



Daffodil
International
University

**Vehicle Component Detection for Congestion Control
using Image Processing and Machine Learning**

By

Abu Zahid Md Jalal Uddin Joy

ID: 161-35-1611

A project submitted in fulfillment of the requirements for the degree of
Bachelor of Science in Software Engineering

Department of Software Engineering

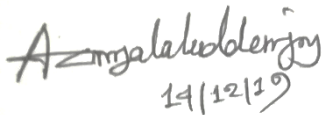
DAFFODIL INTERNATIONAL UNIVERSITY

Dhaka, Bangladesh

Fall 2019

DECLARATION

It hereby declares that this project “**Vehicle Component Detection for Congestion Control using Image Processing and Machine Learning**” under the supervision of **Dr. Touhid Bhuiyan, Professor and Head, Department of SWE**. It also declares that neither this project nor any part of this has been submitted elsewhere for award of any degree.



Abu Zahid Md Jalal Uddin Joy
14/12/19

.....
Abu Zahid Md Jalal Uddin Joy

ID: 161-35-1611

Batch: 19th

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University, Dhaka

Certified by:



.....
19/12/19

Dr. Touhid Bhuiyan

Professor and Head

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University, Dhaka

APPROVAL

This **Project** titled “**Vehicle Component Detection for Congestion Control using Image Processing and Machine Learning**”, submitted by **Abu Zahid Md Jalal Uddin Joy, ID: 161-35-1611** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc.7 in Software Engineering and approved as to its style and contents.

BOARD OF EXAMINERS

Dr. Touhid Bhuiyan
Professor and Head
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University, Dhaka, Bangladesh

Chairman

Dr. Md. Asraf Ali
Associate Professor
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University, Dhaka, Bangladesh

Internal Examiner 1

Asif Khan Shakir
Lecturer
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University, Dhaka, Bangladesh

Internal Examiner 2

Dr. Md. Nasim Akhtar
Professor
Department of Computer Science and Engineering
Faculty of Electrical and Electronic Engineering
Dhaka University of Engineering and Technology(DUET), Gazipur,
Bangladesh

External Examiner

Acknowledgement

The satisfaction of successful completion of this work would be incomplete without mentioning the name of people whose continual cooperation, constant guidance and encouragement made it possible.

I am immensely indebted to my supervisor, **Dr. Touhid Bhuiyan**, Professor and Head, Department of SWE for his wonderful guidance, inspiration, encouragement and through review of the project paper. I also would like to convey my gratitude to my internal reviewer **Mr. Md Maruf Hasan**, Assistant Professor, Dept. of SWE, for his guidance and support that will be unforgettable. I am also grateful to my parents; who have given me the tremendous inspiration and support. Without their mental and financial support, I would not be able to complete my project work. Special gratitude to my beloved and best ever family my beloved father, *Md Akhtar Hossain* and mother, *Mrs Zibon Ara Begum*.

Finally I would like to thank the great almighty “**Allah**” who gives me the patient to do such a work like “**Vehicle Component Detection for Congestion Control using Image Processing and Machine Learning**”.

Table of Contents

1. Chapter	1
Introduction.....	1
1.1. Project Overview.....	1
1.2. Purpose.....	2
1.3. Background	2
1.4. Benefits and Beneficiaries.....	3
1.4.1. Benefits	3
1.4.2. Beneficiaries	3
1.5. Goals.....	4
1.6. Proposed System Model.....	5
1.7. Project Schedule.....	7
1.7.1. Gantt chart.....	7
1.7.2. Release Plan or Milestone.....	9
2. Chapter	10
Software Requirement Specification.....	10
2.1. Functional Requirements.....	10
2.1.1. Learning new vehicles	10
2.1.2. Drawing border box for new learning.....	10
2.1.3. Learning vehicle wheels	11
2.1.4. Adding new faces to server.....	11
2.1.5. Learning new bus doors	11
2.1.6. Storing new class for prediction	11
2.1.7. Modifying any bad data	11
2.1.8. Characterizing the body/part what it had learnt.....	11
2.2. Non Functional Requirements.....	12
2.3. User Documentation Requirements	13
2.3.1. Training Requirements.....	13
2.3.2. Operational and Environmental Requirements.....	13
2.3.3. Expected Physical Requirements.....	13
2.3.4. Legal Requirements	13

3. Chapter	15
Requirement Analysis.....	15
3.1. Use Case Diagram.....	15
3.2. Use Case Description	16
3.2.1. For Drawing Border Box	17
3.2.2. For Changing Class Index.....	17
3.2.3. For Drawing Text Label.....	18
3.2.4. For Changing Image Index	18
3.2.5. For Getting Border Box Area	19
3.3. Activity Diagram.....	20
3.3.1. Activity Diagram for Drawing Border Box	20
3.3.2. Activity Diagram for Changing Class Index	21
3.3.3. Activity Diagram for Drawing Text Label	22
3.3.4. Activity Diagram for Image Indexing.....	22
3.3.5. Activity Diagram Getting Border Box Area	23
3.4. Sequence Diagrams	23
3.4.1. Sequence Diagram for Drawing Border Box.....	24
3.4.2. Sequence Diagram for Changing Class Index	25
3.4.3. Sequence Diagram for Drawing Text Label	25
3.4.4. Sequence Diagram for Indexing Image	26
3.4.5. Sequence Diagram for Getting Border Box Area	27
4. Chapter	28
System Design Specification.....	28
4.1. Development tools and technology	28
User Interface Technology.....	28
Programming Language.....	28
4.2. Class Diagram	30
4.3. Entity Relation Database diagram (ERD)	31
5. Chapter	32
System Test.....	32
5.1. Testing Features	32
5.1.1. Features to be tested.....	32
5.2. Testing Strategy.....	33

5.2.1.	Test approach	33
5.2.2.	Pass / Fail Criteria.....	33
5.3.	Trace Ability Matrix.....	34
5.4.	Testing Environment	34
5.5.	Test Cases.....	35
5.5.1.	Drawing Border Box.....	36
5.5.2.	Image Indexing	36
5.5.3.	Load Database Table.....	36
5.5.4.	Class Index.....	37
5.5.5.	Show image label details	37
6.	Chapter	38
	User Manual	38
6.1.	Image Indexing.....	38
6.2.	Class Indexing	40
6.3.	Drawing Border Box	41
7.	Chapter	42
	Conclusion	42
7.1.	Project Summary	42
7.2.	Limitations	43
7.3.	Obstacles and Achievements.....	43
7.4.	Future Scope.....	44
7.5.	References	44

Table of Figures

Figure 1: Proposed System Model.....	5
Figure 2: Project release plan and milestone	9
Figure 3: Use Case Diagram of Vehicle (Organ) Detection using Image Processing and Machine Learning for Congestion Control.....	16
Figure 4: Activity Diagram for Drawing Border Box	20
Figure 5: Activity Diagram for Changing Class Index.....	21
Figure 6: Activity Diagram for Drawing Text Label.....	22
Figure 7: Activity Diagram for Image Indexing	22
Figure 8: Activity Diagram Getting Border Box Area	23
Figure 9: Sequence Diagram for Drawing Border Box	24
Figure 10: Sequence Diagram for Changing Class Index.....	25
Figure 11: Sequence Diagram for Drawing Text Label.....	26
Figure 12: Sequence Diagram for Indexing Image.....	26
Figure 13: Sequence Diagram for Getting Border Box Area	27
Figure 14: Class Diagram	30
Figure 15: Entity Relation Database diagram (ERD)	31
Figure 16: Image Indexing Scroll Bar	39
Figure 17: How to change Class Indexing	40
Figure 18: How to Draw Border Box	41

1. Chapter

Introduction

Traffic congestion is a severe problem in many major cities across the world and it has become a nightmare for the commuters in these cities (like, Dhaka in Bangladesh). The Traffic congestion can also be caused by large Red light delays, etc. The delay of respective light is hard coded in the traffic light and it is not dependent on traffic. Therefore for simulating and optimizing traffic control to better accommodate this increasing demand is arises. Now-a-days, we are facing a lot of traffic problem due to increased number of vehicles. People are unable to reach their destination on time due to huge traffic. With the help of image processing, we are going to develop an automated project that follows some rules and depend upon several factors i.e. number of vehicles and rate of vehicles approaching towards the traffic signal poles. According to the rules based upon several factors the timer of traffic light can be controlled by detecting vehicles and analyzing the road congestion.

1.1. Project Overview

Traffic control systems should cope with the ever increasing demand by determining the situation on the road network and by controlling traffic flows. Here, the system contains CCTV surveillance footage which captures video photography of road and transmits the footage to the server. These are mounted from the sides of roads. The system gets counted whenever any vehicle passes/come forward to the coverage area of camera on. In Adaptive Traffic Control System which receives information from vehicle such as type, components, position and speed and then it utilize to optimize the traffic signal. Microcontroller controls

the Vehicle Component Detection [1] system and counts number of vehicles passing on road. Microcontroller also store vehicles count in its memory was processed by yolov3 object classifier [2]. Based on different vehicles count, the microcontroller takes decision and updates the traffic light delays as a result. The traffic light is situated at a certain distance. Thus based on vehicle count, microcontroller defines different ranges for traffic light delays and updates those accordingly.

1.2. Purpose

Focusing on the main use of this project, we aim at controlling traffic density using our self-deployed microcontroller.

1.3. Background

Density, speed, and flow are the three critical parameters for road traffic analysis. High-performance road traffic management and control require real-time estimation of space mean speed and density as input for large spatial and temporal coverage of the roadway network. Controlling road traffic using an automated system will bring a new era in our country. Bangladesh is an over populated uncontrolled road traffic country. So, I cherished how to control the overflowed road congestion. Imagining this problem, suddenly I cherished it will be really amazing if that automated software is being developed by me! So I tried hard to develop this project. In the first stage, my task was to detect the parts of a bus. Like the wheels, doors, glasses of bus, people on the road. So, before working on moving image I concentrated on the still image to detect this type of parts. If this project come to true in this country and the full proposal is completed, Bangladesh traffic control authority really will be so much rich.

1.4. Benefits and Beneficiaries

This **Vehicle Component Detection for Congestion Control using Image Processing and Machine Learning** would be benefits and also so many beneficiaries in some point.

