frame, itll be retained. The detected vehicle is thus tracked by various techniques. Vehicle detection is done by using Background Subtraction (BS) algorithm and for tracking blob tracker algorithm can be used. Third module is a PHP web server, where all the user and traffic details are stored and will notify the user in real time.

The system comprises of three main modules.

First module deals with an Android Application where users register with their details, update route details including source, destination, date and time for travel.

Second module is an OpenCV application for tracking and detecting vehicles and vehicle count is updated to server in real time. A video sequence

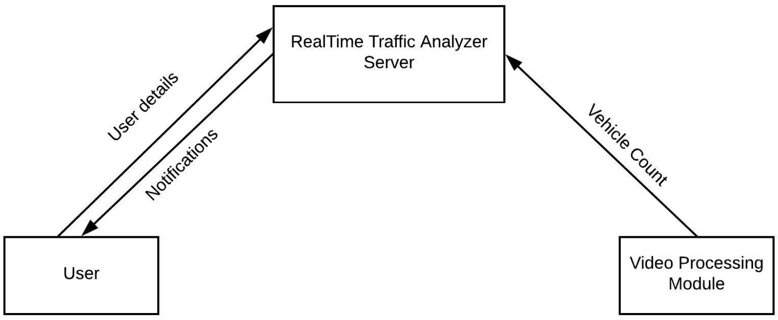

Figure 3.1: System Architecture

of road can be processed and analyzed to detect and count vehicles. Further information, such as the speed of a vehicle or traffic density, can also be calculated by using computer vision. This would directly benefit to two groups of people: road users and traffic administrators. If road users know the real-time traffic information, they can then use the information to choose the best way for traveling and can avoid traffic congestion. On the other hand, traffic administrators can utilize the traffic information in their traffic control systems, resulting to a better traffic management. There are several methods for vehicle detection and counting proposed so far. The real time traffic congestion analyzer for road safety includes, tracking, and counting the vehicles using Blob Detection methods. First, we differentiate the foreground from background in frames by learning the background. Here, foreground detector detects the object and a binary computation is done to define rectangular regions around every detected object. To detect the moving object correctly and to remove the noise some morphological operations have been applied. Then the final counting is done by tracking the detected objects and their regions. Finally, vehicles in a predefined virtual detection zone are recorded and counted.

Third module is the server, where all the user and traffic details are stored.The camera footages are processed from client machines located at different locations and traffic details such as average vehicle count per frame in unit time,number off vehicles passed in unit time are periodically sent to the server. The user logins to server and enter journey details.server periodically keep track of traffic rich locations and notify the active users who travels through those routes

3.1.1 System Design

Module Description

The proposed system consists of 3 modules:

- ANDROID APPLICATION

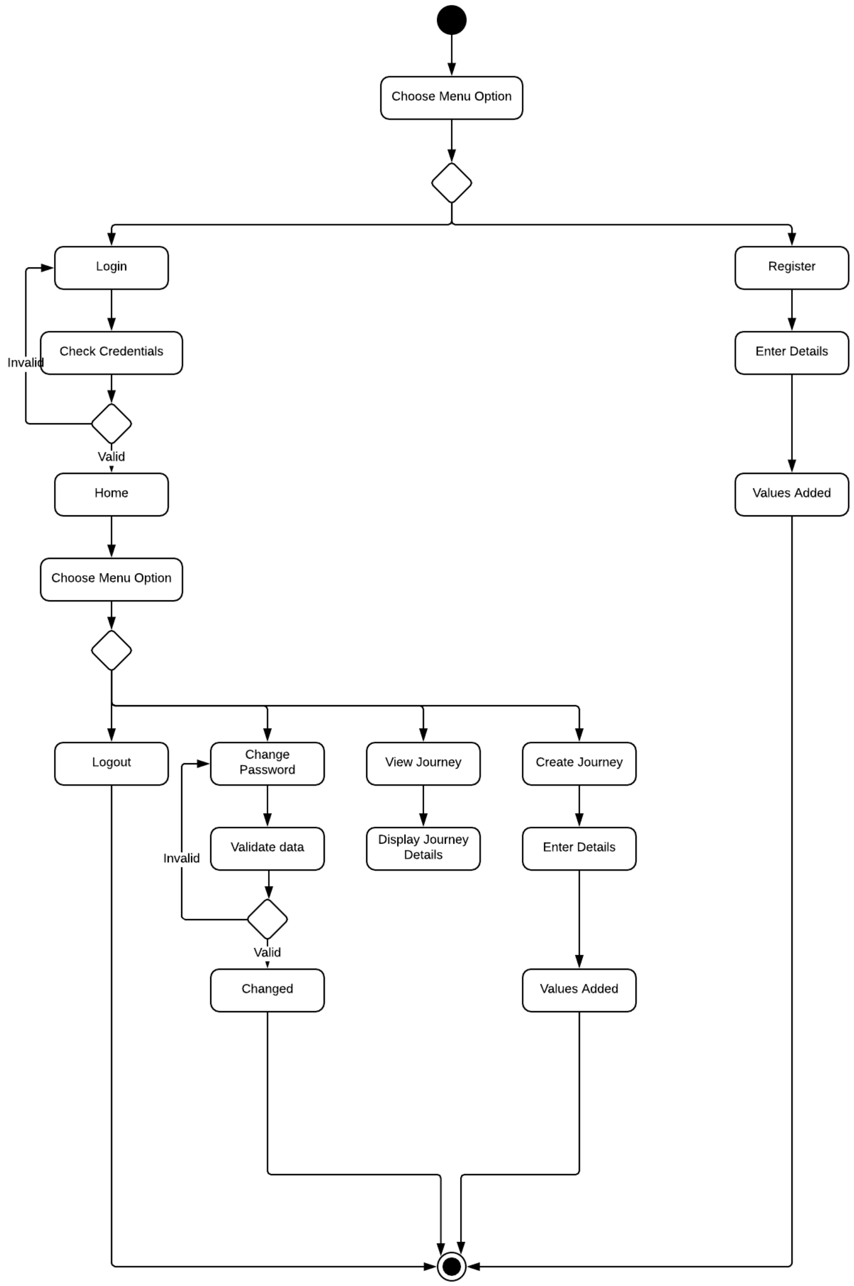

Figure 3.2: Activity Diagram

In order to make the proposed system user friendly, an android application is used to collect the user details. Users can register through the app, can enter details of their journey including source, destination and the route they are going through. From the details stored in the RTA server, they will get an update about the traffic conditions and vehicle count through the android application. The activity diagram is shown above

- VEHICLE DETECTION AND COUNTING

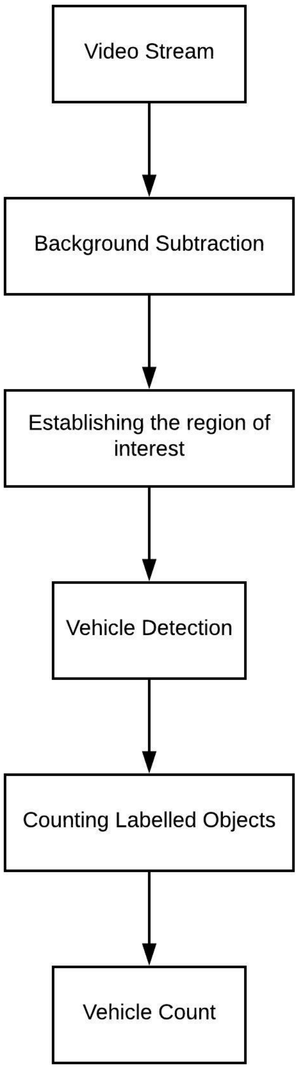

Figure 3.3: Vehicle Counting Flowchart

A simple approach was carried out for vehicle detection and counting using Pixel Based Adaptive Segmenter - based object detection, a non-predictive regional tracking and a counting of tracked objects based on simple rules. Pixel-Based Adaptive Segmenter, follows a non-parametric paradigm. Thus, every pixel is modeled by an array of recently observed background values. the decision block decides for or against foreground based on the current image and a background model. This decision is based on the per-pixel threshold. Moreover, the background model has to be updated overtime in order to allow for gradual background changes.

This module includes four sub-modules, which is discussed below:

– BACKGROUND SUBTRACTION

  Background Subtraction is improved by combining adaptive background generation with two-frame differencing algorithm. We assume that the video stream is captured with a RGB 24 bits format. We use the luminance component of the RGB image to estimate the motion for each frame. Once we have a robust background model, we can use a method called background subtraction to segment each frame into foreground and background objects. A pixel would be part of the foreground, when its value is different enough from its corresponding value in the background model. The main difficulty is to evaluate the distance of each pixel in a color frame (in RGB color space) to the corresponding background pixel. This evaluation allows the classification of all the current image pixels in two categories (foreground and background). In some situations, an oversimplification of the method (e.g. a binarization with a static threshold) may cause erroneous segmentation. In most cases, a simple binarization is not sufficient to obtain a clear foreground. We use a morphological closing to fills the missing foreground pixels and a morphological opening to remove the small isolated foreground pixels.

– ESTABLISHING THE REGION OF INTEREST

  A region of interest (ROI), which will be further processed, as the next step. The user can set the region of interest initially by drawing a line hoizontally or vertically across the video screen. This line seperates the video into two regions, say region A and region B. The vehicle that passes the ROI will be counted and the total count will be passed.

– VEHICLE DETECTION

  In this step, only pixels in the ROI are considered while the others are deleted. Here, we detect the moving objects using a Background Subtraction (BS) algorithm. For vehicle tracking, we need to use a tracking algorithm called blob tracker algorithm. Then send the foreground mask to cvBlob or OpenCVBlobsLib. The cvBlob library provide some methods to get the centroid, the track and the ID of the moving objects. A bounding rectangle is drawn to track the tracked object and is checked whether this tracked object's centroid passes the ROI.

– VEHICLE COUNTING

The main object of this part is to count and register the vehicle flow for each lane. The count of the vehicles that passes the ROI region is the vehicle count and the count of the bounding boxes is the traffic count in that specific area. If there is no vehicle that cross the ROI, it means that vehicle count should be zero. Both the vehicle count and bounding boxes count are updated in the server.

- Real-time Traffic Analyzer SERVER

  OpenCV application detects and count the vehicles. Counting vehicles gives us the information needed to obtain a basic understanding over the flow of traffic in any region under surveillance. The total count of vehicles, including other traffic details such as source and destination of user are stored on the server. Thus server keeps a track of vehicle count from different locations and checks active users in the route. This will help to make a traffic analysis. The RTA server gets updated on every 15 minutes and notify the user back through the android application.

DataFlow Diagram

A data flow diagram (DFD) is a design tool to represent the flow of data through an informationsystem.A context level DFD can be used to show the interaction between a system and outside entities; it can also show the internal data flows within a system. ram. It often shows the information system as a single circular shape with no details of its inner workings: what it shows is its relationships with the external entities.A data flow diagram graphically represents:

- processes - jobs that are done with the data. A process transforms incoming data flow into outgoing data flow.
- data stores - files, databases, archives. They can be manual, digital or temporary.
- external entities - other systems or people beyond the control of the current system.
- connecting data flows - arrows show how data flows from one place to another.

Notations in a Data Flow Diagram

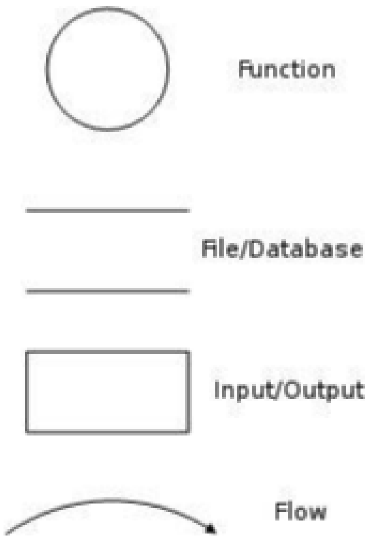

Figure 3.4: Notations in dataflow diagram

Context Diagram (Level 0)

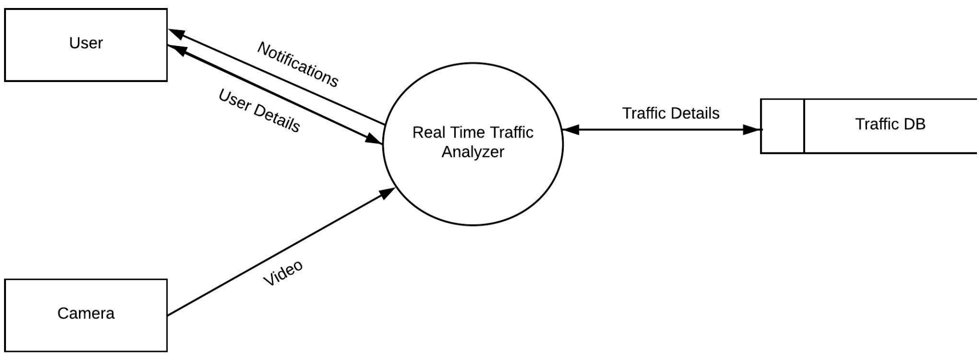

Figure 3.5: Level 0 DFD

Top Level DFD(Level 1)