Mentioning those below:

1.4.1. Benefits

By using this automated software system **Bangladesh Road Transport Authority (BRTA)** will able to control road congestion. There are also many benefits. These are explaining as:

- a) Able to fulfill the Bangladesh government road safety requirement.
- b) Web based system make to easy access in anywhere for authorized person.
- c) Storing vehicle information in cloud so it will make more safe and secure.
- d) Find out any vehicle information easily with details.
- e) Stored penalty information to find out monthly total fee.
- f) Sorting client based on monthly bill pay.
- g) Easily find clients who are unpaid penalty fee.
- h) Show backlog vehicle's information.
- i) Separate authorization make sure the proper security and authenticity.
- j) It reduces the dependency on man power and paper's.

1.4.2. Beneficiaries

Every project creates a new value in the universe & there are so many people/organizations that can be benefited. So considering about that beneficiaries,

- k) Bangladesh Road Transport Authority (BRTA)
- l) Road Traffic Control Authority
- m) Government of Bangladesh
- n) Vehicle owners

- o) Vehicle insurance company

1.5. Goals

Every Project must have a specific goal. This project has also main goals and to develop this computer based application, be helpful to administration to make a decision using graphical view of the business for controlling the road congestion. Such the stakeholders as:

- a) Road Traffic Control Authority: Can control traffic efficiently.
- b) Government of Bangladesh able to monitor how much vehicles are on the road.
- c) Since the system is automated and connected with internet, vehicle owners must be benefited if the vehicle is stolen.
- d) Vehicle Insurance company can increase the service coz the road safety will be raised.

1.6. Proposed System Model

For developing Vehicle Component Detection system, I have proposed a system model. The model described below will clarify the system in a brief.

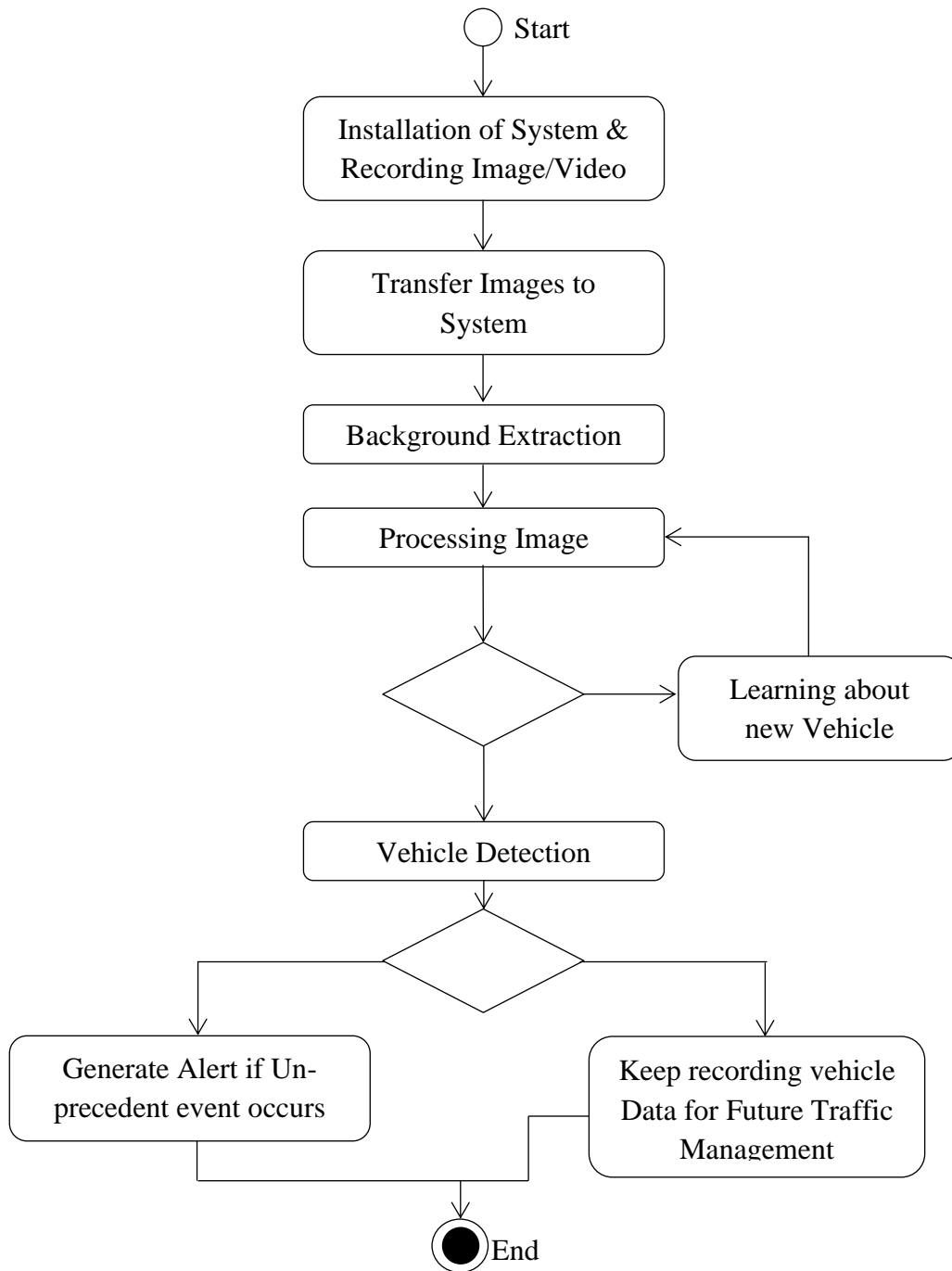


Figure 1: Proposed System Model

In this system, Administrator will install the developed package software to the micro controller processor. When the CCTV camera will record the video all the information will be sent automatically to the server and images will be fragmented frame by frame. The developed system software will process the image. Since the software is being developed by vehicle detection classifier, it will able to detect vehicles. After detecting vehicles the system will keep recorded vehicle Data for Future Traffic Management.

1.7. Project Schedule

Every process has a scheduling plane to complete the project on time properly. It also helps the developer and all related employee for maintain time schedule.

1.7.1. Gantt chart

During the time of developing this project, illustrating how it will run, Gantt chart is used as the timeline. As a project management here it can be viewed as individual tasks, their durations and the sequencing of these tasks. View the overall timeline of the project and the expected completion date are in brief:

Activities	Weeks	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Planning	Ideas	█															
	Problem Definition	█	█														
	Proposal Planning	█	█														
Requirements	Requirement specification		█	█													
	Requirement analysis		█														
QA-1	Quality Assurance			█													
System Design	Sketching				█												
	Design specification				█	█											
	Database Design					█											
Implementation -1	Machine learning about vehicles					█	█	█	█	█	█						
QA-2	Test Cases			█				█	█	█	█						
Implementation -2	Vehicle detection & showing										█	█	█	█	█		

1.7.2. Release Plan or Milestone

This chart is mapping the forward movement in our project plan. It also signifies a change or stage in development and it is just monitoring the tasks and not necessarily following the right path in this project.

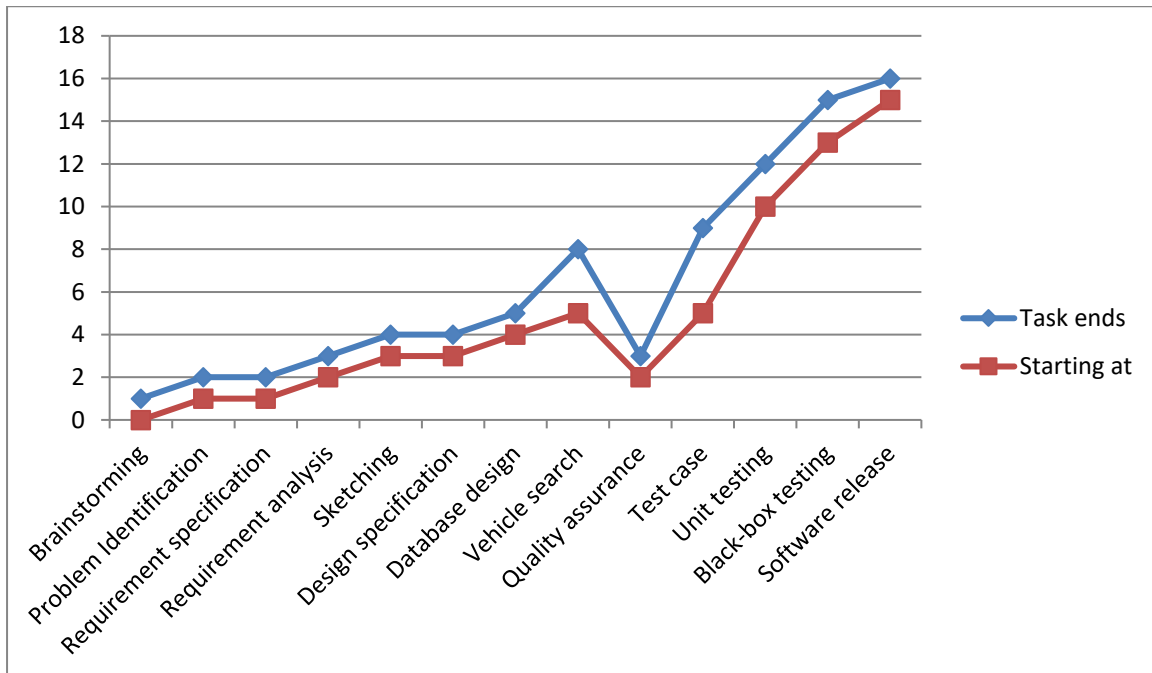


Figure 2: Project release plan and milestone

Here the red line defines task starting week & the blue line defines task ending week. So, distance between the two lines defines total duration of week. This project was initiated to develop between 16 weeks & the team was successfully delivered to the client.

2. Chapter

Software Requirement Specification

This chapter explains the detailed description of this software system to be developed with its functional and non-functional requirements. Here this SRS is developed based the agreement between customer and developer/contractor. It also includes the cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for this project development. To develop the software system we have clear understanding of this type of Software system. To achieve this we have followed continuous communication with customers to gather all requirements.

2.1. Functional Requirements

On this system, these functional requirements are the desired operations described what the software system would able to do. Here the developed SRS are showing to mention functional requirements associating with this project are given below.