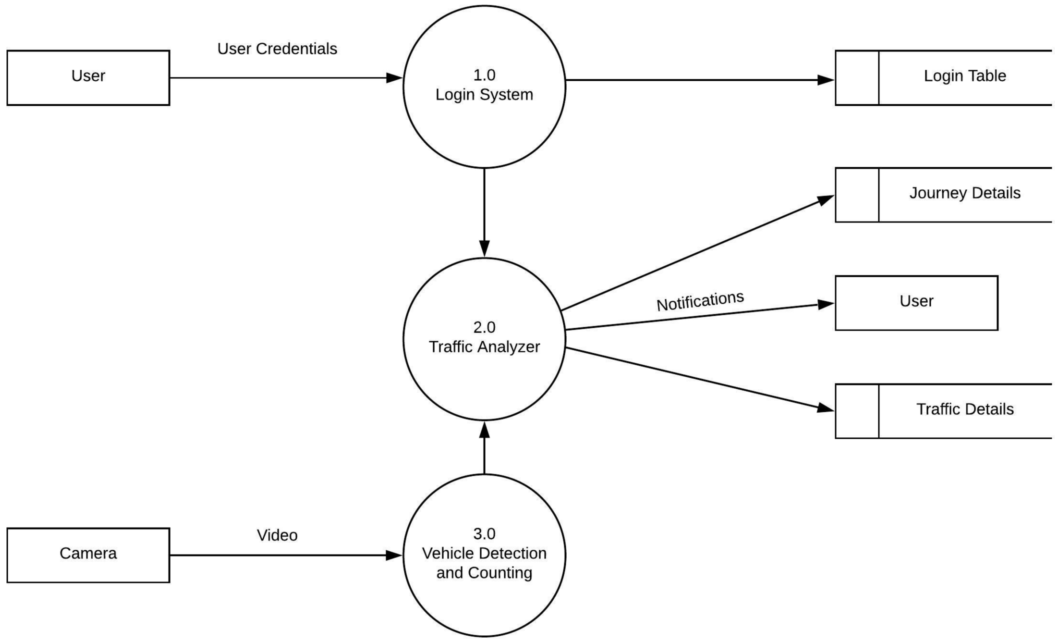

Figure 3.6: Level 1 DFD

Level 2

Level 3

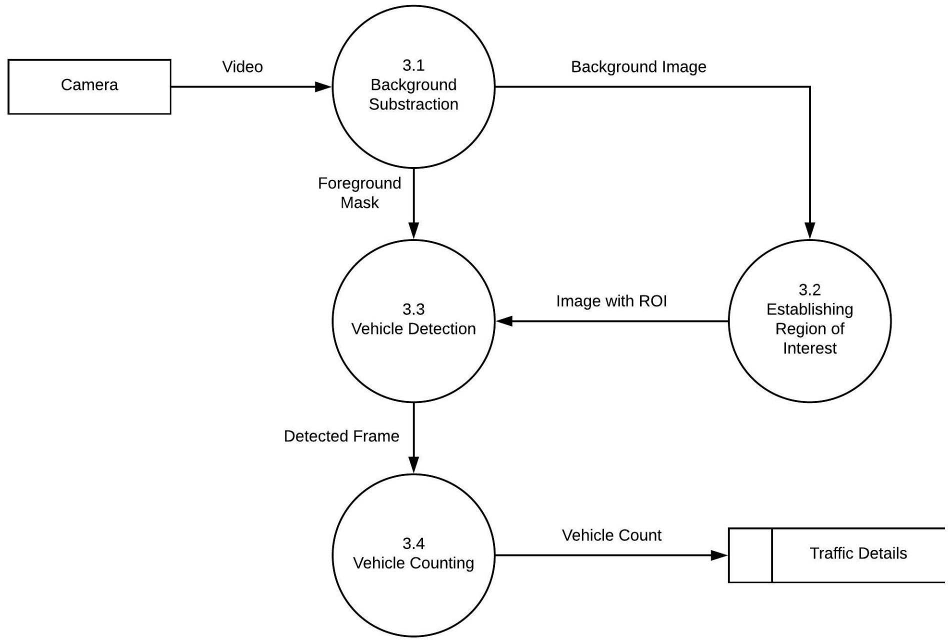

Figure 3.7: Level 2 DFD

3.1.2 Database Design

Table name: login
Description: Login details
Constraints: uname (primary key)

Field	Data type	Description
uname	varchar(10)	username
password	varchar(20)	password
usertype	varchar(10)	type of the user

Table 3.1: Login Details

Table name: route
Description: Route Details
Constraints: routeid (primary key)

Field	Data type	Description
routeid	int(5)	Route id
routename	varchar(20)	Route name
maxvcount	int(5)	vehicle count

Table 3.2: Route Details

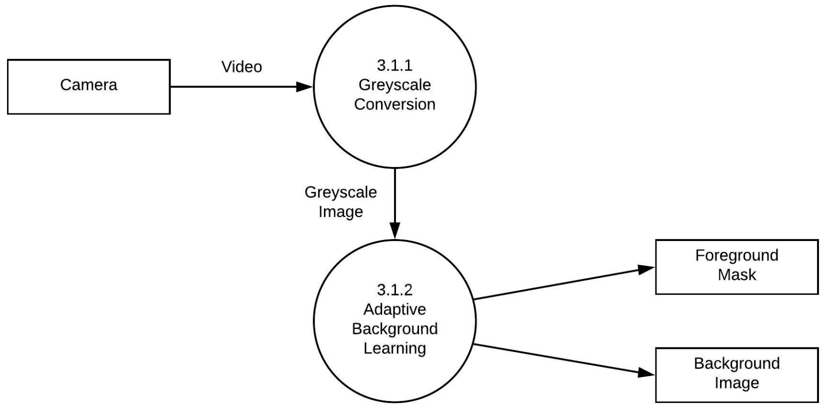

Figure 3.8: Level 3 DFD

Table name: users
Description: User details
Constraints: id (primary key)

Field	Data type	Description
id	int(20)	user id
username	varchar(70)	username
password	varchar(40)	password
email	varchar(50)	email id
createdat	datetime	registered time
updatedat	datetime	updated time
name	varchar(50)	name of the user
mobile	bigint(10)	mobile no
status	varchar(10)	status of the user