2.1.1. Learning new vehicles

Requirements 1	System will able to learn about new bus
Description	This novice system must have to teach about a new vehicle whether it is bus or not by the admin. So, it's able to learn about new vehicle.
Stakeholders	Admin

2.1.2. Drawing border box for new learning

Requirements 2	System must allow admin to draw new border box for any new learning
Description	For any new learning, system shows a new border box to determine the area of learning.
Stakeholders	Admin

2.1.3. Learning vehicle wheels

Requirements 3	System will able to learn about new bus wheels
Description	The brain of this system is newly developed. So, it's totally unknown about any parts of the buses. By drawing the border box, it's able to learn the wheel, Furthermore the class should be changed to "wheels" of the window.
Stakeholders	Admin

2.1.4. Adding new faces to server

Requirements 4	System will allow admin to add new faces
Description	This system is able to learn about new faces on the road or vehicles. Admin have to teach whether it is face or not !
Stakeholders	Admin

2.1.5. Learning new bus doors

Requirements 5	System shall allow to learn about new bus doors
Description	System learns new bus doors by the admin.
Stakeholders	Admin

2.1.6. Storing new class for prediction

Requirements 6	System must store every new learning class for further prediction
Description	System stores the data what it had learnt newly
Stakeholders	System

2.1.7. Modifying any bad data

Requirements 7	System must allow admin to modify or delete of any bad data that was learnt
Description	If the system taught wrong data information, admin is able to delete that bad data
Stakeholders	Admin

2.1.8. Characterizing the body/part what it had learnt

Requirements 8	System is able to characterize the body/part what it had learnt already
Description	The final stage, what the system learnt already is able to characterize or determine by showing the border box of any new images
Stakeholders	System

2.2. Non Functional Requirements

Here these are explained as the non-functional requirement because they are essential to ensure the usability and effectiveness of the entire software system. The undersigned non-functional requirement defines the quality attribute of this software system. They also are representing the set of standards used to judge the specific operation of this system. Failing to meet non-functional requirements also can result in systems that fail to satisfy our user needs. Example, how fast does the website load etc. & so more are describing as below:

S/L	Requirement Name	Description
1	Capacity	Processing an image per second will be satisfied with a minimum number of process time.
2	Reliability	The system service will be available of 999/1000 or 99% which means that out of 1000 request for this service, 999 must be satisfied.
3	Usability	An interface will be as easy to learn how to use and easy to remember how to use. The interface will be easy to learn/use and helpline, buttons, headlines, and help/error message are simple to understand.
4	Performance	Loading performance of this system will take maximum in 2 seconds for a viewing a page
5	Content Accessibility	The system will have maximum 5 clicks to perform an event.
6	Safety	Accidental occurrences or unusual situational images will be blurred during the loading of the web pages. If authority/admin want to view, he/she can uncover it.
7	Security	The access permissions for system data may only be changed by the system's data administrator.
8	Supportability	<p>Requirements are concerned with the ease of changes to the system after deployment.</p> <p>Supportability requirements may have related to some extends. Like:</p> <ul style="list-style-type: none"> a) Testability b) Extensibility c) Adaptability d) Maintainability e) Compatibility f) Configurability g) Serviceability h) Install ability <p>This application meets all of the above requirements related to supportability.</p>

9	Recoverability	All the data of system will be backed up every 24 hours and the backup copies stored in a secured location which is not in the same building/server. The system be able to save or update any data in database within 20 seconds.
---	----------------	---

2.3. User Documentation Requirements

Every document has two types. One is internal documentation which document is generally written by the application engineers. This document prepared to make development life cycle easier for the system engineers or system analysts.

UDR-1	The system engineer documentation.
Description	To develop this application named Vehicle Detection/classification. we have a system analysis team as well as documentation team.
Stakeholders	System analysts or software developers.

2.3.1. Training Requirements

After developing and deploying this application software, user must to take the training requirements that are very necessary to properly train up admin users to the system so that they would be capable to operate this system easily. After launching the full package to the market, firstly I'll provide training to the different users like Admin.

2.3.2. Operational and Environmental Requirements

Performance, capabilities, measurements, process are referred by Operational and environmental requirement

2.3.3. Expected Physical Requirements

There are no expected physical requirements in this system.

2.3.4. Legal Requirements

Legal requirements mainly mention the terms and conditions for privacy and policy of any organizations. The terms and condition of this application is that, no third party software without cloud is allowed to engage to use the data for their safety/security/business purpose.

3. Chapter

Requirement Analysis

This analysis of software engineering is to bridging the gap between system requirement and software design. The undersigned requirements are defined as the software capability that will met or possessed by this system component to satisfy the contract. Ultimately, what we want to achieve is to develop quality software that meets our customers' real needs on time. In this chapter use case, activity diagram & sequence diagram are described as follows.

3.1. Use Case Diagram

The set of actions, services, and functions that the system needs to perform is described below is called as use case diagram.

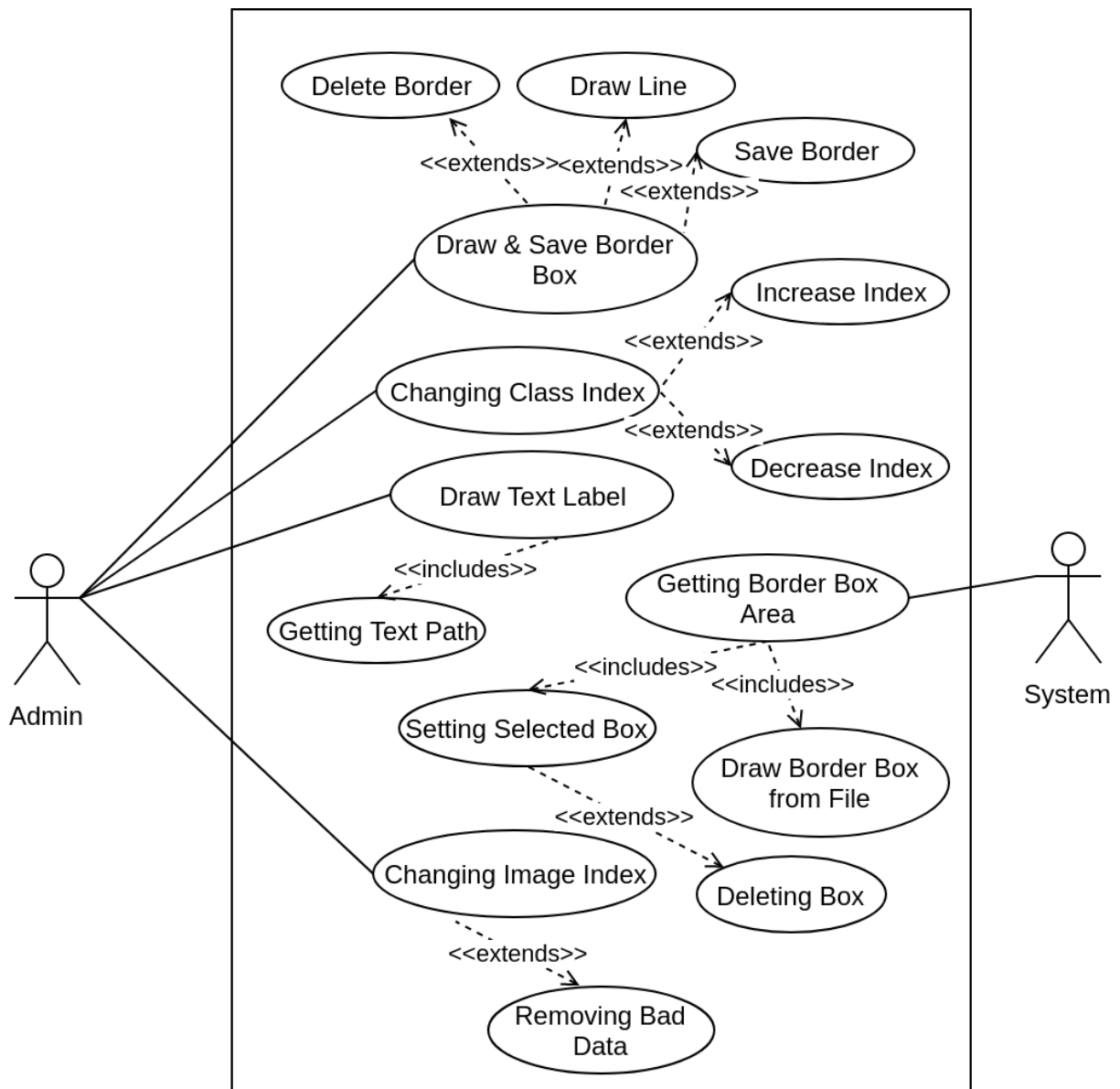


Figure 3: Use Case Diagram of Vehicle (Organ) Detection using Image Processing and Machine Learning for Congestion Control

3.2. Use Case Description

The use case model described above of this business software interactions between a user (called an actor) and a system (usually a computer system). It details the interactions and sets the expectations of how the user will work within the system.

3.2.1. For Drawing Border Box

Use Case	Drawing Border Box	
Goal	To drawing border box for learning purpose	
Preconditions	<ol style="list-style-type: none"> 1. A new image must be loaded 2. That image must include vehicle parts (i.e.: wheels, glasses, doors etc.) 	
Success End Condition	<ol style="list-style-type: none"> 1. Admin loads new image to the system 2. Admin draws new learning border box and define 	
Failed End Condition	<ol style="list-style-type: none"> 1. Admin unable to load new images for the sake of system response failure 2. System busy for doing another process 	
Primary Actors:	Admin	
Secondary Actors:	System	
Trigger	Mouse on click	
Description / Main Success Scenario	Steps	Action
	1.	Admin loads new image to the system
	1.1	System shall allow the new images to its server
	2.	Admin draws new border box using it's predefined class like (bus doors/ wheels/ people faces)
	2.1	System adopt the border region and learns well
Alternative Flows	Steps	Branching Action
	1a	System declines for loading new images
	2a	For processing another job, system shows busy to allow new border box
Quality Requirements	Not Applicable	

3.2.2. For Changing Class Index

Use Case	Changing Class Index	
Goal	To change the class label i.e. whether it is bus door or wheels or people faces or not!	
Preconditions	<ol style="list-style-type: none"> 1. Image must be uploaded 	
Success End Condition	<ol style="list-style-type: none"> 1. Admin will able to learn the system new components 	
Failed End Condition	<ol style="list-style-type: none"> 1. Tough for admin to learn newly developed methods 	
Primary Actors:	Admin	
Secondary Actors:	System	
Trigger	Swap to Change Class	
Description / Main Success Scenario	Steps	Action
	1.	Swap the class changing button
Alternative Flows	Steps	Branching Action
	1a	Unable to change class for the sake of system not responding

	1b	Image size isn't compatible to the system
Quality Requirements	Not Applicable	

3.2.3. For Drawing Text Label

Use Case	Drawing Text Label	
Goal	To help (label) the system what it has learning?	
Preconditions	<ol style="list-style-type: none"> 1. Border Box should be drawn 2. Before drawing border box, concentration on class whether it is right to label or not 	
Success End Condition	1. Admin will able to learn the system new components with name	
Failed End Condition	1. System shows busy for processing another image	
Primary Actors:	Admin	
Secondary Actors:	System	
Trigger	N/A	
Description / Main Success Scenario	Steps	Action
	1.	Draw border box and Text label will show by its own
Alternative Flows	Steps	Branching Action
	1a	Unable to change class for the sake of system not responding
Quality Requirements	Not Applicable	

3.2.4. For Changing Image Index

Use Case	Changing Image Index	
Goal	To modify and change newly added image index	
Preconditions	1. Must have to add new image with indexing	
Success End Condition	1. Admin will able to modify image indexing	
Failed End Condition	1. System failure for changing that indexing	
Primary Actors:	Admin	
Secondary Actors:	System	
Trigger	N/A	
Description / Main Success Scenario	Steps	Action
	1.	Admin adds new image to the server
Alternative Flows	Steps	Branching Action
	1a	Indexing capacity full
Quality Requirements	Not Applicable	

3.2.5. For Getting Border Box Area

Use Case	Getting Border Box Area	
Goal	To show the learnt components	
Preconditions	1. System must be learnt	
Success End Condition	1. System shows what it had learnt	
Failed End Condition	1. System fails to detect the new component	
Primary Actors:	System	
Secondary Actors:	Admin	
Trigger	Run Program	
Description / Main Success Scenario	Steps	Action
	1.	Start to run program, system shows if any components know to it
Alternative Flows	Steps	Branching Action
	1a	Unknown components
Quality Requirements	Not Applicable	

3.3. Activity Diagram

Activity diagram are used to draw the important behavioral diagram in UML diagram to describe dynamic aspects of this system. This is essentially an advanced version of flow chart that modeling the flow from one activity to another activity. Here it is used for modeling how a collection of use cases coordinate to represent business workflows.

3.3.1. Activity Diagram for Drawing Border Box

Drawing border surrounding to the components like, bus wheel, bus door, vehicles etc. is helping the machine what to be learnt! So the workflow of this type of important behavioral activity is needed to understand. At the very first activity diagram, the flow of Drawing Border Box is given as below.

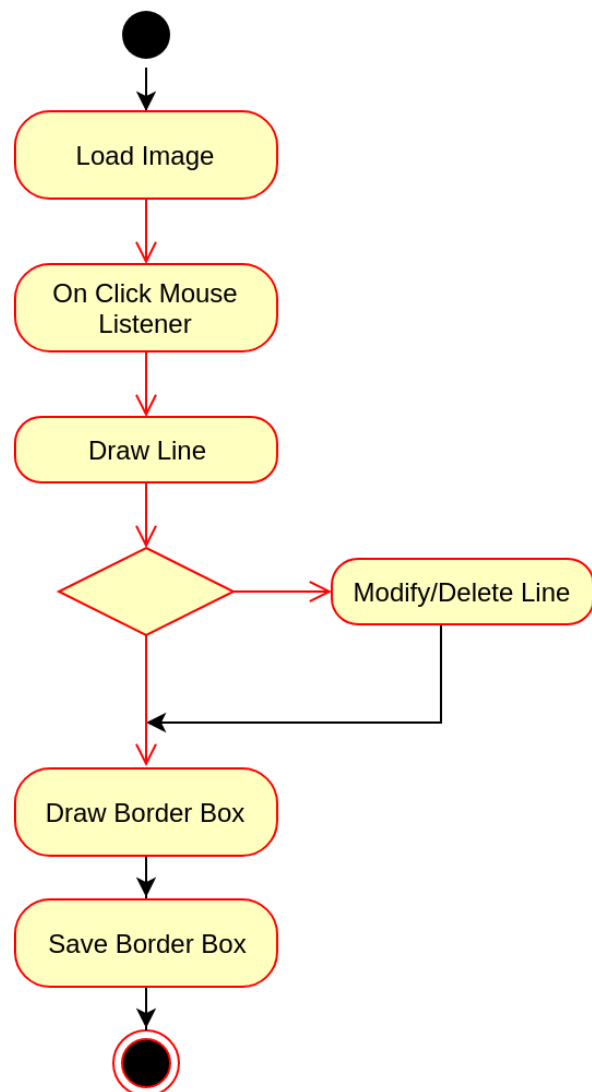


Figure 4: Activity Diagram for Drawing Border Box

3.3.2. Activity Diagram for Changing Class Index

Here is a class index which makes differentiate what the system should learn. After the time of loading new image the system tries to learn from the admin. Here, the admin shall able to change class index by scrolling the scroll bar. The workflow is described as:

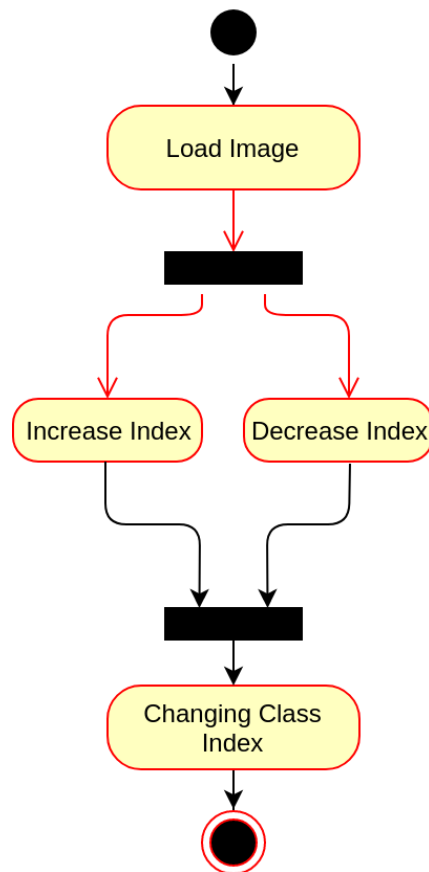


Figure 5: Activity Diagram for Changing Class Index

3.3.3. Activity Diagram for Drawing Text Label

During the time of loading new image to the system, if it is already learnt before; the label will be shown by its own. The system will allow loading that labeling information from the database server and the user will see the name of the components in that image. If not, admin can help the software system to learn about it. So, the admin will help the system by drawing new border box with the text label by changing the class index. If the entire tasks are done successfully, the workflow of drawing text label will be succeeded.

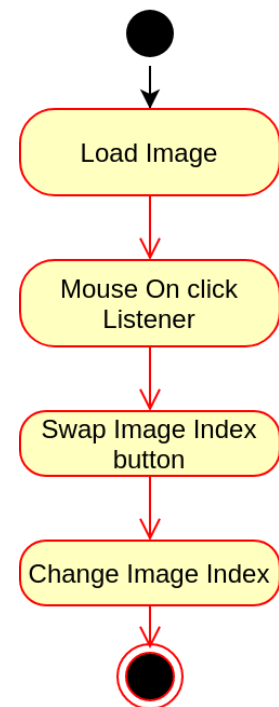


Figure 6: Activity Diagram for Drawing Text Label

3.3.4. Activity Diagram for Image Indexing

As called the object-oriented flowchart the activity diagram of this is defined as a UML

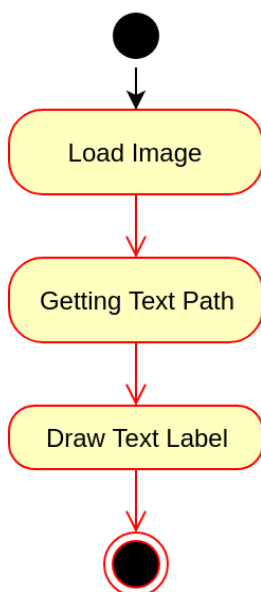


diagram that focuses on the execution and flow of the behavior of image indexing in this system. In this Image indexing activity diagrams consist of activities that are made up of actions which apply to behavioral modeling technology. Here image on the right side used for fork and join nodes, concurrent flow within this activity can be generated. The fork node has incoming edge and numerous outgoing edges. It is similar to one too many decision parameters. When data arrives at loading image edge vehicle component detection has more than one image so that it can

Figure 7: Activity Diagram for Image Indexing

predict correctly throughout the matter. So for why here are several images which help to learn. Keep tracking on mages of the same window of the output screen is quite tough.

3.3.5. Activity Diagram Getting Border Box Area

During the execution time of the software, the system can detect the pre learnt component of vehicles and load the border box. That workflow explained as below.

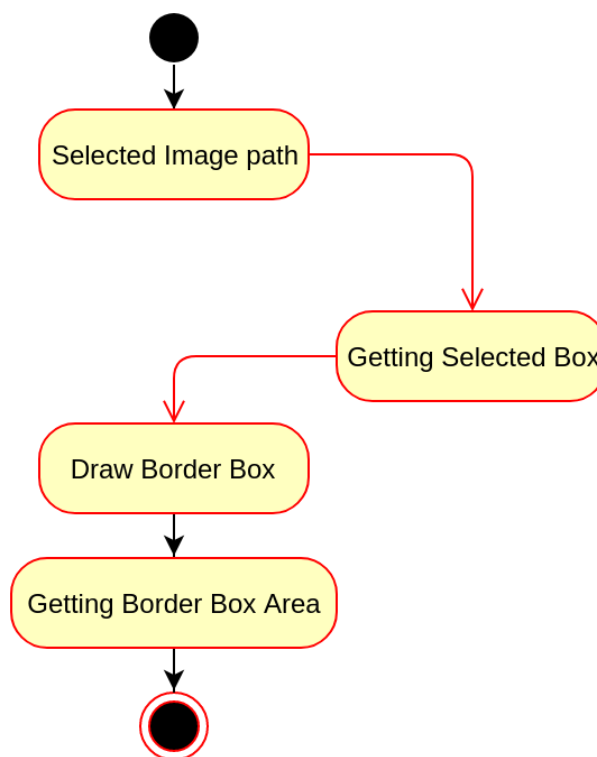


Figure 8: Activity Diagram Getting Border Box Area

3.4. Sequence Diagrams

The sequence diagram of this project is used primarily to show the interactions between objects in the sequential order that those interactions occur. Much like the class diagram, developers typically think sequence diagrams were meant exclusively for them. However, all the business staff of use can find sequence diagrams useful to communicate how the business currently works by showing how various business objects interact. By looking on this

organization’s technical staff can find sequence diagrams useful in documenting how this system will behave. During the design phase, architects and developers can use the diagram to force out the system’s object interactions, thus fleshing out overall system design. For why we focus much on sequence diagrams are given below.