Table 3.3: User Details

Table name: journeydetails
Description: Journey details
Constraints: id (primary key)

Table name: activejourney
Description: Active Journey details
Constraints: id (primary key)

Field	Data type	Description
id	int(20)	id number
userid	int(20)	userid, Foreign key
sdate	date	start date
stime	time(50)	start time
routeid	int(5)	routeid, Foreign key
source	varchar(20)	Source
destination	varchar(20)	destination
duration	tinyint(4)	duration of the journey

Table 3.4: Journey Details

Field	Data type	Description
id	int(5)	Active Journey id
routeid	int(5)	Route id, Foreign key
date	date	Journey Date
roicount	int(5)	Traffic count
avgcount	int(5)	Total vehicles count

Table 3.5: Active Journey Details

3.1.3 User Interface Design

Figure 3.9: Splash screen for Android Application

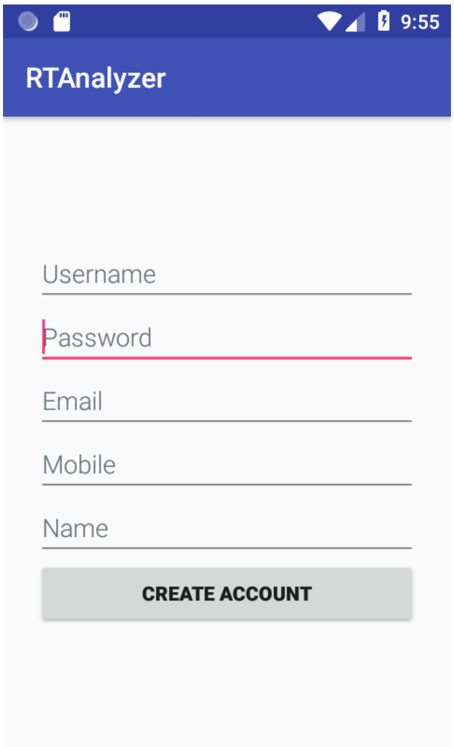

Figure 3.10: User Registration

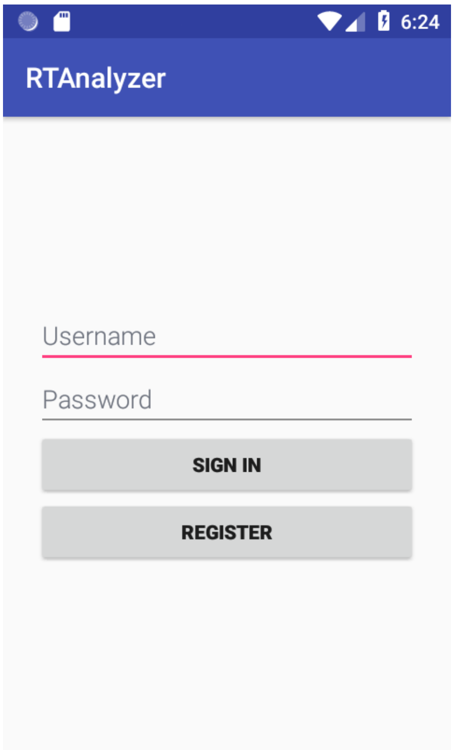

Figure 3.11: Login page

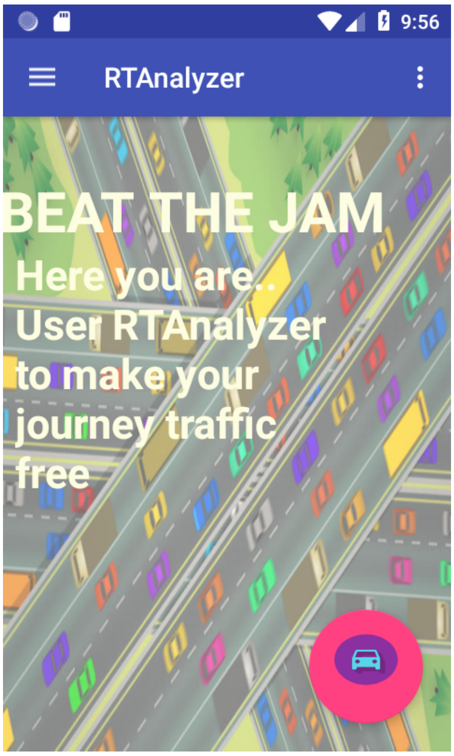

Figure 3.12: Home page

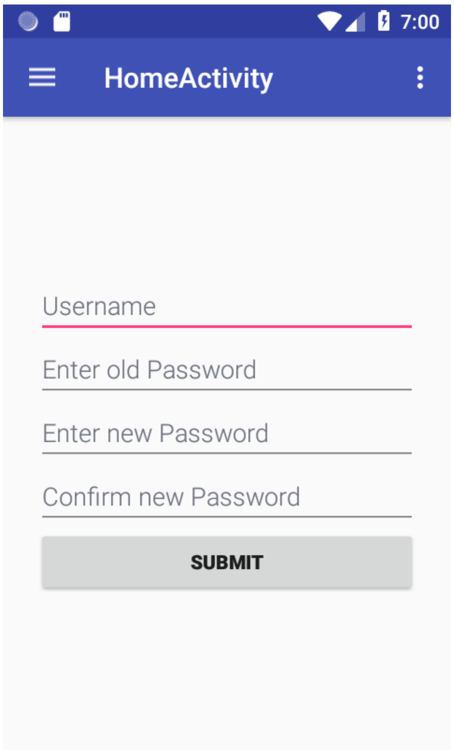

Figure 3.13: Change Password page

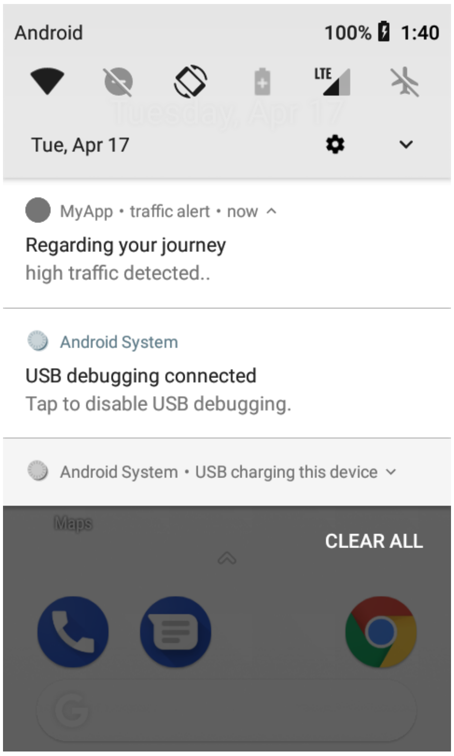

Figure 3.14: Notification

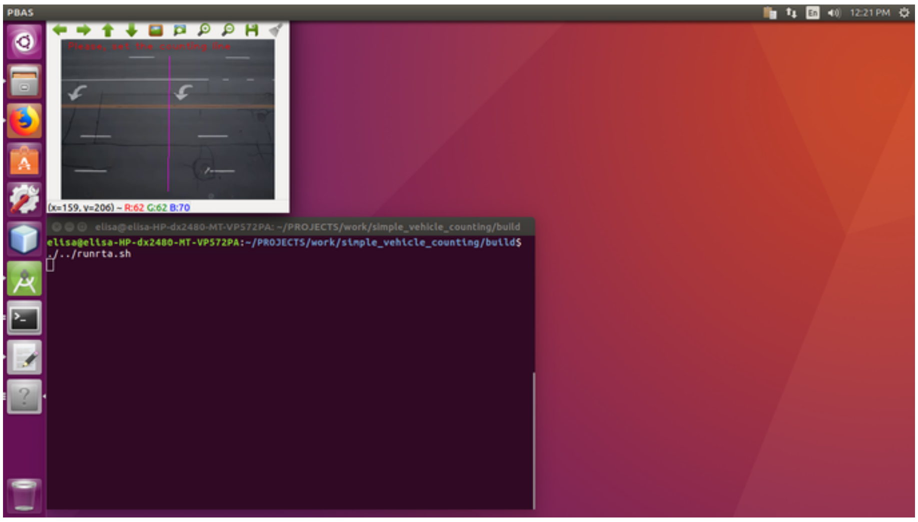

Figure 3.15: Setting Region of Interest

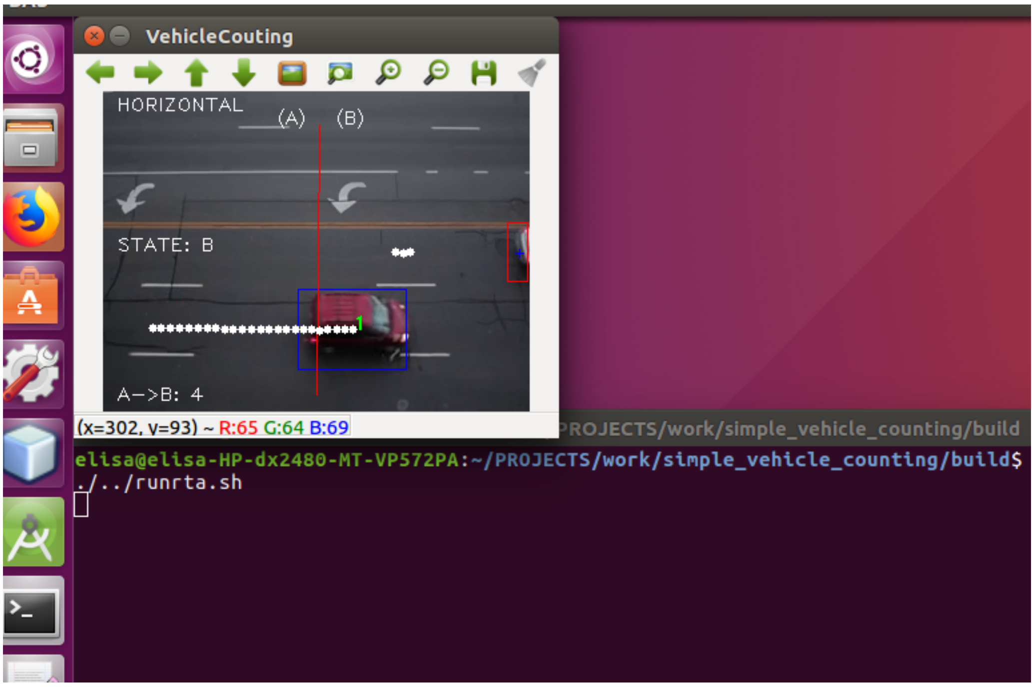

Figure 3.16: Vehicle Counting

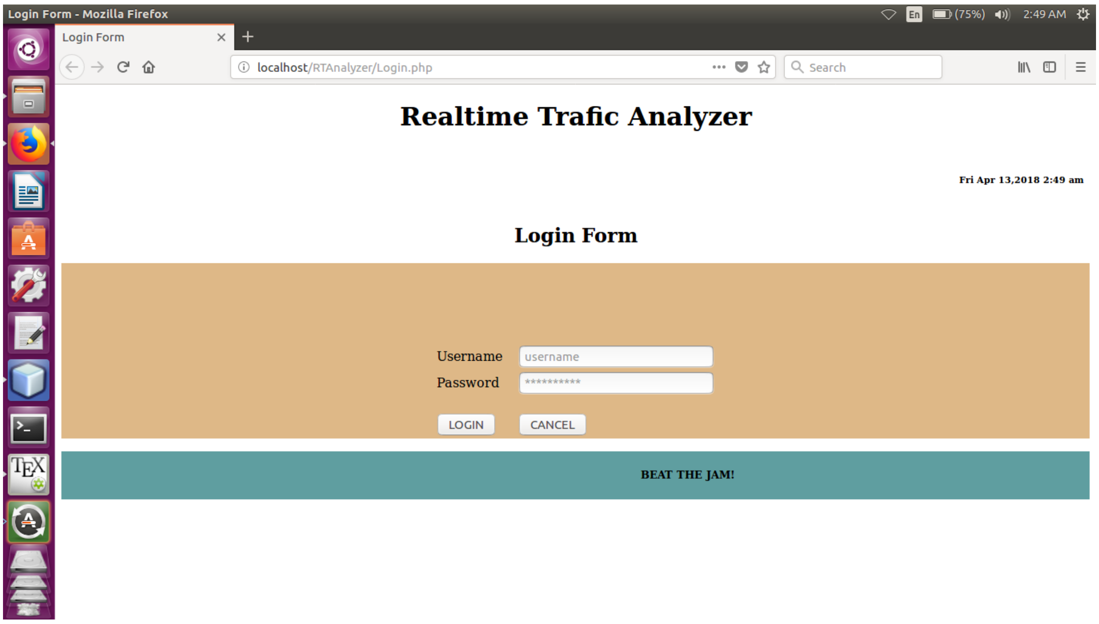

Figure 3.17: Login Page of server

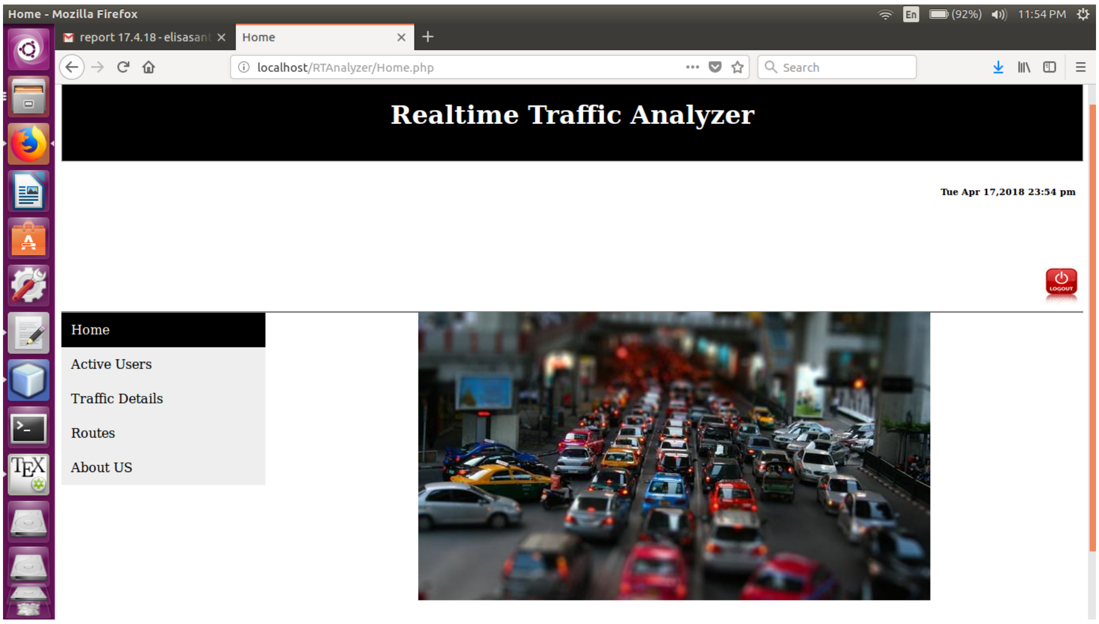

Figure 3.18: Home page of server

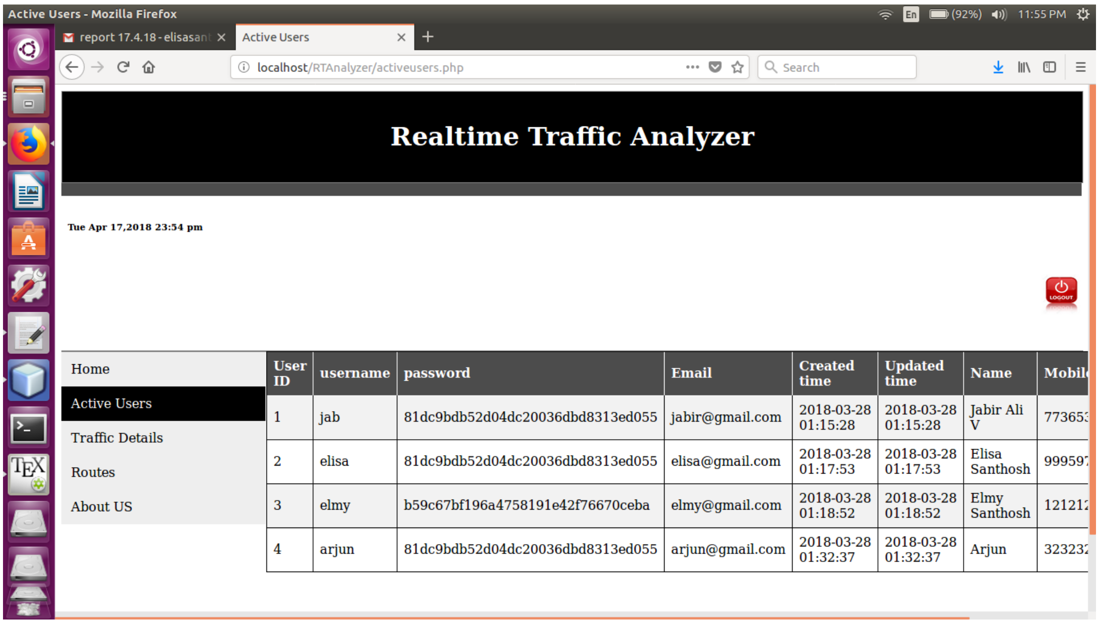

Figure 3.19: Active Users

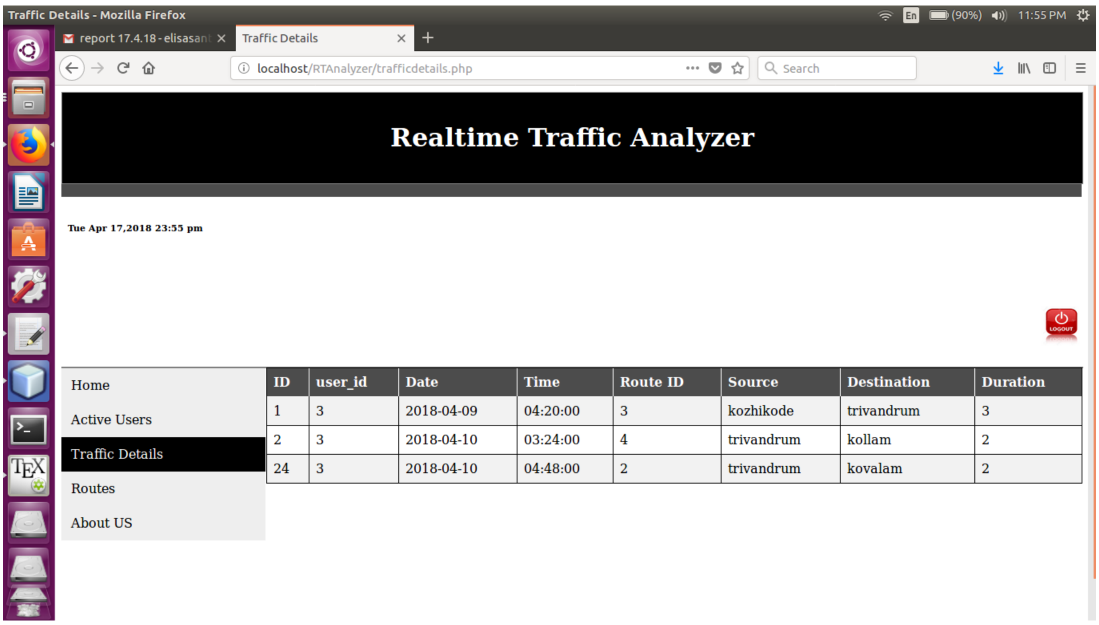

Figure 3.20: Traffic Details