3.4.1. Sequence Diagram for Drawing Border Box

Here the interaction between objects in a sequential order which is attached with administration panel, User Interface window & System. The method *load_image()*, *display_image()* etc. exclusively below are showing how the business object interacts.

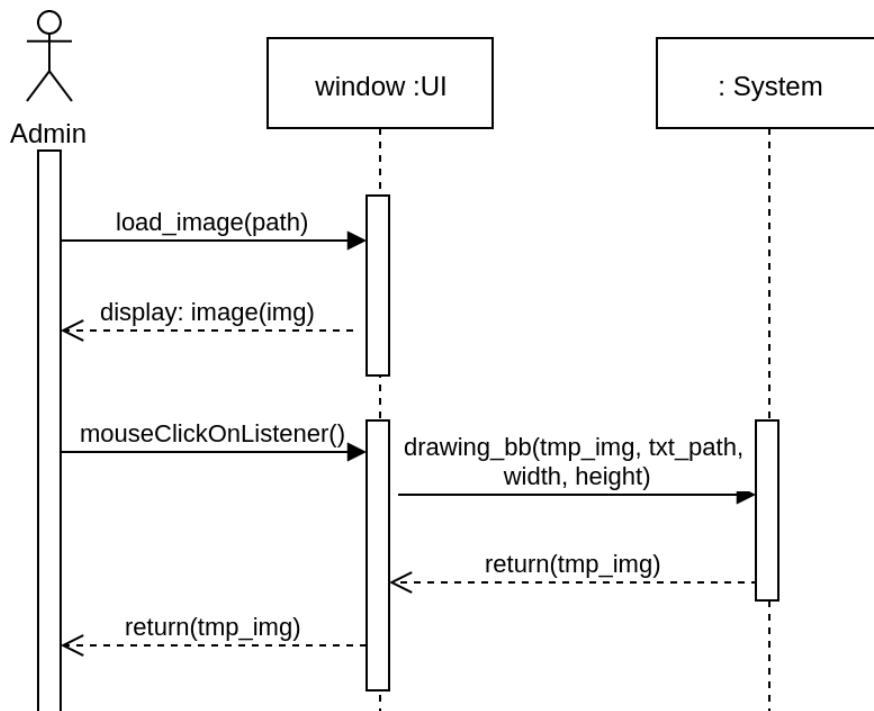


Figure 9: Sequence Diagram for Drawing Border Box

3.4.2. Sequence Diagram for Changing Class Index

During this design phase, architects and developers can use the diagram to force out the system's object interactions, between admin and system. The workflow changing class index has several steps which are described as bellows:

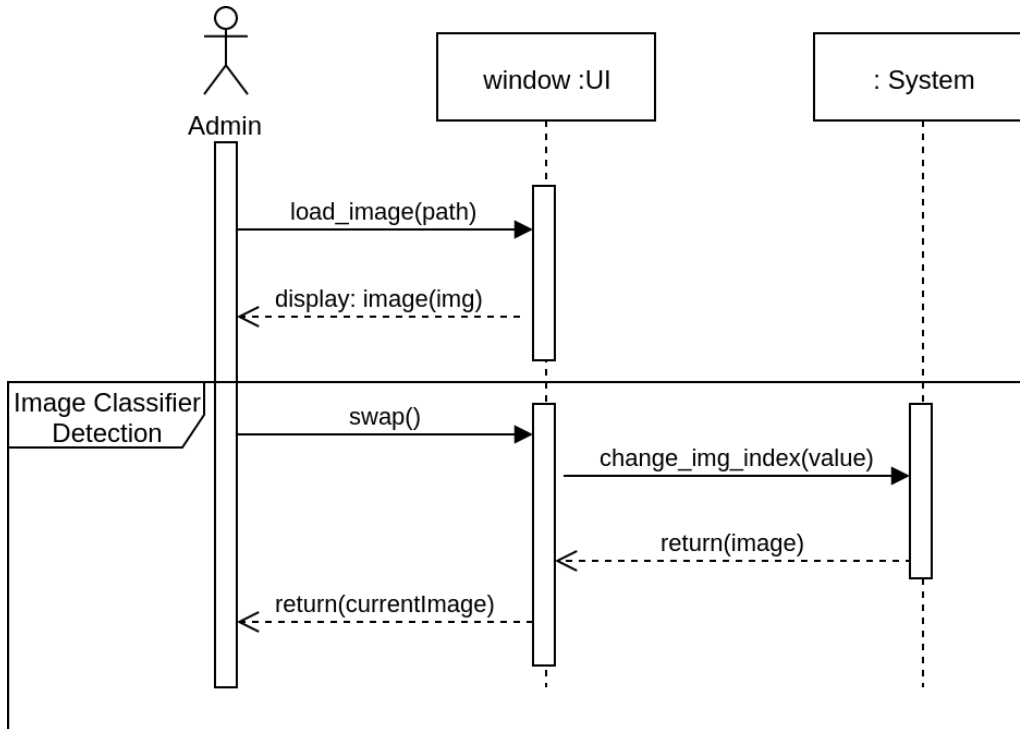


Figure 10: Sequence Diagram for Changing Class Index

3.4.3. Sequence Diagram for Drawing Text Label

Defining the workflow of Text Label is between the admin and system. Here the method `load_image`, `display`, `drawBorder` is described as beside with this diagram.

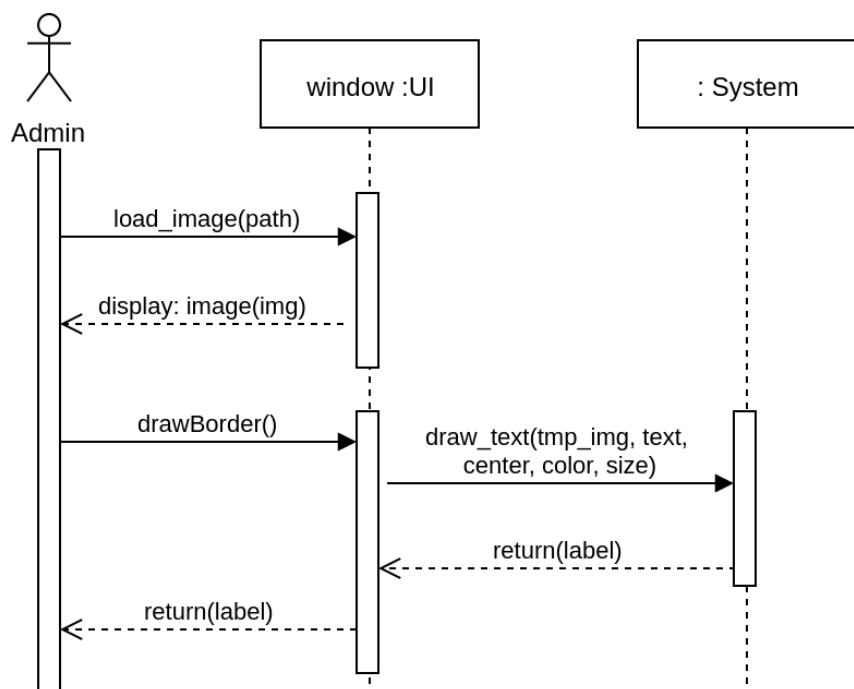


Figure 11: Sequence Diagram for Drawing Text Label

3.4.4. Sequence Diagram for Indexing Image

To index an image the method `load_image()`, `display_image()`, `newIndexing()` is making a bridge between the administration, UI and the system.

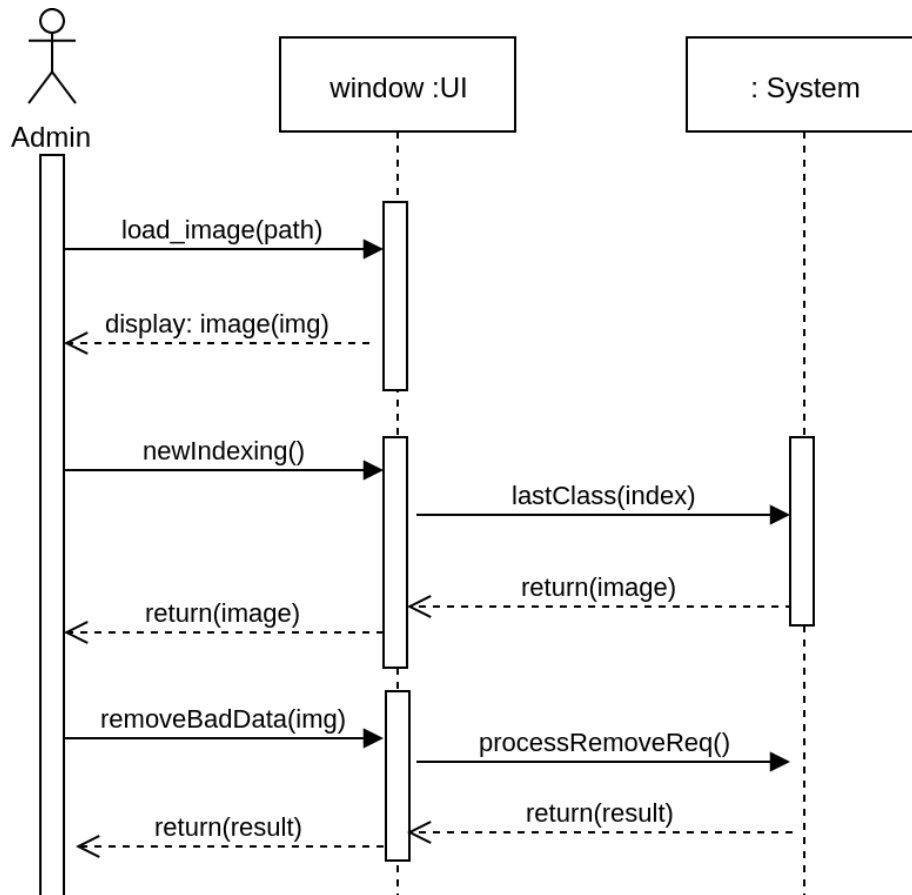


Figure 12: Sequence Diagram for Indexing Image

3.4.5. Sequence Diagram for Getting Border Box Area

For getting border box area from the storage, system has to load the exact request & then loads border labeling of that image using the image path. The component border contains height, weight and text path. The total sequence workflow is given below:

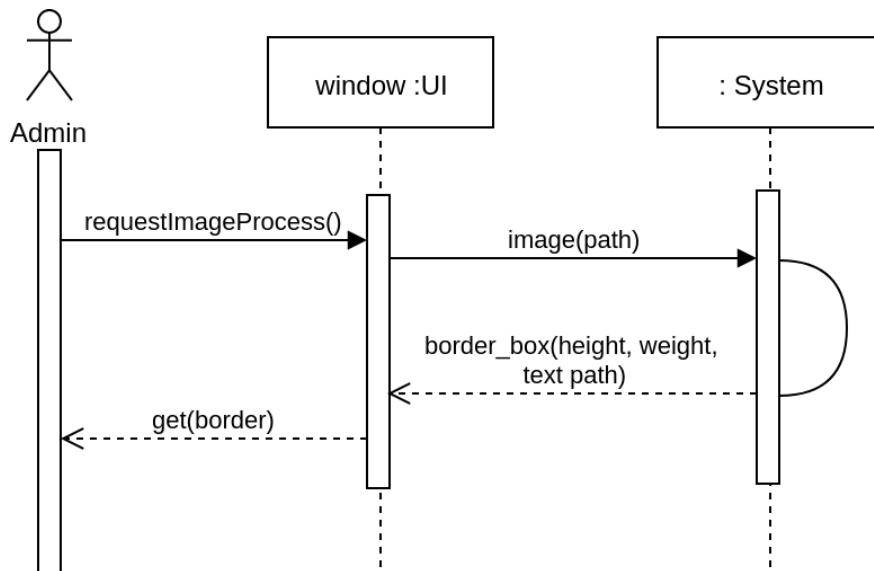


Figure 13: Sequence Diagram for Getting Border Box Area

4. Chapter

System Design Specification

This chapter is described as a detailed document, providing a list of points regarding this software as a product or process. For example, the design specification could include required dimensions, development tools and technology, class diagram, entity relationship diagram, maintenance that will be needed, etc. It also gives specific examples of how the design should be executed, helping others work properly (a guideline for what the person/user should do).

4.1. Development tools and technology

The programming tool or software development tool is a computer program that software developers use to create, debug, maintain, or otherwise support other programs and applications. This branch of knowledge that deals with the creation and use of technical means described as below.

User Interface Technology

I use jQuery for validation using JavaScript, Bootstrap for frontend development. I use Python for data graphical view.

Programming Language

Developing software, the most important part is programming language because the security performance and some of other features are related of programming language. I use Python programming language for server side programming with [3] YOLOv3, openCV [9]. Both of them are open source general purpose language.

Server configurations:

- a) Python support [\[7\]](#)
- b) Local Storage
- c) YoloV3 (You Only Look Once) Object Detector[\[4\]](#)[\[5\]](#)
- d) OpenCV
- e) NumPy
- f) Unix based operating systems (Linux)
- g) Other versions or platforms may work but are not supported and may not have been tested.

4.2. Class Diagram

Class diagram is a static diagram. Here it is going to represent the static view of this application. It is also used for visualizing, describing and documenting different aspects of this system but also for constructing executable code of this software application. In here the diagram describing the attributes and operations of a class and also the constraints imposed on the system. Like activity diagram, sequence diagram can only give the sequence flow of the application; however class diagram is a bit different. It is the most popular UML diagram in the coder community. The main purpose of the class diagram can be summarized as –Analysis and design of the static view of this application. It also describing the responsibilities of this system is following as below:

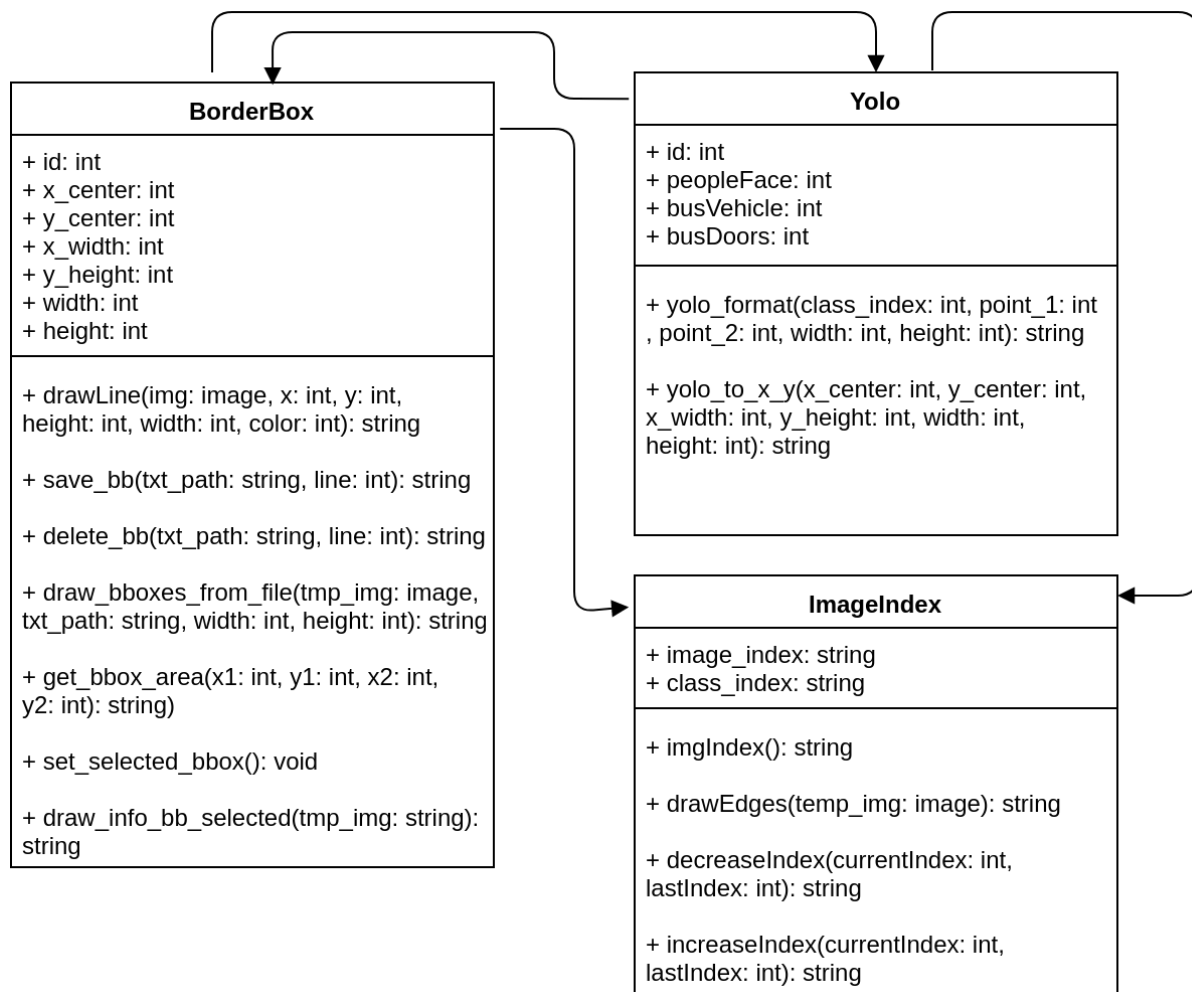


Figure 14: Class Diagram

4.3. Entity Relation Database diagram (ERD)

The entity relationship diagram (ERD) of this is shows the relationships of entity sets stored in the database. Here the entity in this context is an object, a component of data. The entity set is also the collection of similar entities. These entities have the attributes that define its properties. By defining the entities, their attributes, and showing the relationships between them, here the ER diagram illustrates the logical structure of the databases. ER diagrams are used to sketch out the design of a database.

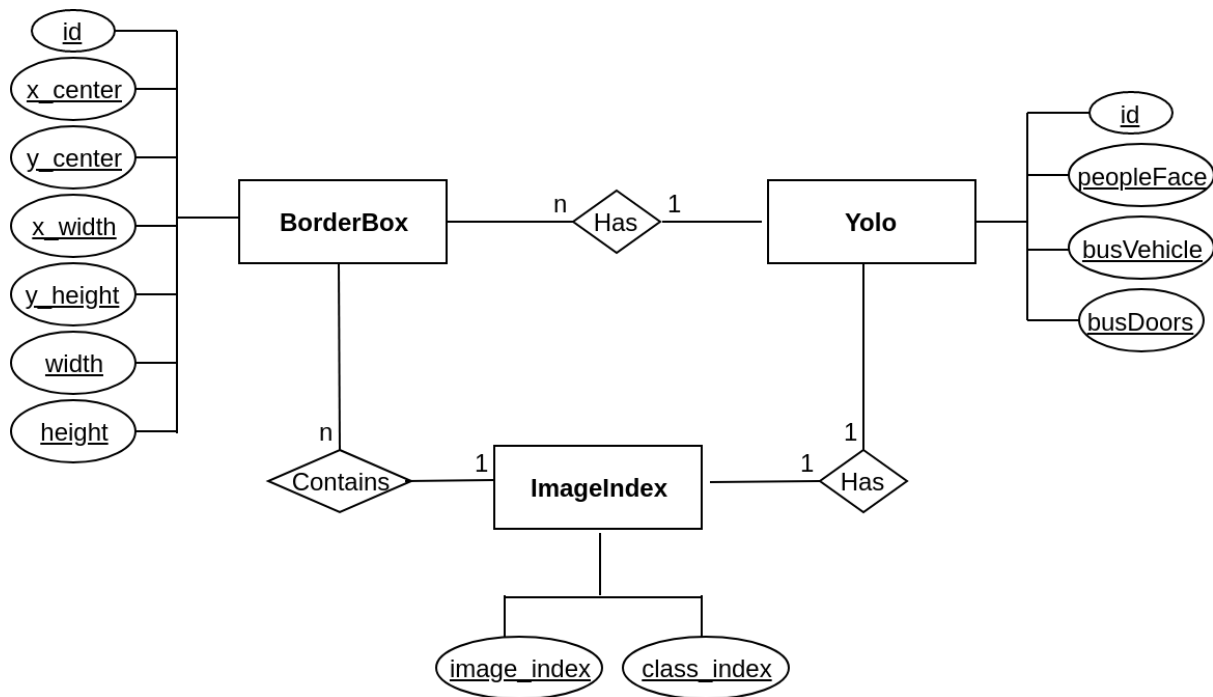


Figure 15: Entity Relation Database diagram (ERD)

5. Chapter

System Test

The System Testing is a testing of complete and fully integrated software product. Usually software has only one element of a larger computer-based system. Ultimately, software is interfaced with other software/hardware systems. Here System Testing is actually the series of different tests whose sole purpose is to exercise the full computer-based system.