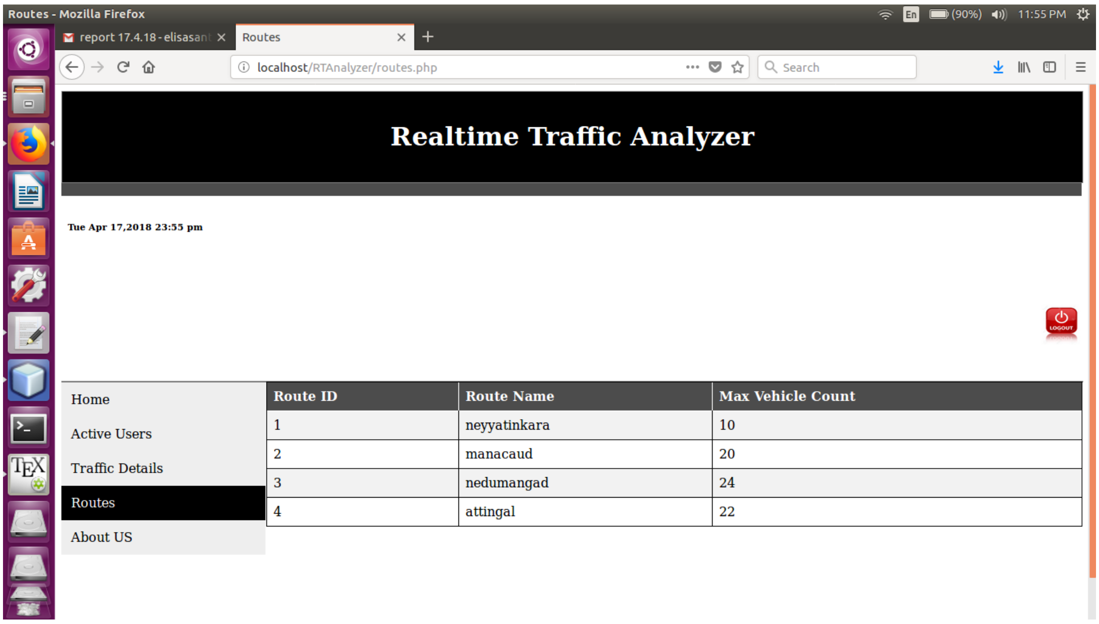

Figure 3.21: Route Details

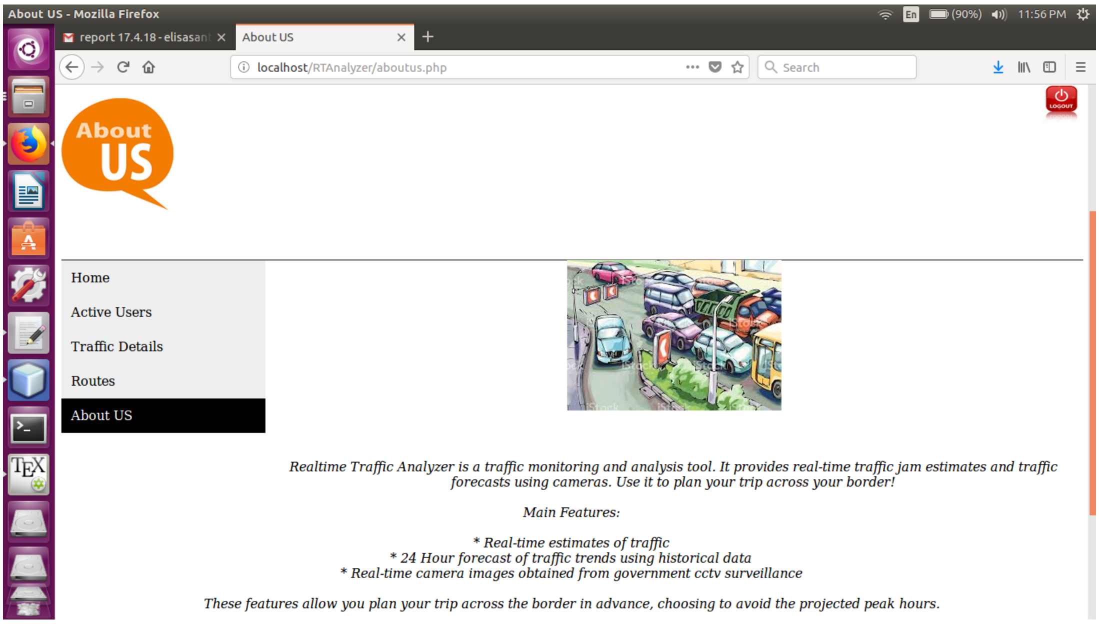

Figure 3.22: Server About US

Chapter 4

Coding

Android Application

Algorithm 1 Algorithm for Android Application is as follows:

1: Create splash screen activity to display welcome screen for 3 second by using Hander class
2: Create activity for login and register. In register button onClickListener pass data to RTAnalyzer server through AsyncTaskdoinBackhround method. Parse data using JSONParser class and receive feedback from server though AsyncTask onPostExecute method.
3: In login button onClickListener pass data through AsyncTask methods as before and when success message is received from user. Create a SharedPreference object and store username and password of user to start session.
4: Create a navigation drawer Activity HomeActivity with navigation icons as home, view journey, about us, change password and logout.
5: In HomeActivity, create a FloatingActionButton for starting new journey and navigate to StartJourneyActivity on click.
6: Create layouts and fragments for home, view journey, change password, about us and logout.
7: Create PasswordChangeFragment to send username and new password to RTAnalyzer server and receive feedback on submit.
8: In StartJourneyActivity, load routes available in server to a spinner by using AsyncTask and JSONParser on click of startjourney button. Pass wholedata to RTAnalyzer server using AsyncTask and JSONParser.
9: Create notification reciver to recive notification from server in case of traffic jam.