5.1. Testing Features

Feature Testing: Features are changes that add new functionality or modify existing functionality.

Functional Testing: This Functional Testing is that type of testing technique that is used to test the functionality of this software is able to cover all the scenarios.

5.1.1. Features to be tested

Features	Priority	Description
Drawing Border Box	1	To learn the exact portion of an image, a border box must be drawn
Image Index	2	Indexing an image to proper exemplification
Load database table	1	Load database and generate query properly in time.
Class Index	1	Changing Learning organ/components
Show image label details	2	To showing what it had learnt

Here,

3 = Low priority,

2 = Medium Priority,

1 = High priority.

5.2. Testing Strategy

For testing any systems there are follow some of strategy for testing the process. Mainly testing the object, methods, total available resources are testing using this testing. Mainly I've used to test some feature which may have error or hacked. After that it should be reviewed by the test team leads. Different kinds of testing strategies are going to be performed according to the type of application system that needs to be tested.

5.2.1. Test approach

Completing the whole process, the tester must take some approach. I'm going to show the two testing approach for clarify the testing process.

- a) **Automation testing:** Automation testing is a name of testing technique by which test engineers prepare some scripts according to test plan and after that they use suitable tools to perform testing of the software. Nowadays, almost every software company follows the approach of automation testing.

- b) **Manual testing:** Manual testing is also a name of technique of testing by searching out the bugs or vulnerability in an application. In this process, test engineers manually test and execute the test cases without having any automation tools.

5.2.2. Pass / Fail Criteria

Pass or fail criteria will be set by the test engineers. They will prepare the pass / fail criteria on the basis of which input data are worked and which are not works well. Those data that are worked well will consider as pass criteria. And rest of input data will be considered as fail criteria.

Now I will give the pass / fail criteria below.

- a) System crash will not be considered as pass case.
- b) If any criteria pass 100% times, then it will be considered as pass criteria only.
- c) If data can't be displayed to the application properly, then it is also to be considered as fail criteria.

5.3. Trace Ability Matrix

Project Manager			Business Analyst Lead		
QA Lead			Target Implementation Date		
BR#	Category / Functionality / Activity	Requirement Description	Use Case Reference	Test Case Reference	Comments
BR-1	Functional	Drawing Border Box	Use case 3.2.1	Test Case 5.6.1	
BR-2	Functional	Image Index	Use case 3.2.4	Test Case 5.6.2	
BR-3	Functional	Load database table	N/A	Test Case 5.6.3	
BR-4	Functional	Class Index	Use case 3.2.2	Test Case 5.6.4	
BR-5	Functional	Show image label details	Use case 3.2.3	Test Case 5.6.5	

5.4. Testing Environment

Testing environment means to prepare the environment with hardware and software so that test engineers can be able to execute test cases as required. Besides hardware and software usage, network configuration might be needed to execute test plans. For making the environment for testing, some key area need to setup. Those are:

- a) Test data
- b) Database server
- c) Front end running environment
- d) System and application

- e) Network
- f) Hardware with server operating system
- g) Documentation is also required. Like: user manuals, installation guides, configuration guides, documents etc.

5.5. Test Cases

This test case refers to some rules and regulations or conditions by which it can be determined whether a system can be able to meet the works or requirements under test cases properly. We know there is a chance to have some fault or break holes in any application. This is very a common scenario. And those issues are solved usually by software testing approaches. But if we don't care of those issues, then the full application development might be ruined. So proper testing must be needed. For testing this application, I have prepared some test cases. Now, I am going to provide them.

5.5.1. Drawing Border Box

Test Case # 1	Test case name: Drawing Border Box		
System: Vehicle (Organ) Detection using Image Processing and Machine Learning For Congestion Control	Subsystem: Admin		
Designed by: Abu Zahid Md Jalal Uddin Joy	Designed date: 25-October-2019		
Executed by:	Executed date: 1 st November 2019		
Short Description: Admin loads new image to the system. System shall allow the new images to its server. Admin draws new border box using its predefined class like (bus doors/ wheels/ people faces). System adopts the border region and learns well.			
Pre-conditions: a) User must be authorized. b) Image should be loaded successfully.			
Step	Expected Result	Pass/Fail	Comment
Mouse on click	Border box drawn	Passed	
Post-condition: Users (admin officer) will successfully able to draw the border box.			

5.5.2. Image Indexing

Test Case # 2	Test case name: Image Indexing		
System: Vehicle (Organ) Detection using Image Processing and Machine Learning For Congestion Control	Subsystem: Admin		
Designed by: Abu Zahid Md Jalal Uddin Joy	Designed date: 25-October-2019		
Executed by:	Executed date: 1 st November 2019		
Short Description: Admin adds new image to the server			
Pre-conditions: a) User must be authorized. b) Image should be loaded successfully.			
Step	Expected Result	Pass/Fail	Comment
Image Indexing swap the button	Swapped	passed	
Post-condition: N/A			

5.5.3. Load Database Table

Test Case # 3	Test case name: Load Database Table		
System: Vehicle (Organ) Detection using Image Processing and Machine Learning For Congestion Control	Subsystem: Admin		
Designed by: Abu Zahid Md Jalal Uddin Joy	Designed date: 30-October-2019		
Executed by:	Executed date: 1 st November 2019		
Short Description: Admin loads database for detecting new image components			
Pre-conditions:			

a) User must be authorized. b) Machine should be learnt successfully before.			
Step	Expected Result	Pass/Fail	Comment
New images loaded and class button change	Component detection successfully	passed	
Post-condition: N/A			

5.5.4. Class Index

Test Case # 4	Test case name: Class Index		
System: Vehicle (Organ) Detection using Image Processing and Machine Learning For Congestion Control	Subsystem: Admin		
Designed by: Abu Zahid Md Jalal Uddin Joy	Designed date: 30-October-2019		
Executed by:	Executed date: 3 rd November 2019		
Short Description: Admin loads Images and defines the four classes that had developed to be learnt by machine/computer.			
Pre-conditions:			
a) User must be authorized. b) Machine should be learnt successfully before. c) Images should contain bus doors, bus wheels or people faces.			
Step	Expected Result	Pass/Fail	Comment
Swap the indexing button to change class	Class index changed	passed	
Post-condition: N/A			

5.5.5. Show image label details

Test Case # 5	Test case name: Show image label details		
System: Vehicle (Organ) Detection using Image Processing and Machine Learning For Congestion Control	Subsystem: Admin		
Designed by: Abu Zahid Md Jalal Uddin Joy	Designed date: 30-October-2019		
Executed by:	Executed date: 3 rd November 2019		
Short Description: Loading each image, the label should be visible			
Pre-conditions:			
a) Machine should be learnt successfully before. b) Images should contain bus doors, bus wheels or people faces and that must be pre learnt			
Step	Expected Result	Pass/Fail	Comment
Loading a new images and label details visible	label details visible	passed	
Post-condition: N/A			

6. Chapter

User Manual

User documentation is it called a user manual, user guide, or other, is usually provided to customers. Still, the users frequently keep asking the same questions again and again. So for why here we are providing the User Manual as bellows.

6.1. Image Indexing

After debugging the software system, Admin is able to change the image indexing [\[8\]](#) and scrolling the bar, admin must have to learn the system.



Figure 16: Image Indexing Scroll Bar

6.2. Class Indexing

To learn the system, it is necessary to scroll the class scroll bar and help to determine the bus doors, bus wheels or people faces and then click the mouse.