openCV Application

Algorithm 2 Algorithm for OpenCV Application is as follows:

1: Input video
2: Set the number of frames
3: Call PixelBasedAdaptiveSegmenter
4: Call VehicleCounting
5: Call BlobTracking
6: Read each frame
7: Process each frame and store in imgmask
8: If imgmask is not None then go to step 9
9: Process each imgmask and store in imgblob
10: Set input for imgbob
11: Set Tracks using getTracks
12: Process each imgblob
13: Update the ROI passed vehicle count and bounding box count in every 2 min to the server
14: If imgmask is None then go to step 16
15: Go to step 6 until true
16: Release frame
17: Update the counts
18: Close all windows

Real-time Traffic Analyzer SERVER

Algorithm 3 Algorithm for PHP server is as follows:

1: Start
2: Enter login credentials
3: List the active users, traffic details, vehicle count
4: Setup a crontab to run in every 15 minutes
5: Receive vehicle count from openCV application
6: Generate notifications to active users if traffic is high in the specified route
7: Stop

Chapter 5

Testing and Implementation

5.1 All the possible testing methods done for the project

System testing is the stage of implementation which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also complete verification to determine whether the objective are met and the user requirements are satisfied.The ultimate aim is quality assurance. Tests are carried and the results are compared with the expected document.In that case of erroneous results,debugging is done. Using detailed testing strategies a test plan is carried out on each module. The test plan defines the unit,integration and system testing approach. The test scope includes the following: A primary objective of testing application systems is to :assure that the system meets the full functional requirements, including quality requirements(Non functional requirements). At the end of the project development cycle, the user should find that the project has met or exceeded all of their expectations as detailed in requirements. Any changes, additions or deletions to the requirements document,functional specification or design specification will be documented and tested at the highest level of quality allowed within the remaining time of the project and within the ability of the test team. The secondary objective of testing application systems will be do:identify and expose all issues and associated risks, communicate all known issues are addressed in an appropriate matter before release This test approach document describes the appropriate strategies, process, work flows and methodologies used to plan, organize, execute and manage testing of software project "Real-time Traffic Congestion Analyzer for Road Safety"

5.1.1 Unit Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Login into the system	Invalid login is Blocked	Same as expected	Pass
2	Register the user	User is registered with valid username and password	Same as expected	Pass
3	Video stream is inputted	Each frame is segmented into foreground and background objects	Same as expected	Pass
4	Region of interest is specified	Foreground pixels from the background pixels are separated based on the intensity	Same as expected	Pass
5	Image with ROI is inputted	Frames are detected and vehicles are counted	Some false detection occurred	pass

Table 5.1: Unit test cases and results

5.1.2 Integration Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	User enters route details	Server is updated with traffic details	Same as expected	Pass
2	Vehicle count is passed to the server	Count is updated every 15 minutes	Same as expected	Pass
3	Sends Notification	User receives traffic updates notification	Same as expected	Pass

Table 5.2: Integration cases and result

5.1.3 System Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Generation of traffic and route details	Generation as per the user input	Same as expected	Pass
2	Generation of Vehicle count	Depending upon the video stream	Same as expected	Pass
3	Generation of Notification	Notify user when heavy traffic occurs	Same as expected	Pass

Table 5.3: System test cases and results

5.2 Advantages and Limitations

The proposed system consists of several advantages compared with previous systems. The vehicle detections can be traditionally achieved through inductive loop detector, infrared detector, radar detector or video-based solution. Compared to other techniques, the video-based solutions based on surveillance camera mounted outdoor are easily influenced by environments such as weather, illumination, shadow, etc. However, because video-based systems can offer several advantages over other methods such as traffic flow undisturbed, easily installed, conveniently modified, etc., they have drawn significant attention from researchers in the past decade. To overcome the previous limitations in vehicle detection and tracking, we present an improved method in this project to accurately separate the vehicle foreground from the adaptive background model with the help of pixel based adaptive segmenter.

Advantages

- Detects and count the vehicles
- Predicts Traffic flows
- Can be used to predict accident zone areas
- Ensures road safety
- Notify registered users about traffic jam in real-time

There are also some limitations to the proposed system. Automatic vehicle counting system counts somewhat less than the actual number of vehicle due to congestion and heavy traffic flow situation in one scenario. Statistical computer vision method counts more numbers of vehicles than the actual number of vehicle in video sequences due to the false positive error factor. Table below shows the experimental results obtained by the proposed method and the comparison done with the similar purpose method.

Experimental Results

Vehicle Video	Exact no. of vehicles in video	Number of vehicles calculated by the system	Success rate in percent
1	17	13	76.47%
2	27	24	88.88%
3	59	57	96.67 %
Average	103	94	91.26 %

Table 5.4: Experimental Results

5.3 Future Extensions if possible

The proposed system can be extended. From the data that server collects from user and OpenCV application, it can learn the properties of traffic, the way or route that the user mostly chooses and all. Thus, the extended system can predict the traffic hike in peak hours for each specific location. Also, users can get notified using google cloud messaging.

Chapter 6

Conclusion

A system has been developed with three modules. The user will register by giving his details and route details and he will be notified about the traffic in the route in real time. The second module is used to detect and count dynamic vehicles on highways efficiently. The detection and tracking and counting of moving vehicle can be extended to real-time live video feeds. Apart from the detection and extraction, process of recognition can also be done. Recognition techniques would require an additional database to match with the given vehicle. The system is designed for the detection and tracking and counting of a multiple moving vehicle. The detection of vehicles in a mix traffic situation of low, medium and high traffic is precisely as expected and the counting algorithm is accurate. Finally, the user details, route details and traffic details are stored in a server, which notifies the user back about the traffic. The server gets updated every 15 minutes and the OpenCV application for vehicle detection and counting updates the server with the help of a timer.

.

Bibliography

[1] . Li, B. Liang and W. Zhang, Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV, Proceedings of 4th IEEE International Conference on Information Science and Technology (ICIST), pp.631634, April 2016

[2] .K. Bhaskar and S.P. Yong, Image processing based vehicle detection and tracking method, Proceedings of 2014 International Conference on Computer and Information Sciences (ICCOINS), pp.15, June 2016

[3] . Kryjak, M. Komorkiewicz and M. Gorgon, Hardware-software implementation of vehicle detection and counting using virtual detection lines, Proceedings of 2014 Conference on Design and Architectures for Signal and Image Processing (DASIP), pp.18, October 2016.