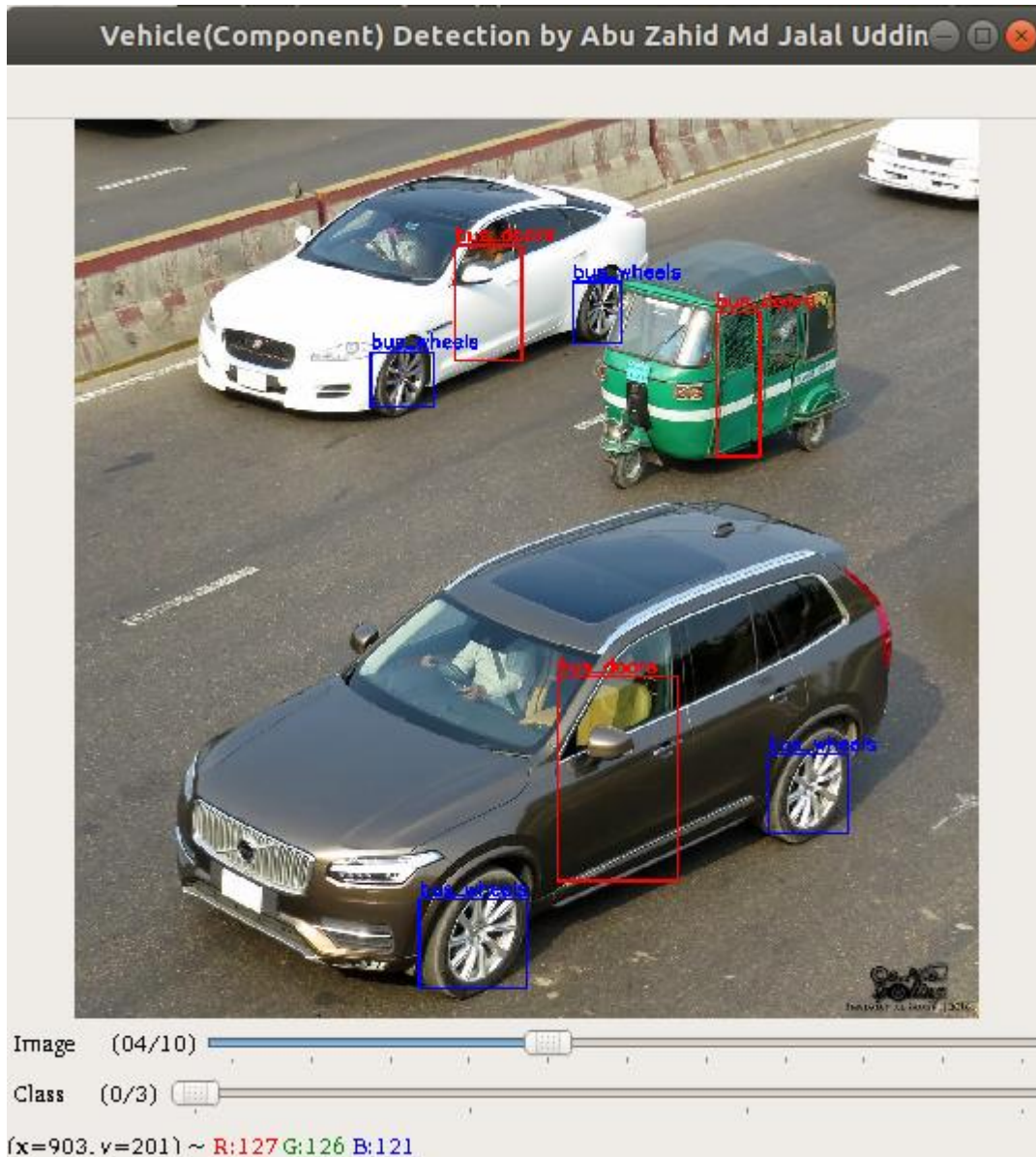


Figure 17: How to change Class Indexing

6.3. Drawing Border Box

After learning about an image, it's really easy to the system to detect the components again and what it had learnt. So, the run the program again and see what it detect! It will able to detect the bus doors, bus wheels, people faces again. It also will draw the border box.

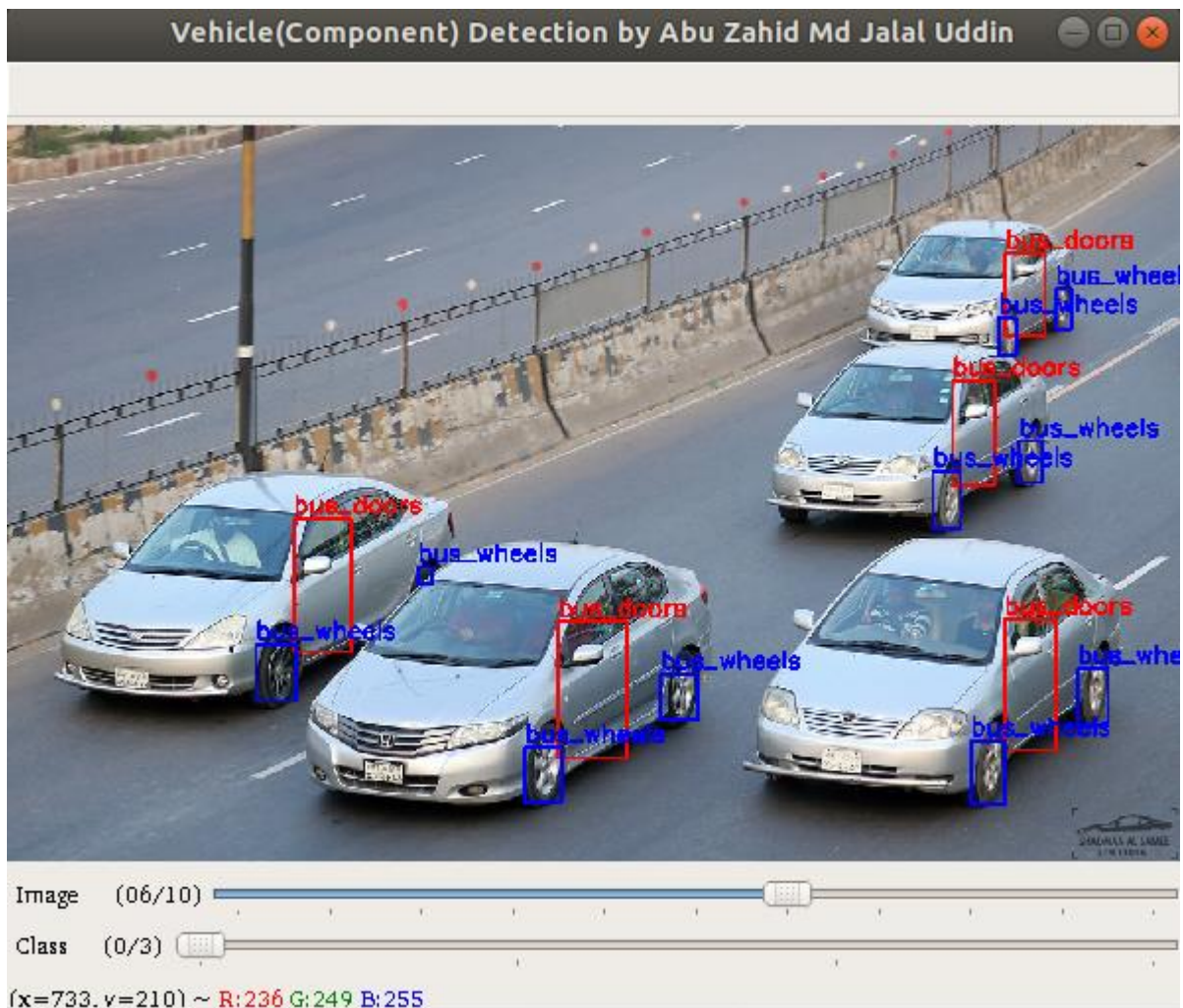


Figure 18: How to Draw Border Box

7. Chapter

Conclusion

This chapter summarizes the conclusions of this project. The technical and logistical difficulties are reviewed carefully. Suggested improvements are outlined and the limitations of the project are reported. Section 7.4 identifies some suggestions for future work, while this chapter provides the overall conclusions on the project and section 7.1 gives a summary of the chapter

7.1. Project Summary

We have started to develop this project since it was May-2019. From the very starting of development of this project, it demands hard working, patience, persistency to meet the requirements of stakeholders. After that I have proposed the design and then started to work. Here I came to know that base knowledge of programming play a vital role for developing any king of software/program. Hence this is a project based on Image Processing and Machine Learning, I'd to know about the basic of deep learning, artificial neural networks. I also had to learn how these technology works. What is the object detection classifier/model? After learning the basic knowledge I did initiated to develop this.

Since developing this project, I've focused on core functionality. After that I have prepared the user interface and take the approval to continue to the next part. It is to be said that, the interface of my application is very simple and easy to understand. After completing that, I have started to write the core functionality of the project.

Developing a project, the entire task isn't completed. There are some other important tasks to perform. And that is testing. It is also known as quality assurance also. Every software

company there is a quality assurance team. Their main responsibility is to find the loop holes or vulnerability of software. If there any bug remains before handover to the stakeholders, there is a change to ruin the whole project. So testing plan is very important. And after developing the project, I have assured the quality of this project.

7.2. Limitations

Developing this project, there also remained some limitations. Now I will describe those in brief.

- a) **Live Streaming [\[11\]](#) Object Detection:** Though this project is implemented by an image and video object detection classifier using yolov3 (object detector) models with OpenCV. This developed project is capable to detect object in images using Deep Learning, OpenCV, and Python but not video streams now.
- b) **Detecting any type objects/vehicles:** This project isn't enough intelligent to detect any kind of vehicles/objects/components of the vehicles yet.

7.3. Obstacles and Achievements

I believe that if there are not any obstacles to develop a project, Obstacles, challenges and achievements are like a path to the success. By starting this project, I came to know the actual flow and real life software development and implementation. After that I have learnt system analysis, database design and many things. My supervisor helps me a lot from the very beginning of the development of this project.

7.4. Future Scope

I have learnt a lot throughout the whole development stage of this project. I have also met with some young technologists. I am very much thankful to all of them as their idea and discussion gave me some opportunities to make my product complete.

Here, I'll take the initiative to develop (complete) this project by video object detection classifier using yolov3 (object detector) models with OpenCV. This project will be capable to detect object in video streaming using Deep Learning [10], OpenCV, Python and so on.

7.5. References

- [1] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik (2013). **Rich feature hierarchies for accurate object detection and semantic segmentation**. 1311.2524, arXiv, cs.CV
- [2] **YOLO object detection with OpenCV**; Adrian Rosebrock; 12 November, 2018; Available: <https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/>
- [3] **YOLO algorithm- objects detection**; Michal Maj; 22 August, 2018; Available: <https://appsilon.com/object-detection-yolo-algorithm/>
- [4] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (8 Jun 2015); **Computer Vision and Pattern Recognition**. arXiv:1506.02640
- [5] Joseph Redmon, Ali Farhadi (8 Apr 2018); **YOLOv3: An Incremental Improvement**; arXiv:1804.02767v1 [cs.CV]

- [6] **Darknet: Open Source Neural Networks in C;**
Available: <https://pjreddie.com/darknet/>
- [7] You only look once (YOLO) -- (1); Aug 20, 2018;
Available: <https://zhanghanduo.github.io/post/yolo1/>
- [8] Keze Wang, Liang Lin, Xiaopeng Yan, Ziliang Chen, Dongyu Zhang, and Lei Zhang
(12 January 2019); **Cost-effective Object Detection: Active Sample Mining with Switchable Selection Criteria**; IEEE arXiv:1807.00147v3 [cs.CV]
- [9] **OpenCV Release**; December 22, 2018; Available: <https://opencv.org/releases/>
- [10] **Latest Deep Learning OCR with Keras and Supervisely in 15 minutes**; November 1st 2017; Available: <https://hackernoon.com/latest-deep-learning-ocr-with-keras-and-supervisely-in-15-minutes-34aec630ed8>
- [11] **Vehicle Image Database**;
Available: http://www.gti.ssr.upm.es/data/Vehicle_database.html